ONLINE ASSESSMENT OF GRAPH THEORY

A thesis submitted for the degree of Doctor of Philosophy

By
Justin Dale Hatt

College of Engineering, Design and Physical Sciences
Brunel University
August 2016

Abstract

The objective of this thesis is to establish whether or not online, objective questions in elementary graph theory can be written in a way that exploits the medium of computer-aided assessment. This required the identification and resolution of question design and programming issues. The resulting questions were trialled to give an extensive set of answer files which were analysed to identify whether computer delivery affected the questions in any adverse ways and, if so, to identify practical ways round these issues.

A library of questions spanning commonly-taught topics in elementary graph theory has been designed, programmed and added to the graph theory topic within an online assessment and learning tool used at Brunel University called Mathletics. Distracters coded into the questions are based on errors students are likely to make, partially evidenced by final examination scripts. Questions were provided to students in Discrete Mathematics modules with an extensive collection of results compiled for analysis. Questions designed for use in practice environments were trialled on students from 2007 – 2008 and then from 2008 to 2014 inclusive under separate testing conditions. Particular focus is made on the relationship of facility and discrimination between comparable questions during this period. Data is grouped between topic and also year group for the 2008 – 2014 tests, namely 2008 to 2011 and 2011 to 2014, so that it may then be determined what factors, if any, had an effect on the overall results for these questions.

Based on the analyses performed, it may be concluded that although CAA questions provide students with a means for improving their learning in this field of mathematics, what makes a question more challenging is not solely based on the number of ways a student can work out his/her solution but also on several other factors that depend on the topic itself.

Acknowledgements

My experience of academic study and research within the United Kingdom has indeed been exhausting and challenging. However, I am very grateful for this experience and especially for all of the support I have received during this time. There are many people I wish to thank for this extensively long experience… also, this is the one part of the thesis where I can speak casually, so I am taking advantage of this opportunity while I can.

I must thank the academic staff within the Department of Mathematics at Lakehead University. The academic support I received was incredible and I was treated with great care throughout my undergraduate degree. I especially wish to thank Dr. Liping Liu, who introduced me to the Atlantic Association for Research in the Mathematical Sciences (AARMS), which eventually led me to the opportunity to relocate to the United Kingdom to conduct postgraduate research.

I must thank Dr. Anthony and Mrs. Sarah Aldous, whom I met at AARMS, as well as Dr. Manti Mendi for supporting me in my move to the United Kingdom and my transition into a postgraduate research career. Their support has been excellent and I am especially grateful to Dr. and Mrs. Aldous for providing me a place to stay whenever needed, especially more recently when I visited them in Singapore.

You will probably notice that I am playing to a particular timeline here, so that is why it is fitting now to thank my Ph.D. supervisor, Dr. Martin Greenhow, for putting up with me all of this time. This has not been such a straightforward Ph.D. degree to complete and neither of us knew what was going to come out of this kind of research. However stressful or difficult the experience may have appeared to be at times, the experience of conducting research and understanding its benefits are absolutely incredible and I am very happy to have been able to use my mathematics training and also a bit of my educational training to create such useful and practical research in mathematics.

I want to thank Bloomsbury Central Baptist Church for their incredible support of me since I joined them in 2007. It was quite the stretch for me to travel from Uxbridge in London's Zone 6 to Holborn in Zone 1 just to attend a church, but to then find fellow Canadians in attendance, along with so many academics… from the Reverend Doctors of Theology to a former President of a postsecondary institution to medical doctors, Masters students, etc., was simply too good to pass up. The community atmosphere has been incredible and although some difficult times have occurred, there is still a great sense of community care between everyone. I am especially grateful to the ministers, Rev. Dr. Ruth Gouldbourne and Rev. Dr. Simon Woodman, for their extended support, both spiritually and financially (for help in attending an international conference several years ago now), and former minister, Rev. Dr. Simon Perry, for his casual, relaxed view in understanding our place in the world and how God works through us within society. I also wish to thank Tim Jones, Dave and Sandy Porter, Peter How, Elsie Martin, Rosel Schmidt, Ruth Johnson, Barbara Collett, Chris Green, Helen Walsh, Howard Brown, the Filipino Community, Daniel and Bryant Taffou, and especially Ben Haden for the friendship, kindness, and incredible support all of them have provided me within the church during this time.

Speaking of Canadians, I really have to thank the London Expat Canadian Meetup community of expatriate Canadians living in the United Kingdom. I have made so many friends through my involvement with this group as co-organiser and the friendships have been absolutely excellent. I especially wish to thank the organiser, Dave Mathews, along with Vicky Whaley, Johnson Tse, Cynthia Fisher, and so many others (Sorry for not being able to mention all of you!) for all of their kindness and gratitude in keeping me calm and sane during this experience as a Canadian living overseas.

I also want to thank other people back home, including Darlene Robertson, Theresa Flanagan, Tiffany Johnson, Kimberly Johnson, Misty-Dawn Anderson, Travis Gagné, Pat Norman, Shelley Sowers, Heidi Conrod, Conrad Cline, Shannon Neathway, Helen Woodcock, Sydney Linton, Dominique Scott, Deanna Hanley-Luevano, Kim Lomax, Miranda Jones, Tracy Spear, Jen Campbell, Liz Monks, Jillian Monaghan, Penny Cooke, Stacey Tucker, Jenna Martell, and any others I have forgotten to mention (Again… sorry!) for the help and support provided to me back in Canada when I greatly needed it.

During my time in employment as a secondary school teacher, I have made some rather good friends whom I wish to thank. I want to thank Joanne Wright and Jo Franklin for their help in adjusting to teaching in a new educational environment, Ian Tay, Rob Mathers, Hilary Downey, Helene Ng Wheeler, Stephanie Cuthbertson, Dave Hinchcliffe, Günter Morson, and Cameron Mitchell, with whom I worked outside London.

That should be most everyone. Sorry to those who I forgot to mention… again; it's a Canadian thing to be overly apologetic and, ironically, I'm not apologising for that.

Table of Contents

List of Figures

List of Tables

To my mother for convincing me to get away from home in order to explore my potential elsewhere and to God for putting it all in place so that I could be in this position today.

Chapter 1 Introduction

The objective for this introduction is to provide a motivation for the work undertaken in this thesis and to understand it in the context of the following:

- Possible users of this software
- Previous and current computer-aided assessment in use, providing advantages and disadvantages of use, an understanding of its framework, and some applications already available
- Types of assessment and the design of questions
- Analysing assessments

1.1 Background and Motivation

Decision mathematics is a subject within mathematics that spans multiple topics and reaches multiple disciplines. In the A-level syllabi of the three major U.K. examination boards, namely EdExcel[1], AQA[2], and OCR (with MEI)[3], although the topics are generally the same throughout, the location of topics between each module varies, as shown in Table 1.1; this is rather important because Decision Mathematics 1 (D1) is an AS-level module, whereas Decision Mathematics 2 (D2) is an A2-level module. D2 can only be studied by students if they have already studied D1. Students who study to obtain one A-level credit only need to take two applied modules. Statistics 1 (S1) and Mechanics 1 (M1) are alternative options and students will already be familiar with statistics as they will have learned some of the key topics from S1 in their GCSEs.

Upon contacting OCR, it was learned that for the 2013 – 2014 academic year, approximately 13,800 students sat their D1 examination (between their regular OCR module and their OCR MEI module), but only about 1,540 students sat D2. However, OCR was hesitant to provide exact numbers and information, citing that this information was "commercially sensitive".

Information provided from AQA, however, provided some detailed insight. They provided exact numbers from 2009 – 2014. The results are shown in Table

1.2. They have significantly higher numbers of students sitting their D1 and D2 examinations than OCR, but what is more interesting is the significant increase in students sitting the D1 examinations in 2013 – 2014. According to AQA, students who sat the 2012 – 2013 examination were well prepared[4], so it is unexpected that there were a significant number of students resitting this module in 2013 – 2014. Requests to communicate with EdExcel resulted in a link[5] to a webpage on their site, which provides details only for each mathematics award it offers, rather than student numbers for each module; it is worth noting that EdExcel, unlike AQA and OCR, is not a registered charity, and so, OCR's noted concern of "commercially sensitive" data might be a reason for the lack of available information.

Module	Decision Mathematics 1	Decision Mathematics 2
AQA	Algorithms Graphs & Networks Spanning Trees Matchings Shortest Paths in Networks Route Inspection Problem Travelling Salesperson Problem Linear Programming	Critical Path Analysis Allocation Dynamic Programming Network Flows Linear Programming: Simplex Method Game Theory for Zero Sum Games
EdExcel	Algorithms Algorithms in Graphs The Route Inspection Problem Critical Path Analysis Linear Programming Matchings	Transportation Problems Allocation Travelling Salesperson Problem Linear Programming: Simplex Method Game Theory Network Flows Dynamic Programming
OCR	Algorithms Graph Theory Networks Linear Programming (including Simplex Method)	Game Theory Network Flows Matchings and Allocations Critical Path Analysis Dynamic Programming

Table 1.1 Topics covered by U.K. examination boards for Decision Mathematics 1 and Decision Mathematics 2. Data expected to be updated with upcoming changes to A-level syllabus across the country.

Examination boards do give syllabi, past examination papers, mark schemes, and examiners' reports for each assessment it provides. However, examination boards can choose to not award marks for follow-through work completed correctly where a mistake occurred earlier in the problem solving process. Also, examiners' reports do not detail reasons for students' errors in problem solving, but rather simply general details about each question and how students performed overall.

Year	D1	D2
2013 – 2014	15,222	2,847
2012 – 2013	11,918	1,986
2011 – 2012	11,352	1,602
2010 – 2011	10,123	1,553
2009 – 2010	9,183	1,540
TOTAL	57,798	9,528

Table 1.2 Numbers of students who sat AQA D1 and D2 examinations from 2009 – 2014.

At the postsecondary level, discrete mathematics can play a similar role to decision mathematics in that it can encompass many similar topics. The topics of linear programming and graph theory, which appear in D1 and D2, can appear as separate modules in postsecondary mathematics courses. Linear programming often involves the use of simplex tableaus to determine optimal solutions to problems using different methods, whereas graph theory will use a combination of graphs and adjacency matrices to better understand networks and their algorithms.

Brunel University currently has two modules that have a focus on topics within graph theory, namely MA0422 (Discrete and Decision Mathematics)[6] and MA2726 (Elements of Combinatorics)[7]. There are approximately 100 students who register for MA0422 each academic year, all of whom are enrolled the Foundations of Information Technology (FoIT) programme. There are approximately 125 students who register for MA2726 each academic year, all of whom are enrolled in a B.Sc. course for mathematics. Previously, graph theory appeared in the module, MA2920, and it was not necessary at the time for all students who were enrolled in the B.Sc. course for mathematics to study this module; however, all B.Sc. mathematics students must now study MA2726.

At Brunel University, the numbers of students taking discrete mathematics is high, comprising all mathematics and computer science students, along with electrical engineering students. It is estimated that over 400 students are studying discrete mathematics to some extent at the university. Students studying economics or business studies may indirectly encounter some of the topics found in discrete mathematics or graph theory later in their courses. The potential for the number of students in the United Kingdom studying discrete mathematics at the post-secondary level is excellent with possibly over 60,000 students in the post-secondary sector alone studying discrete mathematics[1,8].

Due to the significantly large number of students needing to study graph theory-related topics, it is important that any modules teaching these topics are manageable. As will be explained later, the use of online learning and assessment is helpful in providing additional learning tools to students and managing large-scale assessments, which is especially important in a post-secondary environment. Additionally, although there is some expected functionality of matrices that can be coded to generate algorithms with which to solve problems in linear programming, generating graphs so that observable properties can be inspected is more difficult and answering questions related to these properties can require some intriguing mathematical insight. This thesis will specifically look at designing graph theory questions for use in an online environment. Later sections in this chapter will further explain the rationale behind the research conducted, which will then lead to the research questions to be answered for this thesis.

1.2 Computer-Aided Assessment and Learning

1.2.1 Definitions

The history of the internet is somewhat recent; it was not until 1991 that the Internet was introduced for public use as the World Wide Web[9]. However, online assessment and learning (also known as **e-assessment** and **e-learning**) dates back to the 1960s and the use of PLATO (Programmed Logic for Automatic Teaching Operations)[10] and TICCIT (Time-shared Interactive Computer-Controlled Information Television)[10]. Today, there are numerous online education

[1] Estimate calculated using statistics from the Higher Education Statistics Agency for 2013 – 2014.

applications and software tools available, some of which are commercial and others are freely available, often accessible online through the internet. Various relevant applications used in the United Kingdom will be mentioned later in this chapter.

In this thesis, e-assessment and e-learning will be replaced by the following terms:

Computer-aided assessment (CAA) and **computer-aided learning** (CAL) are terms used to define the assessment and learning practices commonly seen in a classroom setting, but using computers as a means for conducting them. CAL applications aid in a student's learning without necessarily having to assess input data. However, CAA will assess student responses to questions and can provide a lecturer with additional tools for managing and analysing an assessment to better understand the strengths and weaknesses of students.

Some of the advantages and disadvantages to using CAA include the following[11,12]:

<u>Advantages</u>

- Readily available
- Large-scale assessments are easier to organise and manage
- Assessments can include randomised sets of questions based on selected criteria
- Supports different learning environments
- Reporting software can provide detailed feedback about an assessment

<u>Disadvantages</u>

- Limitations with some question types
- Worries over security of data when setting up an invigilated assessment
- Knowing with certainty who is answering the questions
- Restrictions on availability and usability of technology may cause some students to be unable to interact with software
- Use of other software or online applications whilst answering questions

Some question types seen previously as difficult to implement in CAA are now being investigated for possible use. Computers are continually becoming more accessible and new features allow more people to use them with more ease.

However, not all problems have been resolved; for instance, there are still some question types that are difficult to implement. To explain this better, we need to explain the types of questions that can be asked. According to the Teaching for Success National Faculty Success Center[13], an objective question is a question that has clear, correct answers which can be verified upon a simple analysis of the answer, whereas a subjective question is a question which must be scored based on a detailed analysis of the answer using a specified set of criteria. However, this explanation does not clearly define what are “clear, correct answers”, “a simple analysis”, and “a detailed analysis”. Objective questions can be analysed more easily as outcomes are independent of any assessor bias[14]. However, not all objective questions can easily be coded into CAA as the correct answer(s) currently needs to be provided within the question coding (usually as a result of some algorithm implemented at runtime rather than as a pre-determined list).

Example 1.1 *Give an example of a graph of 8 vertices that can be coloured with a maximum of 4 colours.*

The question in Example 1.1 is an objective question. However, correct answers cannot necessarily be pre-set into the question coding as there are no known algorithms for determining adequate solutions.

To better understand objective questions within CAA, we must understand that objectivity occurs in the scoring of answers. Therefore, for the purpose of this research, the following definition will be applied:

Def. 1.1 An **objective question** is a question, which has answers that can be determined using an algorithm and can be automatically marked by the system.[12,15].

For the teacher, it is also important to understand the different types of assessment that can be provided to students:

Def. 1.2 A **formative assessment**[16] is an assessment that analyses the quality of answers and provides detailed feedback regarding the progress of the individual who answered the questions. This type of assessment is usually given during the learning process.

Def. 1.3 A **summative assessment**[16] is an assessment that details the achievement status of the individual answering the questions, usually by means of scoring answers and summing up the scores. This type of assessment is usually given after the learning process has been completed.

Def. 1.4 A **diagnostic assessment**[17] is an assessment in which basic mental capacities are assessed individually to determine an individual's current ability to comprehend the topic material. This type of assessment is usually provided at the beginning of the learning process.

Summative assessments are the easiest to produce using CAA. However, when providing answers to students, it is possible to code detailed feedback for students to see; this provides a measure of formative feedback which will help students to better understand the topic material within the question. Diagnostic assessments can be used formatively or summatively, so CAA can also be used to design a reasonable diagnostic assessment with formative feedback to help the student progress later in their learning. Nonetheless, summative assessments provide scored measures of ability in answering questions correctly, which is important here for conducting the necessary statistical analyses to be used to determine the effectiveness of the questions and to investigate why some questions were easier to answer than others.

The main difference in these assessment types is the timing of the learning that takes place. Diagnostic tests occur to test assumed (or **prerequisite**) knowledge prior to learning new material that builds on this prerequisite knowledge. Formative assessments occur during the learning process. Summative assessments generally occur at the end of the learning process. Diagnostic and formative assessments will have additional learning taking place after these assessments have been conducted, so student learning will likely have some impact on the design of these assessments. Summative assessments, on the other hand, do not need to consider the impact of student learning as a factor in the design of the assessments. This thesis will explore the design of a versatile and robust library of graph theory questions within CAA, which can then be used in the design of assessments; this thesis will not require any knowledge about a student's assumed skills or knowledge, but instead, will focus on the feasibility of a teacher or lecturer using CAA in graph theory in setting functional, user-friendly assessments for students to attempt. Therefore, to fulfil this purpose, summative feedback was used in the statistical research conducted for this thesis to analyse attempted questions in relation to each other and to overall assessments.

1.2.2 Software Applications

There are many CAA applications now available online with which students can practise answering relevant mathematical questions:

Numbas[18] is a free, open-source tool available online. It has a lot of flexibility in how it can be used; it can be used online or offline and tests from Numbas can be uploaded onto various learning platforms. Various styles of media can be added to the design of tests to provide a better structured test to students. It can include interactive graphics using the open-source library, JSXGraph, to provide additional flexibility in engaging with the test topic material.

DEWIS[19] is an e-assessment system used by the University of the West of England (UWE). It has been used extensively by the UWE in different subjects to test mathematical skills. It is well-designed for efficient use by anyone engaging with the software. It was designed intentionally to be independent of commercial software so that modifications can more easily be made. It allows for a detailed analysis of all assessment attempts, which is very useful for pedagogic research.

The System for Teaching and Assessment using a Computer algebra Kernel (STACK)[20] was originally created at Birmingham University and is now a Moodle question type that was developed in partnership with the Open University. It is an open-source system that emphasizes the use of formative assessment by providing detailed, efficient feedback, sometimes suggestive of the answers provided by students. It is visually efficient for students as it will re-design answers in a *mathematically appropriate fashion* (e.g. displaying “$5x^3 - 2x + \frac{1}{x}$” instead of “5x^3-2x+(1/x)”) so that they may then determine if they wish to submit the answers shown by the system. Answers are deemed correct usually if the difference between the student’s answer and the system’s answer is approximately zero, rounded to an appropriate level of accuracy.

Maple T.A.[21] is a public software package which uses the Maple software package. It was developed by Maplesoft, now a subsidiary of Cybernet Systems Co. Ltd. There are additional testing features, which will allow students to answer coordinate geometry and graphing questions by drawing directly onto sets of coordinate axes. It has the added flexibility of asking and assessing any objective question, including the question provided in Example 1.1. It is respectful of mathematical equivalence, implying that it will accept multiple correct answers for the same question.

MyMathLab[22] is a commercial software CAA package available by Pearson Education, Inc. specifically for use in higher education. Similar to Maple T.A., MyMathLab is user-friendly for both the student and the teacher. A selection of courses are available for purchase and teachers can modify assessments within courses in order to have more control over assessment schemes.

WileyPLUS[23] is a commercial software CAA/CAL package with a great emphasis on CAL. Similar to MyMathLab, it has courses that are available for purchase. They also have WileyPLUS with ORION, which is an adaptive, personalised learning system that allows teachers to conduct diagnostic assessments and measure progress through continuous assessment during each course.

Although all of these CAA tools have excellent functionality and visually appealing features, they do not appear to have questions designed around graph theory for available use in CAA. Commercial systems were problematic due to the lack of flexibility in creating one's own coded questions and then using the advanced features to present questions and analyse submitted answers more thoroughly. Some commercial systems also did not make it absolutely clear if their questions included the use of random parameterisation in the design of their questions. Non-commercial systems had provided a lot of additional flexibility, but at the time this research began, very little was known about these systems and many of these features may have been implemented well after research began on this thesis; for instance, Gwynllyw & Henderson[24] note that their consideration into creating DEWIS came after licensing issues with QuestionMark Pereception (QMP); this system will be discussed later.

Of particular interest to this thesis is the work of Ruokokoski[25], who visits multiple subjects within mathematics to investigate the possibility of random parameterisations within CAA questions. Ruokokoski uses STACK to design questions and makes an effort at design some relevant questions in graph theory. One problem noted within Ruokokoski's research is the design of graphs to appear with suitable characteristics and random parameterisation within questions. However, it was also important when preparing this research to understand the relevance of designing the questions themselves. Looking back at the work shown by Ruokokoski[5], some of the questions that were designed do not inform the student of the formatting required to answer the questions correctly. Ruokokoski also seems to focus mostly on graphs when it comes to designing graph theory questions. The visual element of a graph is a key feature to graph theory, but it is not the only feature as adjacency matrices can also be used to define a particular graph or network, although it is mainly numerical in presentation.

maths e.g.[26] is an online databank of CAA/CAL questions that can be used mainly at the postsecondary level with random parameterisation embedded within question codes. Some of these questions included those which were originally created at Brunel University under the title, Mathletics[27], which uses QMP software to facilitate question generation and assessment reporting. A licence was required to operate the QMP software, but some questions from the Mathletics data sets can now be attempted online freely. Similar to the work of Ruokokoski, there were some issues with the visual appearance of graphs, but due to the added flexibility of Mathletics, this can be addressed. QMP provides excellent assessment information, which is especially helpful for the purposes of this particular research. Due to the usefulness and practicality of this software, this research will focus on the use of Mathletics running within QMP to answer the research questions.

1.3 Design of Questions

As it still appears to be the case that the implementation of graph theory within CAA is a relatively new concept, it is important that such a system be designed with versatile and robust questions that can be assessed and provide an organised assessment system for a teacher or lecturer to use. As such, student factors, such as characteristics of individual students, background studies of students, etc., will not be considered within this thesis. However, since this research is within the scope of assessment and learning, it was helpful to explore some basic educational theory in an attempt to better focus the design of questions prior to creating assessments with them. As will be explained later, educational theory will not be considered in the analyses that will occur later, but it has been helpful to have this understanding of question design when preparing graph theory questions for CAA.

1.3.1 Features of QuestionMark Perception

This section looks at the features readily available within Questionmark Perception[28] version 3, which runs Mathletics. Questions that were designed and analysed in this thesis use version 3 of the software rather than the current version 5; this is because issues arose in the latest version with authoring capabilities, which limited the amount of flexibility that was desired in designing suitable questions for topics in graph theory.

Mathletics provides good features for organising databanks of questions, providing more control to teachers and lecturers in setting assessments and analysing student results. Ellis, Greenhow, and Hatt[29] discuss relevant features, some of which are discussed in this section, that bring graph theory questions into the application; additional information about the technical features of Mathletics and the implementation of graph theory into it are noted in the works of Hatt & Greenhow[30] and Hatt[31] and will be detailed, along with screenshots of designed questions, in Chapter 2, Chapter 3, Chapter 4 and Chapter 5.

The management of question design allows question authors to edit questions as needed for their particular assessments. One useful feature is defined as follows:

Def. 1.5 A **random parameter**[32] is an element within a question code that can take on multiple values, as assigned within the question coding.

Randomised parameters are added using Javascript and MathML coding so that different realisations of the same question will appear every time. Random parameters are not built-in, as if through a wizard tool, into QMP, so any random parameters that could be included are done practically and directly into the design and coding of questions. **Scalable vector graphics** (SVG)[33] are also used to bring randomised parameters into any graphs to be displayed. In mathematics, this is especially useful as it can provide individualised testing to students, providing each student with a unique test to complete with identical assessment objectives. Most questions designed for use in graph theory include randomised parameters and SVG, which minimises copying and allows multiple attempts to be made for practising to answer questions.

Keywords can be assigned to questions on the system in the question descriptions to make them easier for teachers and lecturers to find upon searching. Questions can also be tagged to provide additional organisation. All questions designed for use in graph theory are tagged using a perceived difficulty level.

Questions can also be organised in categories and subcategories, depending on subject area, using Question Manager; for instance, a question on Kruskal's algorithm will appear in Decision Mathematics → Minimum Spanning Trees → Kruskal's algorithm[34]. This helps a teacher or lecturer by organising topics so that they are easily searchable when preparing an assessment. Questions relating to graph theory have been included in its own category of Mathletics, as illustrated in Figure 1.1.

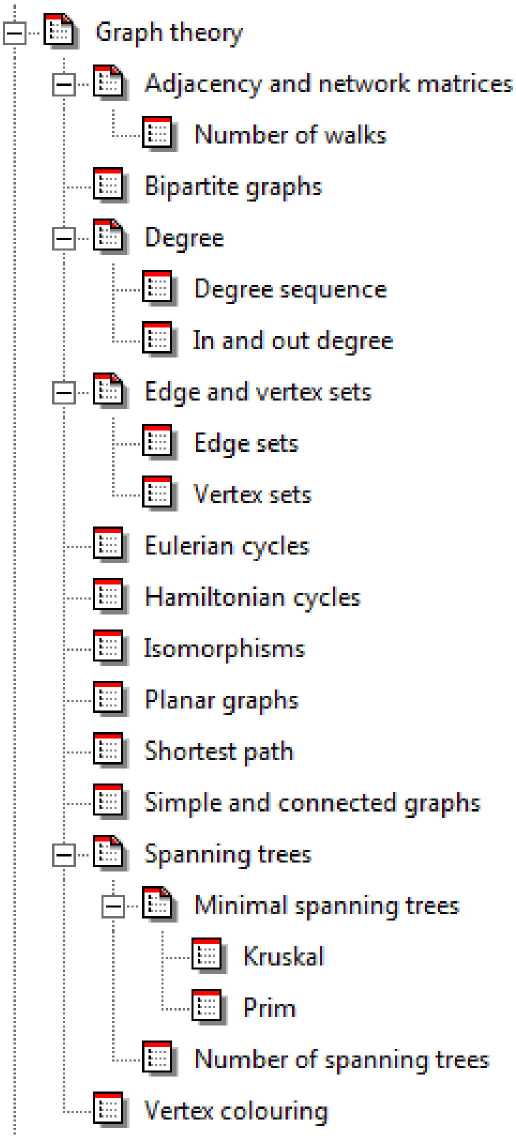


Figure 1.1 Library of topics in the category of Graph theory in Mathletics, updated 2015. Each topic has a series of questions associated with it so that questions can be chosen based on preference.

Question authors have control over each aspect of the question, from its appearance to the assessment and feedback of provided solutions. Different marking schemes can be implemented, including partial marking and negative marking[35]. Feedback can be detailed as needed with randomised parameters and SVG helping to explain all of the relevant information required to answer each question. Assessments can be customised to suit the demands of the teacher or lecturer administering the module and results of assessments provide lots of additional information that can be used to modify future assessments or understand students' abilities better.

For students, there is an accessibility feature[36] in Mathletics, which allows them to view questions on the screen to their preference. This feature is especially important for students with particular difficulties in reading questions effectively based on colour and text font, size, and colour; this can include students with colour blindness or dyslexia[37]. Although these features are useful in the design of the software application, they will not have an impact on the assessments themselves as students are expected to set their text viewing preferences at the beginning of each assessment.

1.3.2 Question Types

The different types of questions that exist within Mathletics are given below. The different types of questions, as will be discussed later, may have some impact on student performance within an assessment.

1.3.2.1 Multiple-Choice Questions

The history of multiple-choice (MC) questions dates back to at least 1913 when Yerkes designed a multiple-choice device to assess the behaviours of animals and humans to form ideas[38]. During World War II, Harrower-Erickson designed a multiple-choice group Rorschach test for screening purposes[39], but this was found unsuitable by the Psychiatric Unit at the U.S. Naval Training Station, Newport, Rhode Island one year later[40].

Much research has since been conducted regarding MC questions. Torres, Lopes, Babo, and Azevedo[41] discuss a strategy for creating useful MC questions in mathematics. They refer to the MATH model shown in Figure 1.2 as a basis for designing their questions and then note some difficulty in writing good distracters. There is some discussion on the homogeneity of distracters in order to avoid guessing the correct answer by a process of elimination. However, this is debatable as distracters can be determined through the understanding of relevant mathematical theory or by viewing students' attempts at solving problems and determining common errors they are making in the process. Common errors may not necessarily be homogeneous to the correct answer, so this needs to be considered further; there will also be considerable focus on linking relevant mathematical theory to the design of distracters. They also mention “None of the above” as a common answer to choose if a student cannot see the answer immediately. However, as will be shown in chapter 3 and chapter 4, the creation of distracters can allow for “None of the above” to be a valid option to MC questions.

Group A	Group B	Group C
Factual knowledge	Information transfer	Justifying and interpreting
Comprehension	Application in new situations	Implications, conjectures, and comparisons
Routine use of procedures		Evaluation

Figure 1.2 MATH (Mathematical Assessment Task Hierarchy) for question design in postsecondary education.

A subcategory of this question type is the **true or false** question, where a statement is given and the correct answer is one of two possibilities, namely True (T) or False (F). An immediate problem with this question type is that there is a 50% probability of answering the question correctly, implying that the question does not necessarily challenge the students’ learning of the subject material within the question. However, Baruah, Gill, and Greenhow[42] investigate this question type by suggesting a **4TFUSP** (4 True, False, or Undecidable; Subject and Property) question, where four different statements are each given with each subject receiving one property that might be associated with it. This question not only brings in another option, namely the Undecidable (U) option, but the assessment of the question creates another challenge in that you may not receive any marks if some of the four randomised answers are incorrect. Cumulative probabilities of answering each statement correctly are provided in Table 1.3. Probabilities are calculated using a **binomial distribution**, where, from a set of n trials, there will be x successes and $n - x$ failures; the probability of x successes is given by the formula,

$$P(X = x) = \binom{n}{x} p^x (1-p)^{n-x}, \text{where } p \text{ is the probability of success.}$$

To challenge students' understanding of graph theory, it is necessary to investigate the use of graphs and adjacency matrices in questions. The 4TFUSP question type could be helpful in investigating students' awareness of properties of graphs, but students need to be able to analyse graphs and adjacency matrices in some detail to answer these questions correctly. Therefore, although 4TFUSP questions may have some usage in the design of additional graph theory questions in the future, this question type was not considered in the design of graph theory questions analysed in this thesis.

Number of correct answers, x	**4**	**3**	**2**	**1**	**0**
Cumulative probability, $P(X \leq x)$	$\frac{81}{81} = 1$	$\frac{80}{81}$	$\frac{72}{81} = \frac{8}{9}$	$\frac{48}{81} = \frac{16}{27}$	$\frac{16}{81}$

Table 1.3 Cumulative probabilities of correctly answering a 4TFUSP question in Mathletics.

1.3.2.2 Numerical Input Questions

Numerical input (NI) questions are perhaps the most common type of mathematics question. These questions ask for a number to be typed into a text box in order to be verified by the system. Many NI questions can be designed by verifying the numerical response given by a student with the answer provided in the question coding.

Some NI questions are responsive numerical input (RNI) questions. RNI questions make use of possible distracters students may trigger upon submitting a solution. If a distracter is triggered, then the feedback will highlight the possible error(s) made by the student as a means of providing additional feedback and warning students about such problems in answering questions.

A subcategory of this question type is the **hotline** question[42], where students are asked to find an error in a line of a detailed answer to a problem. With the use of graphs and adjacency matrices in a higher level subject like graph theory, the solution to a question may require numerous steps in the working and so, there can be numerous lines shown in the original questions.

Also, some of the topics presented investigate defined properties of graphs and so, do not necessarily have algorithmic methods with which to solve related problems. Although it may have some usage in the design of additional graph theory questions in the future, this question type was not considered in the design of graph theory questions analysed in this thesis.

However, with some questions, approximations may need to be considered for accuracy. Some questions in chapter 3 and chapter 4 will investigate this to see how better to design numerical input questions for this scenario.

1.3.2.3 Word Input Questions

Word input (WI) questions are similar to NI questions, but with any text being allowed as input. Unlike NI questions, because text is involved, the formatting of each answer will become an issue in assessing them. WI questions can also modify student answers by removing any unnecessary spaces prior to evaluating submitted answers. Graph theory requires an understanding of edges and vertices, so it is inevitable that WI questions will be necessary for the data set in graph theory. Questions in chapter 3 and chapter 4 which require word input will highlight various issues and how they are resolved in order to ensure students input answers in the correct format.

Similar to NI questions, WI questions can also be responsive (RWI). RWI questions will work in a similar way to RNI questions, but additional distracters could include the formatting of answers so as to remind students to double-check their work prior to submitting it. WI and RWI questions can also include pop-up windows, as shown in Figure 5.2. These pop-up windows are helpful in reminding students to double-check their work before resubmitting their answers. If their answers do not change when they click Submit a second time, then their answers are analysed; if their answers are changed, then the pop-up window will appear again and this will continue until a submitted answer is identical to a previously submitted answer.

1.4 Analysis of Questions

To effectively analyse graph theory questions designed using Mathletics, two key values will be analysed:

Def. 1.6 **Facility** of a question is the mean value awarded to all students who attempted the question.

This is the main value to be analysed for the purposes of this thesis as it will assess the difficulty of each question, which will be helpful in analysing characteristic differences between related questions.

Def. 1.7 (**Index of**) **Discrimination** of a question is the correlation between each student's whole score for the question and the total score awarded for the whole assessment.

This value will also be analysed within this thesis as it will help to determine the reliability of the design of each question within an assessment. Since correlated values range from -1 to 1, any negative or low non-negative values will suggest the question may need to be improved for future assessments.

The statistical analyses of these values are discussed in the methodology in chapter 5.

1.5 Research Questions and Hypothesis

The objective of this thesis is to determine if an online assessment tool can be used to design questions with random parameterisation for use in graph theory and, if so, then also determine what within these questions can cause them to be more difficult than other questions in the same field of mathematics.

To complete this thesis, these research questions need to be answered:

1. How can the potential of computer-aided assessment be exploited to set versatile and robust questions in graph theory?

In chapter 2, technical features of Mathletics will highlight the capabilities of the software that have been included to accommodate suitable questioning in graph theory. A library of 49 questions, 14 belonging to introductory graph theory and 35 belonging to intermediate / advanced graph theory, have been prepared in Mathletics. Chapter 3 and chapter 4 will look at the design of graph theory questions within Mathletics, along with all of its particular features, to demonstrate its capabilities.

2. What question features exist that could change how students interact with questions?

   All of the questions designed in Mathletics for use in graph theory will be detailed. To do this, a description of each topic will be given, followed by a description of the designed questions, their functionality, question appearance, and feedback provided. The design of questions in chapter 3 and chapter 4 will highlight different question characteristics, such as question type, use of a graphs or adjacency matrices, and question style (e.g. mathematical problem or word problem in context). This information will form a basis for setting up the methodology of the statistical analyses in chapter 5.

3. Which factors, if any, can cause an objective question in graph theory to be more difficult than other questions in the same topic?

   This will investigate the features found earlier to determine if comparable features have any significant impact on the answers students give to similar questions. The statistical analyses in chapters 6 and 7 will investigate different possible factors to determine if factors exist which can cause some questions to become more difficult to answer correctly.

   The first analysis, conducted in 2008, involved students from the Brunel University mathematics module, MA2920: *Algebra and Discrete*

Mathematics[43]. Students completed two sets of practice tests, namely a “visual test” using graphs and a “logical test” using adjacency matrices, prior to sitting an invigilated test, which combined graphs and adjacency matrices in each question. The analysis will determine what potential these questions have in the assessment of graph theory, but they will also highlight any patterns that may cause a significant change in overall assessment scores.

The second analysis, conducted in 2015, involves different cohorts of students who sat the Brunel University mathematics module, MA0422: *Discrete and Decision Mathematics*[6] from 2008 – 2014. Due to a change from 2011, in which caused topics to be removed, added, or edited significantly, data has been grouped separately for 2008 – 2011 and for 2011 – 2014. This analysis will determine if there are particular characteristics that cause one question to be more difficult to answer than another question. As there are multiple questions in each topic, an analysis within each topic will explore possible issues; however, it is also important to analyse topics together as some topics may be perceived as being “harder” to answer than others.

The assessments conducted throughout are performed so that the best result out of five attempts in one invigilated test session is recorded as the student’s final result for the assessment. In each assessment, upon submitting their answers, students received immediate feedback about their answers and how to solve problems correctly if they answered incorrectly. This structuring of assessments provides summative and formative feedback to students during the assessment.

Note that since 2008, additional work from Zaczek[2,44] has updated some of the questions already prepared for graph theory and have been used in this analysis. Questions on spanning tree algorithms

[2] It is unfortunate that Zaczek’s work does not reference previous work completed by Hatt[30,31,35] on graph theory and Mathletics. However, Zaczek’s work does follow from these previous works. Questions about Prim’s and Kruskal’s algorithms are new to Mathletics since 2008, so these will be referenced later in Section 3.10.

specifically focus on the use of **Prim’s algorithm** and **Kruskal’s algorithm** and will also be included in this analysis.

The hypothesis is that although online, objective learning and assessment can be beneficial for use within discrete mathematics and especially graph theory, some issues will emerge, e.g. visual components versus numerical components, which will cause students to have difficulty answering similar questions, which could have an impact on the future of question design in graph theory. It also needs to be emphasized that this thesis focuses on the efficacy of the questions themselves within Mathletics and CAA and so, will not make significance of any factors relating to students, their learning environments, or their progress; however, considerations focusing on these factors will be discussed later for possible future research.

1.6 A Note About The References

Some of the references featured will come from alternative sources as opposed to traditional “textbook” resources. While efforts have been made to minimise the amount of alternative references, some still remain as they were still seen as being helpful in understanding some of the details relevant to this research and therefore, these references must still be noted.

Chapter 2 Design of Template Codes for Graph Theory in Mathletics

2.1 Question and Template Design: An Introduction

QuestionMark Perception allows question designers to create robust questions with random parameters and scalable vector graphics (SVG) that allow for a better presentation of questions to appear on the screen. In the case of graph theory, it is important that SVG appears in questions when possible. The creation of graphs involves the addition of edges and vertices, so these features need to be included as part of the programming behind the scenes. Additional programming will be required for graphs that need to include special properties, such as Eulerian graphs, planar graphs, and graphs used in vertex colouring. Some additional features on top of this may also be required, such as loops, values for weighted edges, arrows for directed edges, etc. Additional considerations can also be provided within QuestionMark Perception so that the graph can be easily seen in proportion to font sizes, colours, etc., according to each student's preferences. Having one programmed code to design a graph of n vertices with selected edges and all of these features to be included when necessary would be exceptionally helpful as the general structure of the graph remains intact whilst adding features through the use of function variables. However, creating a programming code for a graph can use a lot of character space and unfortunately, each question code is limited to 32,000 characters[45]. Templates can be called from questions, though, so that character use is reduced, so it is advantageous to use this approach when designing questions. To do this, not only does the graph need to be created, but so does the template.

Creating the template is simple, but any useful functions for use in graph theory need to be included in the template. The use of labelling can easily call algebraic programming functions; in addition, questions involving vertex colouring can explore algebraic functions. Therefore, the linear algebra template will often be used. However, including a significantly large code for graphs in the linear algebra template will cause any questions using this template to take much longer to load since templates scan through all of the codes in order to use those that are being called. Alternatively, templates can be combined and used as a “new” template for those questions requiring both linear algebra and graph theory functions; this is not so difficult as little memory is typically used in template files.

The following sections will highlight and detail the key codes created for use in Mathletics. Notes about key portions of the code are provided in **bold** print.

2.2 Simple Network

The first code in the template shows the code for a simple network. The network is designed by fixing a random number of points, as chosen by the question designer, so that the points are equally spaced around a circle. Smaller circles are created, with each point acting as the centre for each circle, so that these points are made visible as vertices in the graph. Edges are created between sets of two points as required by the question designer. Arrows are added if directed edges are required and weights are also added where weighted networks are required; arrows appear as triangles along an edge and weights appear slightly off an edge so that values of weights may be legible.

```
function
SVG_network(A,weights,arrow_ratio_along_line,filled,weights_ratio_along_line,we
ights_font_colour,weights_font_scale){NRC = MatrixSize(A);
var n = NRC[0];
var fs = getFontSize()/16;
if(weights_font_colour == null){weights_font_colour = getFgColor();
}
if(weights_font_scale == null){weights_font_scale = fs;}
size1 = fs*800; size2 = fs*800; size3 = fs*(100*fs*fs+410*fs+250); size4 =
fs*(100*fs*fs+410*fs+200);
   svg_start = '<iSvg:svg height="'+size1+'" width="'+size2+'" viewBox="0 0
'+size3+' '+size4+'"><iSvg:g id="canvas">';
   svg_end = '<iSvg:g></iSvg:svg>';
SVG_graph = "";
r = getFontSize()/16*200; rrloop = getFontSize()/16*50; rloop = r + rrloop;
offset = r+getFontSize()/16*200; offset2 = r+getFontSize()/16*75;
```

```
x_coord = new Array(n); y_coord = new Array(n);
x_coord_loop = new Array(n); y_coord_loop = new Array(n);
x_coord_label = new Array(n); y_coord_label = new Array(n);
for(k = 0; k <= n-1; k++){
  x_coord[k] = r*Math.cos(2*k*Math.PI/n)+offset;
  y_coord[k] = r*Math.sin(2*k*Math.PI/n)+offset;
  x_coord_label[k] = (r+offset2)/2*Math.cos(2*k*Math.PI/n)+(offset);
  y_coord_label[k] = (r+offset2)/2*Math.sin(2*k*Math.PI/n)+offset;
  x_coord_loop[k] = rloop*Math.cos(2*k*Math.PI/n)+offset;
  y_coord_loop[k] = rloop*Math.sin(2*k*Math.PI/n)+offset;}
var colour = getFgColor();
for(i = 1; i <= n; i++) {
  SVG_graph += SVG_ellipsebl(x_coord[i-1],y_coord[i-
1],0.1,0.1)+SVG_scale_text(x_coord_label[i-1],y_coord_label[i-1],alphabet(i-
1,1),colour,fs);
     for(j = 1; j <= n; j++){
        if(A[i][j] == 1){
if(arrow_ratio_along_line != 0){
SVG_graph += SVG_arrow(x_coord[i-1],y_coord[i-1],x_coord[j-1],y_coord[j-
1],arrow_ratio_along_line,filled);}
else{
SVG_graph += SVG_line(x_coord[i-1],y_coord[i-1],x_coord[j-1],y_coord[j-1])};
     }
}
```

The next portion of the code gives loops on a single vertex when required:

```
        if(A[i][i] != 0){SVG_graph += SVG_ellipse(x_coord_loop[i-1],y_coord_loop[i-
1],rrloop,rrloop);
}}
```

The last portion of the code adds network weights so that they are always on top and legible.

```
for(i = 1; i <= n; i++) {for(j = 1; j <= n; j++){ if(A[i][j] != 0){
x1 = x_coord[i-1]; x2 = x_coord[j-1]; y1 = y_coord[i-1]; y2 = y_coord[j-1];
var length_of_line = Math.pow((x1-x2)*(x1-x2)+(y1-y2)*(y1-y2),0.5);
var angle_of_line = angle_from_xy(x2-x1,y2-y1);
var xtext = x1*(1-weights_ratio_along_line)+x2*weights_ratio_along_line;
var ytext = y1*(1-weights_ratio_along_line)+y2*weights_ratio_along_line;
if(weights_ratio_along_line != 0){SVG_graph +=
SVG_scale_text(xtext,ytext,weights[i][j],weights_font_colour,weights_font_scale*fs
)};}}}
return svg_start + SVG_graph + svg_end;}
```

Code 2.1 The code for an undirected and unlabelled network.

2.3 Digraphs

The next code in the template shows the code for a digraph. This code is very similar to the network code, but with fewer variables to be considered; this may be helpful for question designers who specifically want to draw digraphs rather than detailed networks with some directed edges.

```
function
SVG_digraph(A,ratio_along_line,filled,double_path_colour,double_path_skinnyne
ss,svg_start){
if(double_path_skinnyness == null){double_path_skinnyness = 10};
NRC = MatrixSize(A); var n = NRC[0];
var fs = getFontSize()/16; size1 = fs*800; size2 = fs*800; size3 =
fs*(100*fs*fs+410*fs+250); size4 = fs*(100*fs*fs+410*fs+200);

if(svg_start == null){svg_start = '<iSvg:svg height="'+size1+'" width="'+size2+'"
viewBox="0 0 '+size3+' '+size4+'"><iSvg:g id="canvas">';} svg_end =
'<iSvg:g></iSvg:svg>'; SVG_graph = "";
r = fs*200; rrloop = fs*50; rloop = r + rrloop offset = r+fs*200; offset2 = r+fs*75;
x_coord = new Array(n); y_coord = new Array(n);
x_coord_loop = new Array(n); y_coord_loop = new Array(n);
x_coord_label = new Array(n); y_coord_label = new Array(n);
for(k = 0; k <= n-1; k++){
  x_coord[k] = r*Math.cos(2*k*Math.PI/n)+offset;
  y_coord[k] = r*Math.sin(2*k*Math.PI/n)+offset;
  x_coord_label[k] = (r+offset2)/2*Math.cos(2*k*Math.PI/n)+(offset);
  y_coord_label[k] = (r+offset2)/2*Math.sin(2*k*Math.PI/n)+(offset);
  x_coord_loop[k] = rloop*Math.cos(2*k*Math.PI/n)+offset;
  y_coord_loop[k] = rloop*Math.sin(2*k*Math.PI/n)+offset;}
colour = getFgColor();
for(i = 1; i <= n; i++) {SVG_graph += SVG_ellipsebl(x_coord[i-1],y_coord[i-
1],0.1,0.1)+SVG_scale_text(x_coord_label[i-1],y_coord_label[i-1],alphabet(i-
1,1),colour,fs);
     for(j = 1; j <= n; j++){if(A[i][j] == 1){
if(ratio_along_line != 0){SVG_graph += SVG_arrow(x_coord[i-1],y_coord[i-
1],x_coord[j-1],y_coord[j-1],ratio_along_line,filled)}else{
SVG_graph += SVG_line(x_coord[i-1],y_coord[i-1],x_coord[j-1],y_coord[j-1])};}
  if(A[i][j] == 2){
cxx = (x_coord[i-1]+x_coord[j-1])/2; cyy = (y_coord[i-1]+y_coord[j-1])/2;
dx = (x_coord[i-1]-x_coord[j-1]); dy = (y_coord[i-1]-y_coord[j-1]);
if(dx == 0){theta = 90}else{theta = Math.atan(dy/dx)*180/Math.PI};
rxx = Math.pow((x_coord[i-1]-x_coord[j-1])*(x_coord[i-1]-x_coord[j-1])+(y_coord[i-
1]-y_coord[j-1])*(y_coord[i-1]-y_coord[j-1]),0.5)/2;
ryy = rxx/double_path_skinnyness; // gives a skinny ellipse
SVG_graph += SVG_ellipse_rotate(cxx,cyy,rxx,ryy,theta,double_path_colour);}}
```

The next portion of the code gives loops on a single vertex when required:

```
        if(A[i][i] != 0){SVG_graph += SVG_ellipse(x_coord_loop[i-1],y_coord_loop[i-
1],rrloop,rrloop);}}

return svg_start + SVG_graph + svg_end;}
```

Code 2.2 The code for a digraph.

2.4 Labelled Digraph

The next code in the template shows the code for a digraph with labels included.

```
function
SVG_digraph_label(A,labels,ratio_along_line,filled,double_path_colour,double_pa
th_skinnyness,svg_start){
if(double_path_skinnyness == null){double_path_skinnyness = 10};
NRC = MatrixSize(A);
var n = NRC[0]; var fs = getFontSize()/16;
size1 = fs*800; size2 = fs*800; size3 = fs*(100*fs*fs+410*fs+250); size4 =
fs*(100*fs*fs+410*fs+200);
if(svg_start == null){svg_start = '<iSvg:svg height="'+size1+'" width="'+size2+'"
viewBox="0 0 '+size3+' '+size4+'"><iSvg:g id="canvas">';}
   svg_end = '<iSvg:g></iSvg:svg>';
SVG_graph = "";
r = getFontSize()/16*150; rrloop = getFontSize()/16*50; rloop = r + rrloop
offset = r+getFontSize()/16*200; offset2 = r+getFontSize()/16*75;
x_coord = new Array(n); y_coord = new Array(n);
x_coord_loop = new Array(n); y_coord_loop = new Array(n);
x_coord_label = new Array(n); y_coord_label = new Array(n);
for(k = 0; k <= n-1; k++){
  x_coord[k] = r*Math.cos(2*k*Math.PI/n)+offset;
  y_coord[k] = r*Math.sin(2*k*Math.PI/n)+offset;
  x_coord_label[k] = ((1.4*r)+offset2)/2*Math.cos(2*k*Math.PI/n)+(0.9*offset);
  y_coord_label[k] = ((1.4*r)+offset2)/2*Math.sin(2*k*Math.PI/n)+(offset);
  x_coord_loop[k] = rloop*Math.cos(2*k*Math.PI/n)+offset;
  y_coord_loop[k] = rloop*Math.sin(2*k*Math.PI/n)+offset;}
colour = getFgColor();
for(i = 1; i <= n; i++) {
  SVG_graph += SVG_ellipsebl(x_coord[i-1],y_coord[i-
1],0.1,0.1)+SVG_scale_text(x_coord_label[i-1],y_coord_label[i-
1],labels[i],colour,fs);
     for(j = 1; j <= n; j++){if(A[i][j] == 1){
if(ratio_along_line != 0){SVG_graph += SVG_arrow(x_coord[i-1],y_coord[i-
1],x_coord[j-1],y_coord[j-1],ratio_along_line,filled)}else{
SVG_graph += SVG_line(x_coord[i-1],y_coord[i-1],x_coord[j-1],y_coord[j-1])};
```

```
    }
  if(A[i][j] == 2){
cxx = (x_coord[i-1]+x_coord[j-1])/2; cyy = (y_coord[i-1]+y_coord[j-1])/2;
dx = (x_coord[i-1]-x_coord[j-1]); dy = (y_coord[i-1]-y_coord[j-1]);
if(dx == 0){theta = 90}else{theta = Math.atan(dy/dx)*180/Math.PI};
rxx = Math.pow((x_coord[i-1]-x_coord[j-1])*(x_coord[i-1]-x_coord[j-1])+(y_coord[i-1]-y_coord[j-1])*(y_coord[i-1]-y_coord[j-1]),0.5)/2;
ryy = rxx/double_path_skinnyness;
```

The portion of the code appearing above provides a "skinny" ellipse; such an ellipse is more presentable on screen for the purposes we require than a wider, shorter ellipse.

```
SVG_graph += SVG_ellipse_rotate(cxx,cyy,rxx,ryy,theta,double_path_colour);}}
```

The next portion of the code gives loops on a single vertex when required:

```
      if(A[i][i] != 0){SVG_graph += SVG_ellipse(x_coord_loop[i-1],y_coord_loop[i-1],rrloop,rrloop);}}


return svg_start + SVG_graph + svg_end;}
```

Code 2.3 The code for a labelled digraph.

2.5 Vertex Colouring

The next code shows the code for a graph that is used in vertex colouring. A similar coding to the network coding above is used, where circles are created to show visible vertices, but instead of colouring them in one colour, a variety of colours can be chosen; of course, for the purposes of vertex colouring, colours need to be specifically chosen, so that is considered in the coding below.

```
//function
SVG_digraph_label_colours(A,labels,ratio_along_line,filled,double_path_colour,vertex_colours,numeric_order,double_path_skinnyness,vertex_radius,svg_start)
```

If labels == 0, then an alphabetical order is given to the labels. If you want large, coloured vertices to appear, then set vertex_radius = 10. Otherwise, set vertex_radius = 0.1.

```
function
SVG_digraph_label_colours(A,labels,ratio_along_line,filled,double_path_colour,ve
rtex_colours,numeric_order,double_path_skinnyness,vertex_radius,svg_start){
if(double_path_skinnyness == null){double_path_skinnyness = 10};
NRC = MatrixSize(A);
var n = NRC[0];
substituted_labels = new Array();
if(labels == 0){
  for(i = 1; i <= n; i++){
     substituted_labels[i] = alphabet(i-1,1);
  }
}else{
  for(i = 1; i <= n; i++){
     substituted_labels[i] = labels[i];
  }
}
var fs = getFontSize()/16;
size1 = fs*800;
size2 = fs*800;
size3 = fs*(100*fs*fs+410*fs+250);
size4 = fs*(100*fs*fs+410*fs+200);
if(svg_start == null){svg_start = '<iSvg:svg height="'+size1+'" width="'+size2+'"
viewBox="0 0 '+size3+' '+size4+'"><iSvg:g id="canvas">';}
   svg_end = '<iSvg:g></iSvg:svg>';
SVG_graph = "";
r = getFontSize()/16*150;
rrloop = getFontSize()/16*50;
rloop = r + rrloop;
offset = r+getFontSize()/16*200;
offset2 = r+getFontSize()/16*75;
x_coord = new Array(n);
y_coord = new Array(n);
x_coord_loop = new Array(n);
y_coord_loop = new Array(n);
x_coord_label = new Array(n);
y_coord_label = new Array(n);
for(k = 0; k <= n-1; k++){
  x_coord[k] = r*Math.cos(2*k*Math.PI/n)+offset;
  y_coord[k] = r*Math.sin(2*k*Math.PI/n)+offset;
  x_coord_label[k] = ((1.4*r)+offset2)/2*Math.cos(2*k*Math.PI/n)+(0.9*offset);
  y_coord_label[k] = ((1.4*r)+offset2)/2*Math.sin(2*k*Math.PI/n)+(offset);
  x_coord_loop[k] = rloop*Math.cos(2*k*Math.PI/n)+offset;
  y_coord_loop[k] = rloop*Math.sin(2*k*Math.PI/n)+offset;
}
colour = getFgColor();
for(i = 1; i <= n; i++) {

     for(j = 1; j <= n; j++){
        if(A[i][j] == 1){
```

```
if(ratio_along_line != 0){SVG_graph += SVG_arrow(x_coord[i-1],y_coord[i-
1],x_coord[j-1],y_coord[j-1],ratio_along_line,filled)}else{
SVG_graph += SVG_line(x_coord[i-1],y_coord[i-1],x_coord[j-1],y_coord[j-1])};
    }
  if(A[i][j] == 2){
cxx = (x_coord[i-1]+x_coord[j-1])/2;
cyy = (y_coord[i-1]+y_coord[j-1])/2;
dx = (x_coord[i-1]-x_coord[j-1]);
dy = (y_coord[i-1]-y_coord[j-1]);
if(dx == 0){theta = 90}else{theta = Math.atan(dy/dx)*180/Math.PI};
rxx = Math.pow((x_coord[i-1]-x_coord[j-1])*(x_coord[i-1]-x_coord[j-1])+(y_coord[i-
1]-y_coord[j-1])*(y_coord[i-1]-y_coord[j-1]),0.5)/2;
ryy = rxx/double_path_skinnyness;
```

The portion of the code appearing above provides a "skinny" ellipse; such an ellipse is more presentable on screen for the purposes we require than a wider, shorter ellipse.

```
SVG_graph += SVG_ellipse_rotate(cxx,cyy,rxx,ryy,theta,double_path_colour);
    }
  }
```

The next portion of the code gives loops on a single vertex when required:

```
        if(A[i][i] != 0){
          SVG_graph += SVG_ellipse(x_coord_loop[i-1],y_coord_loop[i-
1],rrloop,rrloop);
    }
}
for(i = 1; i <= n; i++) {
SVG_graph += SVG_ellipseblx(x_coord[i-1],y_coord[i-
1],vertex_radius,vertex_radius,vertex_colours[(numeric_order[i-1]-0)-
1])+SVG_scale_text(x_coord_label[i-1],y_coord_label[i-
1],substituted_labels[i],colour,fs);
}
return svg_start + SVG_graph + svg_end;
          }
```

Code 2.4 The code for vertex colouring a graph.

2.6 Wheel Graphs

The next code in the template shows the code for a wheel graph. This graph looks at the points formed in a circle, but the centre point of that circle is included as an additional point. All points formed around the circle must join the centre point and neighbouring points in the circle in order to create a proper wheel graph.

```
function
SVG_wheelgraph_label(rows,labels,ratio_along_line,filled,double_path_colour,dou
ble_path_skinnyness,svg_start){

A = getrandommatrix(rows, rows, 0, 0, 1);
Dis = getrandommatrix(rows, rows, 0, 0, 1);
for (i = 1; i <= rows-2; i++){
  A[i][i+1] = (A[i][i+1]-0)+1;
}
A[rows-1][1] = (A[rows-1][1]-0)+1;
for(i = 1; i <= rows-1; i++){
 A[i][rows] = (A[i][rows]-0)+1;
}
for(i = 1; i <= rows; i++){
  for(j = i; j <= rows; j++){
     Dis[i][j] = Math.max((A[i][j]-0),(A[j][i]-0));
     Dis[j][i] = Math.max((A[i][j]-0),(A[j][i]-0));
  }
}

if(double_path_skinnyness == null){double_path_skinnyness = 10};
NRC = MatrixSize(Dis);
var n1 = NRC[0] - 1;
var n = NRC[0];
var fs = getFontSize()/16;
size1 = fs*1200;
size2 = fs*1200;
size3 = fs*(100*fs*fs+410*fs+250);
size4 = fs*(100*fs*fs+410*fs+200);
if(svg_start == null){svg_start = '<iSvg:svg height="'+size1+'" width="'+size2+'"
viewBox="0 0 '+size3+' '+size4+'"><iSvg:g id="canvas">';}
   svg_end = '<iSvg:g></iSvg:svg>';
SVG_graph = "";
r = getFontSize()/16*150;
rrloop = getFontSize()/16*50;
rloop = r + rrloop
offset = r+getFontSize()/16*150;
offset2 = r+getFontSize()/16*75;
x_coord = new Array(n);
y_coord = new Array(n);
x_coord_loop = new Array(n);
y_coord_loop = new Array(n);
x_coord_label = new Array(n);
y_coord_label = new Array(n);
```

```
for(k = 0; k <= n1-1; k++){
  x_coord[k] = r*Math.cos(2*k*Math.PI/n1)+offset;
  y_coord[k] = r*Math.sin(2*k*Math.PI/n1)+offset;
  x_coord_label[k] = ((1.4*r)+offset2)/2*Math.cos(2*k*Math.PI/n1)+(0.85*offset);
  y_coord_label[k] = ((1.4*r)+offset2)/2*Math.sin(2*k*Math.PI/n1)+(offset);
  x_coord_loop[k] = rloop*Math.cos(2*k*Math.PI/n1)+offset;
  y_coord_loop[k] = rloop*Math.sin(2*k*Math.PI/n1)+offset;
}
  x_coord[n-1] = offset;
  y_coord[n-1] = offset;
  x_coord_label[n-1] = (r/5)+offset;
  y_coord_label[n-1] = (r/5)*Math.sin(2*Math.PI/n1)+offset;
  x_coord_loop[n-1] = offset;
  y_coord_loop[n-1] = offset;
colour = getFgColor();
//SVG_graph += SVG_ellipsebl(x_coord[n-1],y_coord[n-
1],0.1,0.1)+SVG_scale_text(x_coord_label[n-1],y_coord_label[n-1],"",colour,fs);
for(i = 1; i <= n; i++) {
  SVG_graph += SVG_ellipsebl(x_coord[i-1],y_coord[i-
1],0.1,0.1)+SVG_scale_text(x_coord_label[i-1],y_coord_label[i-1],labels[i-
1],colour,fs);
    for(j = 1; j <= n; j++){
      if(A[i][j] == 1){
if(ratio_along_line != 0){SVG_graph += SVG_arrow(x_coord[i-1],y_coord[i-
1],x_coord[j-1],y_coord[j-1],ratio_along_line,filled)}else{
SVG_graph += SVG_line(x_coord[i-1],y_coord[i-1],x_coord[j-1],y_coord[j-1])};}
  if(A[i][j] == 2){
cxx = (x_coord[i-1]+x_coord[j-1])/2;
cyy = (y_coord[i-1]+y_coord[j-1])/2;
dx = (x_coord[i-1]-x_coord[j-1]);
dy = (y_coord[i-1]-y_coord[j-1]);
if(dx == 0){theta = 90}else{theta = Math.atan(dy/dx)*180/Math.PI};
rxx = Math.pow((x_coord[i-1]-x_coord[j-1])*(x_coord[i-1]-x_coord[j-1])+(y_coord[i-
1]-y_coord[j-1])*(y_coord[i-1]-y_coord[j-1]),0.5)/2;
ryy = rxx/double_path_skinnyness; // gives a skinny ellipse
SVG_graph += SVG_ellipse_rotate(cxx,cyy,rxx,ryy,theta,double_path_colour);}}
```

The next portion of the code gives loops on a single vertex when required:

```
      if(A[i][i] != 0){
        SVG_graph += SVG_ellipse(x_coord_loop[i-1],y_coord_loop[i-
1],rrloop,rrloop);
    }}

return svg_start + SVG_graph + svg_end;
        }
```

Code 2.5 The code for a wheel graph.

2.7 Ladder Graphs

The next code in the template shows the code for a ladder graph. Since all ladder graphs have an even number of vertices (i.e. in order for a proper ladder shape to be formed), the *laddersize* variable is the "height" of the ladder (i.e. how many vertices high the ladder will go to be formed). The ladder shape is formed in the code below, meaning that the "circular" pattern used in the previous codes is not considered here.

```
function SVG_laddergraph(laddersize,ratio_along_line,filled,svg_start){
double = 2*laddersize;
A = getrandommatrix(double, double, 0, 0, 1);
Dis = getrandommatrix(double, double, 0, 0, 1);
for(i = 1; i <= laddersize - 1; i++){
  A[i][i+1] = 1;
}
for(i = 1; i <= laddersize; i++){
  A[i][i+laddersize] = 1;
}
for(i = laddersize+1; i <= double - 1; i++){
  A[i][i+1] = 1;
}
# alert(A);

for(i = 1; i <= double; i++){
  for(j = i; j <= double; j++){
     Dis[i][j] = Math.max((A[i][j]-0),(A[j][i]-0));
     Dis[j][i] = Math.max((A[i][j]-0),(A[j][i]-0));
  }
}
fs = getFontSize();
size1 = fs/16*1000;
size2 = fs/16*1200;
size3 = fs/16*1200;
size4 = fs/16*1400;
if(svg_start == null){svg_start = '<iSvg:svg height="'+size1+'" width="'+size2+'"
viewBox="0 0 '+size3+' '+size4+'"><iSvg:g id="canvas">';}
   svg_end = '<iSvg:g></iSvg:svg>';
SVG_graph = "";
r = getFontSize()/16*150;
offset = r+getFontSize()/16*100;
offset2 = -r+getFontSize()/16*75;
label_shift = getFontSize()/16*5;
```

This portion of the code details the placement of vertices in the "ladder shape".

```
x_coord = new Array(n);
y_coord = new Array(n);
x_coord_label = new Array(n);
y_coord_label = new Array(n);
for(k = 0; k <= laddersize-1; k++){
  x_coord[k] = (r*k)-(2*offset2);
  y_coord[k] = offset;
  x_coord_label[k] = x_coord[k];
  y_coord_label[k] = offset-label_shift;}
for(k = laddersize; k <= double-1; k++){
  x_coord[k] = (r*(k-laddersize))-(2*offset2);
  y_coord[k] = 2*offset;
  x_coord_label[k] = x_coord[k];
  y_coord_label[k] = 2*offset+4*label_shift;}
colour = getFgColor();
for(i = 1; i <= double; i++) {
  SVG_graph += SVG_ellipsebl(x_coord[i-1],y_coord[i-
1],0.1,0.1)+SVG_scale_text(x_coord_label[i-1],y_coord_label[i-
1],labels[i],colour,fs/16);
 for(j = 1; j <= double; j++){
        if(A[i][j] == 1){
           if(ratio_along_line != 0){
              SVG_graph += SVG_arrow(x_coord[i-1],y_coord[i-1],x_coord[j-
1],y_coord[j-1],ratio_along_line,filled)
           }else
{
              SVG_graph += SVG_line(x_coord[i-1],y_coord[i-1],x_coord[j-1],y_coord[j-
1])
};
           }
   }
}
return svg_start + SVG_graph + svg_end;
           }
```

Code 2.6 The code for a ladder graph.

2.8 Shortest Path Graphs

The last code in the template shows the code for a graph used in shortest path problems. In these problems, ordering of vertices needs to go from left to right to help illustrate the movement from a starting vertex to a terminating vertex. The code below achieves this using negatives and is detailed below.

SVG_shortest_path(A,labels,ratio_along_line,filled,double_path_colour,vertex_colours,numeric_order,double_path_skinnyness,vertex_radius,svg_start)

If labels == 0, then alphabetical order is given to the labels.

If you want large, coloured vertices to appear, then set vertex_radius = 10; otherwise, set vertex_radius = 0.1.

If numeric_order == 0, then normal increasing ordering occurs; otherwise, an array is required for numeric_order.

```
function
SVG_shortest_path(A,labels,ratio_along_line,filled,double_path_colour,vertex_colours,numeric_order,double_path_skinnyness,vertex_radius,svg_start){
if(double_path_skinnyness == null){double_path_skinnyness = 10};
NRC = MatrixSize(A);
var n = NRC[0];
substituted_labels = new Array();
if(labels == 0){
  for(i = 1; i <= n; i++){
     substituted_labels[i] = alphabet(i-1,1);
  }
}else{
  for(i = 1; i <= n; i++){
     substituted_labels[i] = labels[i];
  }
}
var fs = getFontSize()/16;
size1 = fs*800;
size2 = fs*800;
size3 = fs*(100*fs*fs+410*fs+250);
size4 = fs*(100*fs*fs+410*fs+200);
if(svg_start == null){svg_start = '<iSvg:svg height="'+size1+'" width="'+size2+'"
viewBox="0 0 '+size3+' '+size4+'"><iSvg:g id="canvas">';}
   svg_end = '<iSvg:g></iSvg:svg>';
SVG_graph = "";
r = getFontSize()/16*150;
rrloop = getFontSize()/16*50;
rloop = r + rrloop;
offset = r+getFontSize()/16*200;
offset2 = r+getFontSize()/16*75;
```

This section of the code details the placement of vertices. For this particular type of function, it is important that vertices appear in a left-to-right formation to help illustrate the path from a starting vertex to a terminating vertex. Negatives are included in the x-coordinates to begin on the left-hand side, but have also been included in the y-coordinates, as powers of -1, in order to use vertices above and below as ordering moves from left to right.

```
x_coord = new Array(n);
y_coord = new Array(n);
x_coord_loop = new Array(n);
y_coord_loop = new Array(n);
x_coord_label = new Array(n);
y_coord_label = new Array(n);
for(k = 0; k <= n-1; k++){
  x_coord[k] = (-1)*r*Math.cos(2*Math.ceil(k/2)*Math.PI/n)+offset;
  y_coord[k] = Math.pow(-1,k)*r*Math.sin(2*Math.ceil(k/2)*Math.PI/n)+offset;
  x_coord_label[k] = (-
1)*((1.3*r)+offset2)/2*Math.cos(2*Math.ceil(k/2)*Math.PI/n)+(0.98*offset);
  y_coord_label[k] = Math.pow(-
1,k)*((1.3*r)+offset2)/2*Math.sin(2*Math.ceil(k/2)*Math.PI/n)+(offset);
  x_coord_loop[k] = (-1)*rloop*Math.cos(2*Math.ceil(k/2)*Math.PI/n)+offset;
  y_coord_loop[k] = Math.pow(-
1,k)*rloop*Math.sin(2*Math.ceil(k/2)*Math.PI/n)+offset;
}

colour = getFgColor();
for(i = 1; i <= n; i++) {
  if(numeric_order == 0){
     SVG_graph += SVG_ellipseblx(x_coord[i-1],y_coord[i-
1],vertex_radius,vertex_radius,vertex_colours[i-
1])+SVG_scale_text(x_coord_label[i-1],y_coord_label[i-
1],substituted_labels[i],colour,fs);
  }else{
     SVG_graph += SVG_ellipseblx(x_coord[i-1],y_coord[i-
1],vertex_radius,vertex_radius,vertex_colours[(numeric_order[i-1]-0)-
1])+SVG_scale_text(x_coord_label[i-1],y_coord_label[i-
1],substituted_labels[i],colour,fs);
  }
     for(j = 1; j <= n; j++){
        if(A[i][j] == 1){
if(ratio_along_line != 0){SVG_graph += SVG_arrow(x_coord[i-1],y_coord[i-
1],x_coord[j-1],y_coord[j-1],ratio_along_line,filled)}else{
SVG_graph += SVG_line(x_coord[i-1],y_coord[i-1],x_coord[j-1],y_coord[j-1])};
     }
  if(A[i][j] == 2){
cxx = (x_coord[i-1]+x_coord[j-1])/2;
cyy = (y_coord[i-1]+y_coord[j-1])/2;
```

```
dx = (x_coord[i-1]-x_coord[j-1]);
dy = (y_coord[i-1]-y_coord[j-1]);
if(dx == 0){theta = 90}else{theta = Math.atan(dy/dx)*180/Math.PI};
rxx = Math.pow((x_coord[i-1]-x_coord[j-1])*(x_coord[i-1]-x_coord[j-1])+(y_coord[i-
1]-y_coord[j-1])*(y_coord[i-1]-y_coord[j-1]),0.5)/2;
ryy = rxx/double_path_skinnyness; // gives a skinny ellipse
SVG_graph += SVG_ellipse_rotate(cxx,cyy,rxx,ryy,theta,double_path_colour);
    }
  }
```

The next portion of the code gives loops on a single vertex when required:

```
      if(A[i][i] != 0){
        SVG_graph += SVG_ellipse(x_coord_loop[i-1],y_coord_loop[i-
1],rrloop,rrloop);}}

return svg_start + SVG_graph + svg_end;}
```

Code 2.7 The code for a graph to be used in a shortest path problem question.

Chapter 3 Content and Design of Graph Theory Problems within Online Learning and Assessment

3.1 Degree

3.1.1 Addition of Weights

Four prepared questions test students' understanding of order. The first question introduces the student to the concept of order by just looking at the weights of the edges within the corresponding digraph. The idea behind the question is to set the concept of in(out)degree in a concrete and easily understood setting.

A trading card game is played amongst 12 players. After one round of play, the game organizer created an network matrix, as shown below, listing the trades.

For example, the highlighted value, $a_{2,9}$ = 3 denotes the number of cards Irene gave to Nicola.

Using the adjacency matrix, determine the number of cards Reginald gave away.

	Ivan	Irene	Darlene	Norman	Rosie	Yasmin	Connie	Reginald	Nicola	Sophie	Enrico	Emily
Ivan	0	1	5	8	7	3	6	8	6	1	3	4
Irene	4	0	1	0	5	5	3	9	3	0	6	5
Darlene	9	1	0	5	10	6	4	3	3	5	4	8
Norman	8	6	6	0	8	6	8	1	4	6	7	6
Rosie	3	1	6	1	0	1	8	8	5	4	0	0
Yasmin	7	1	7	3	6	0	7	6	6	5	3	4
Connie	4	1	2	5	8	5	0	1	7	0	1	6
Reginald	10	4	8	5	0	9	8	0	9	0	9	9
Nicola	2	2	1	2	6	4	1	1	0	5	4	9
Sophie	2	6	9	7	4	4	3	8	9	0	9	1
Enrico	5	3	2	4	2	3	2	8	1	0	0	6
Emily	8	8	6	8	2	4	3	9	5	1	8	0

Figure 3.1 Example of a question on order and network matrices.

The student is asked to determine the number of cards Reginald gave away; however, the coding also allows the possibility for the question to have asked to see how many cards he received. In this question, the correct answer is the sum of the values in the 8th row, i.e. 10 + 4 + 8 + 5 + 0 + 9 + 8 + 0 + 9 + 0 + 9 + 9 = 71.

As a guideline to help students further, there is a note at the beginning of the question that states the meaning of entry, $a_{2,9}$. Also, this is a numeric input (NI) question, which is useful for determining whether or not the student correctly added up the proper values (i.e. the row entries or the column entries for the selected person; zero entries will not affect final outcomes). Doing this helps to enable the student to see the difference between indegree and outdegree.

This question randomly selects one number between eight and fifteen; this number will represent the number of people to be listed in the adjacency matrix. Furthermore, each entry in the adjacency matrix is a random number between zero and ten. However, for determining the values to be used, it is important to use fewer tens simply due to the ease of calculating a sum with more tens included. Therefore, when displaying the adjacency matrix, it is important to edit the normally used formula, Math.round(Math.random()*10), slightly so that fewer tens can appear. Therefore, the formula being used for each entry is given as Math.round(Math.random()*9.7).

3.1.2 Order of an Undirected Graph

The next question involves asking the student for the degree of a particular vertex (in this case, the vertex is Cormac). Note that this is not the same as asking for the sum of entries in a row or column because degree only refers to the number of connections to (or from) a particular vertex. Because students may easily make this error, it is important to highlight this concept and compare it with the previous question on the sum of weights in order to more clearly demonstrate the difference between the two concepts. Also, because this question is asking for the degree of a particular vertex, the adjacency matrix must be symmetrical in every case. However, do note that although the corresponding network matrix does not have to be symmetric, it is made symmetric in this question to avoid confusion for students.

The programming used in this question is the same as the previous question, but instead, there are more zeros being included. In doing this, the likelihood of having similar answers every time is reduced, albeit by a small amount. This question could also easily be made into a **Responsive, Numeric**

Input (RNI) question, allowing for some additional feedback for answering the question by calculating the sum of the weights in the associated row or column. This type of question is most useful to teachers who can then see how often various distracters are used by a cohort of students. However, the feedback states the solution with much detail, including a graphical interpretation of the data, and furthermore, such an incorrect answer would not warrant any partial credit anyway. Therefore, it is more appropriate to use Numerical Input for this problem.

A trading card game is played amongst 13 players. After one round of play, the game organizer created a network matrix, as shown below, listing the trades.

For example, the highlighted value, $a_{3,4} = 3$, denotes the number of cards Claudia gave to Wayne.

Using the adjacency matrix, determine the degree for Cormac.

Please note that this matrix (obviously) corresponds to a weighted, simple graph.

	Paolo	Lenka	Claudia	Wayne	Hannah	Tony	Tia	Rupert	Lee	Joseph	Cormac	Linda	Mundeep
Paolo	0	1	2	1	0	2	3	7	7	3	0	4	0
Lenka	1	0	3	1	0	3	1	4	1	0	6	6	2
Claudia	2	3	0	3	6	3	2	2	1	1	1	8	1
Wayne	1	1	3	0	6	2	0	4	1	1	1	1	0
Hannah	0	0	6	6	0	4	1	2	0	2	0	8	4
Tony	2	3	3	2	4	0	1	8	2	1	1	5	2
Tia	3	1	2	0	1	1	0	9	4	6	5	4	7
Rupert	7	4	2	4	2	8	9	0	3	0	2	0	2
Lee	7	1	1	1	0	2	4	3	0	2	1	2	1
Joseph	3	0	1	1	2	1	6	0	2	0	6	3	3
Cormac	0	6	1	1	0	1	5	2	1	6	0	1	2
Linda	4	6	8	1	8	5	4	0	2	3	1	0	7
Mundeep	0	2	1	0	4	2	7	2	1	3	2	7	0

Figure 3.2 Example of a question about degree in Graph Theory.

SOLUTION

The degree of a vertex is the number of edges connected to the vertex. For directed graphs, there are two types of degrees, namely indegree and outdegree. In this question, the ***vertex*** in question is Cormac. Also note that this is a **symmetric matrix**.

As each trade between players is an edge in a network, you need to count the number of **players** with whom Cormac traded and **NOT** the total number of cards traded.

Therefore, as there are 10 players who traded cards with Cormac, the degree is 10.

Below is the graph that represents the network matrix. Recall that any non-zero entry represents **one** edge in the corresponding adjacency matrix and hence, only represents one edge in the graph itself.

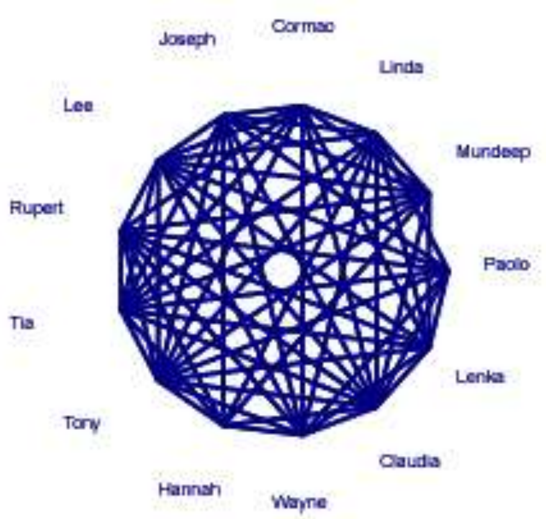


Figure 3.3 Solution to example on degree in Figure 3.2.

3.1.3 Order of a Directed Graph

The third question asks either for the indegree or the outdegree of a particular vertex. In the first example, the question is asking for the indegree for Linda. Similar to the previous question, this question is asking for the number of people that have traded any number of cards (greater than zero) to or from a particular player. However, in this question, students can very easily misinterpret indegree and outdegree or they may even simply not read the question properly and calculate the wrong degree. Therefore, having an RNI question here is a considerable option and hence, has also been created.

A trading card game is played amongst 10 players. After one round of play, the game organizer created an network matrix, as shown below, listing the trades.

For example, the highlighted value, $a_{6,2}$ = 3 denotes the number of cards Hwan-Hee gave to Tania.

Using the adjacency matrix, determine the indegree for Linda.

	Tiffany	Tania	Edward	Bjoern	Rita	Hwan-Hee	Dennis	Mandeep	Daniel	Linda
Tiffany	0	1	3	2	3	2	2	3	3	3
Tania	3	0	4	2	0	3	1	2	3	1
Edward	2	0	0	2	2	2	1	3	2	3
Bjoern	1	0	3	0	1	3	2	3	1	4
Rita	4	1	4	3	0	2	3	2	2	3
Hwan-Hee	1	3	4	3	0	0	3	4	2	3
Dennis	3	3	0	1	3	3	0	1	4	2
Mandeep	1	1	2	3	2	1	2	0	2	1
Daniel	1	1	0	3	3	3	1	1	0	4
Linda	2	0	0	0	3	0	2	2	1	0

Figure 3.4 An example of a question about the indegree of a vertex.

The second example is looking at the outdegree for Hannah.

A trading card game is played amongst 9 players. After one round of play, the game organizer created an network matrix, as shown below, listing the trades.

For example, the highlighted value, $a_{5,6}$ = 2 denotes the number of cards Tony gave to Rebecca.

Using the adjacency matrix, determine the outdegree for Hannah.

	Xavier	Sidney	Cleo	Richard	Tony	Rebecca	Hannah	John	Winifred
Xavier	0	2	2	3	2	2	1	0	0
Sidney	1	0	4	1	3	4	0	3	2
Cleo	3	3	0	0	3	1	2	2	2
Richard	2	0	1	0	2	3	0	1	1
Tony	1	4	3	2	0	2	3	2	1
Rebecca	2	4	1	3	0	0	3	1	3
Hannah	1	1	2	3	1	1	0	0	1
John	3	2	3	3	1	1	0	0	2
Winifred	0	2	2	3	2	1	2	1	0

Figure 3.5 An example of a question about the outdegree of a vertex.

3.1.4 Degree Sequences

Also important when understanding order is understanding how particular sequences of vertex degrees can lead to particular graphs:

Def. 3.1 A **degree sequence** is a monotonic, non-increasing sequence of vertex degrees[46].

This is usually applied to undirected graphs as digraphs can have differing indegrees and outdegrees for the same vertex of a particular graph. For undirected graphs, it is necessary to note that the sum of degrees in the sequence needs to be even as every edge joins two vertices.

The example shown in Figure 3.6 illustrates one particular graph and the formatting of the question. Note the italicised print in the example. Also note the question format is special because it amalgamates two other question types, namely Word Input (WI) and Responsive Numeric Input (RNI), to create a **Responsive Word Input** (RWI) question. The italicised print makes it clear how to input the answer. In this case, all values need to be separated only by a comma and there should be no spaces whatsoever in the response. Additionally, there are two hidden distracters in this question, namely

- List is in non-decreasing, numerical order (e.g. {4,4,6,6,7,8,9})
- List is in alphabetical order of vertices (e.g. {4,7,4,6,9,6,8})

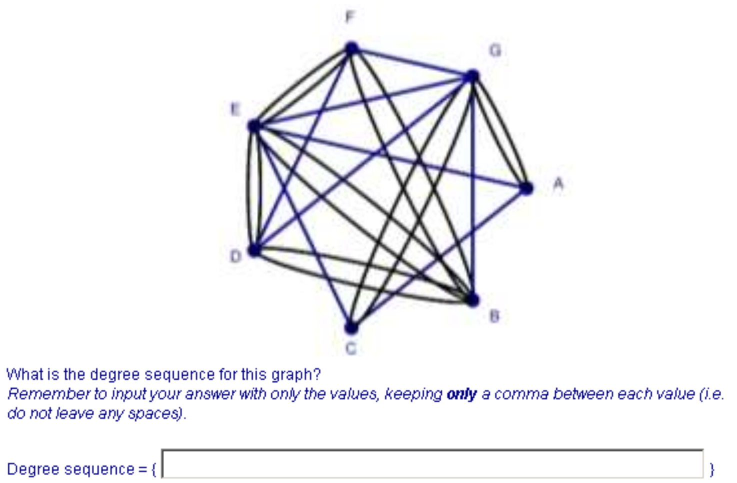


Figure 3.6 An example of a Responsive Word Input (RWI) question about the degree sequence of a graph.

If a student was to implement the first distracter, then (s)he would receive a partial credit of 1 mark out of 2 for having the correct values in a particular, numeric order, although the order was incorrect. However, if the student implements the second distracter, then (s)he would not receive any marks as (s)he did not attempt to arrange the values in any particular, numeric order, but rather typed in the values as they appear alphabetically in the graph.

3.2 Adjacency Matrices

3.2.1 Translating a Graph and an Adjacency Matrix

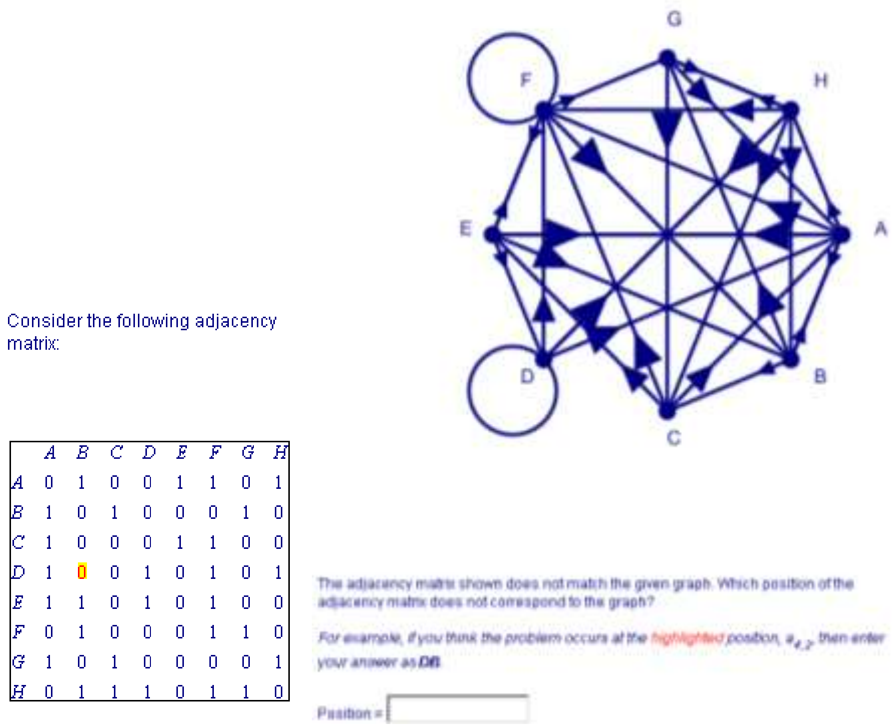


Figure 3.7 An example of a Responsive Word Input (RWI) question, asking to find the error between an adjacency matrix and its corresponding graph.

To use adjacency matrices, it is important to be able to look at a graph and translate the information from it into an adjacency matrix; likewise, it is important to be able to look at an adjacency matrix and be able to draw a graph from it. The question shown in Figure 3.7 allows the student to practice doing this by asking him/her to find an error between an adjacency matrix and its corresponding graph.

This is an RWI question as the correct answer is a string of two letters instead of a typical, numerical response. This question uses digraphs instead of undirected graphs, whose adjacency matrices are symmetric, thus making it easier to detect an error, whereas simple digraphs are not symmetric and so, it is more difficult to find any error in the corresponding adjacency matrix. The highlighted position is also used to help the student understand how to input the answer in the text box.

In this question, the correct answer is $\overrightarrow{FE}$. In the matrix, $\overrightarrow{FE} = 0$, but the edge exists in the graph as there is an arrow going from F to E.

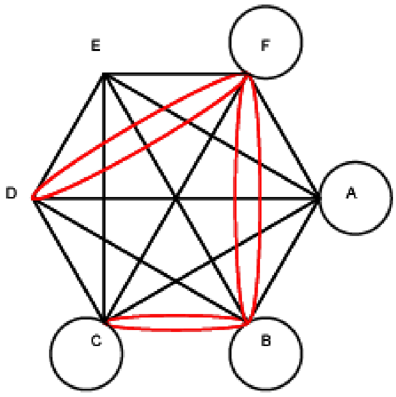
3.2.2 Finding an Appropriate Adjacency Matrix

There are three questions to this section, which look at comparing adjacency matrices to their graphs. The first question looks at a graph and asks students to determine the correct adjacency matrix. An example of this is shown in Figure 3.8. However, before looking at the example, it is important to recall the following definition of a loop[47]:

Def. 3.2 A **loop** is an edge that connects a vertex to itself or a pair of edges that are both connected to the same pair of vertices (also known as **parallel edges**).

Note that the red edges represent loops travelling between two vertices. The correct answer to the problem in Figure 3.8 is “None of these!”.

Dora is given the following graph. What is the corresponding incidence matrix?
Double paths are shown in red for clarity.



○

	A	B	C	D	E	F
A	2	1	1	1	1	1
B	1	2	1	1	1	2
C	1	1	2	1	1	1
D	1	1	1	0	1	2
E	1	1	1	1	0	1
F	1	2	1	2	1	2

○

	A	B	C	D	E	F
A	2	2	2	2	2	2
B	2	2	2	2	2	0
C	2	2	2	2	2	2
D	2	2	2	1	2	0
E	2	2	2	2	1	2
F	2	0	2	0	2	2

○

	A	B	C	D	E	F
A	1	1	0	0	0	0
B	1	1	1	0	1	0
C	1	1	1	1	0	0
D	1	1	1	0	0	0
E	1	1	1	1	0	0
F	1	0	1	0	1	1

○

	A	B	C	D	E	F
A	1	1	1	1	1	1
B	1	1	1	1	1	1
C	1	1	1	1	1	1
D	1	1	1	0	1	1
E	1	1	1	1	0	1
F	1	1	1	1	1	1

○ None of these!
○ I don't know!

Figure 3.8 An example of a question asking to match the graph to one of four adjacency matrices.

3.2.3 Finding an Appropriate Graph

The second question looks at an adjacency matrix and asks the student to determine the correct graph. This question is as simple as the previous question since the creativity of having a student select an incorrect answer only lies with the creation of one matrix and then adding (or subtracting) random edges from it to create the other graphs.

Because this question simply involves matching the adjacency matrix to the correct graph, there are virtually no common errors that can be made, other than a simple error in matching the edges, or lack thereof, to the given adjacency matrix. Therefore, this question is only worth one mark. Also, for the same reason, the feedback is limited to suggest that this problem may be solved using the method of deduction and that they need to carefully eliminate each distracter.

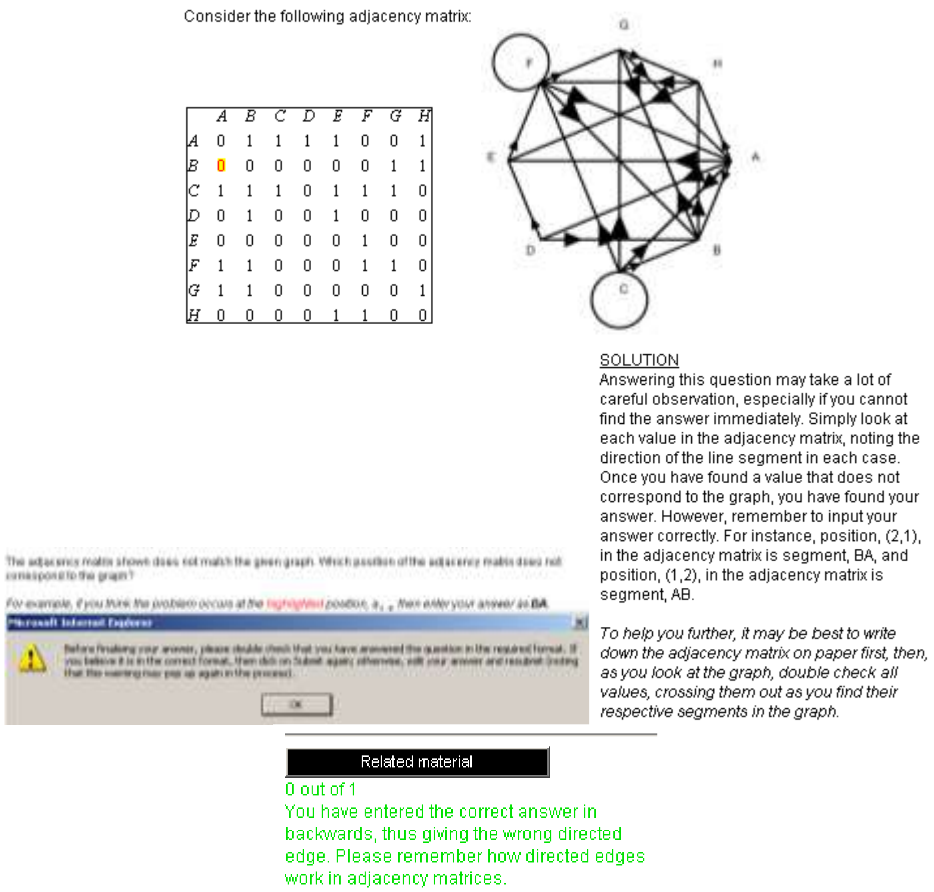


Figure 3.9 An example of a Responsive Word Input (RWI) question, asking to find an error in the corresponding graph for a given adjacency matrix, including a pop-up window that appears after first submitting an answer.

The third question, shown in Figure 3.9, looks at an adjacency matrix and a corresponding digraph for it. In this question, the student is asked to examine the digraph and determine which edge is not in the digraph, but appears in the adjacency matrix. In order for the student to input a feasible correct answer, details about how to input the answer are given immediately above the answer box. However, for added effectiveness, one position in the adjacency matrix is randomly highlighted and a corresponding answer is then given; note that this may actually be the correct answer, but will often not be the case. Also, as this looks at a digraph, it is possible for the answer to not be in alphabetical order (i.e. $\overrightarrow{BA}$ instead of $\overrightarrow{AB}$), but this should be expected of any capable student undergoing such questions in graph theory.

When submitting an answer, a pop-up window appears, asking the student to make sure that the answer submitted is in the proper format and that it is exactly what they wish to submit. The reason for choosing a pop-up window is because it appears instantly and usually with a sound effect included so that students are immediately drawn to it. Also, once a pop-up window appears, students cannot continue answering the question until the pop-up window is closed, so it compels them to pay attention to it. Once the pop-up window is closed, if the student is still satisfied with his/her answer, then (s)he can resubmit the answer; otherwise, (s)he can submit another answer; in doing so, though, the pop-up will reappear, but this is noted in the pop-up window already.

As this question simply involves a quick search for a missing edge, one mark is given for a correct answer. However, if a student were to give the reverse of a correct answer, then no marks are given, but additional feedback is given to suggest what error the student may have made.

3.3 Edge and Vertex Sets

Four questions have been prepared within the Edge and Vertex Sets section of Mathletics. All of the questions in this section use the Word Input (WI) format with a character check to ensure that the length of a student's answer is the same as that of the correct answer; this also prevents students from writing any derogatory remarks or any other unnecessary things in the answer box.

3.3.1 Vertex Sets

The first question in this section asks to determine the vertex set for a randomly chosen graph. This question is relatively simple and the parentheses are already included so that students do not accidentally select a different set of parentheses when attempting the question.

3.3.2 Vertex Sets for Unconnected Graphs

In the question shown in Figure 3.10, some students may fail to recognise that although vertices, C and D, are *disconnected* from the rest of the graph, they are still part of the graph, as shown. Therefore, some students may give the answer, {A,B,E,F,G}, even though the correct answer is {A,B,C,D,E,F,G}.

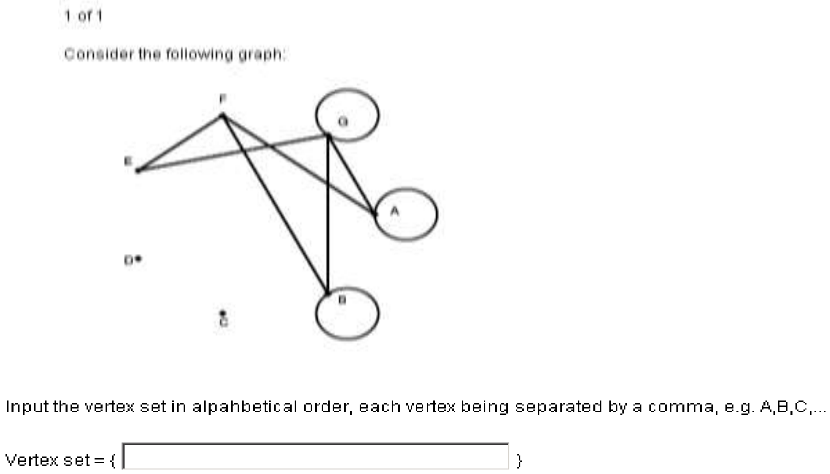


Figure 3.10 An example of a question asking to determine the vertex set of a graph with two vertices, namely C and D, having no edges connected to them.

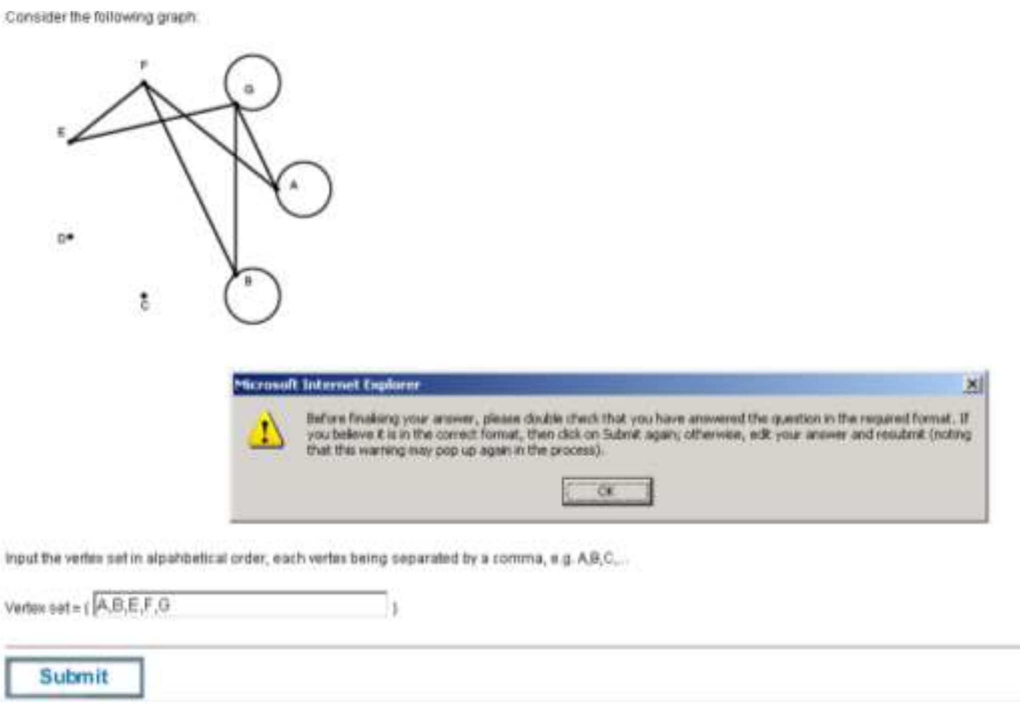


Figure 3.11 An alert box appears, asking the student to verify the answer entered before clicking on Submit a second time.

One issue when dealing with this question is the generic format of the Word Input questions. If a student answered A,B,E,F,G, then (s)he would be given an alert, noting that the answer is invalid because of the number of characters in the answer. Because of this, students may ponder about the reasoning for this. One option, which has been implemented in this question and shown in Figure 3.11, is to change the alert so that it does not give away the answer so easily. For both of these questions, the alert will appear automatically, reminding students to verify their answers. If a student decides to change his/her answer, then the alert will appear again. However, if a student decides to keep his/her answer, then, upon clicking the Submit button again, the alert will not appear and the answer will be recorded and evaluated.

3.3.3 Edge Sets for Undirected Graphs

This question does not involve digraphs, so students just need to enter each edge separately and in alphabetical order. There are no specified common errors that could be made here, other than a simple error in listing all of the edges. The only problematic situation is the inclusion of loops in a graph. Using the format provided in the question, students should notice that the loop for vertex, A, should be labelled as AA because the labels show the beginning and end vertices for each edge; similarly, the loop for vertex, D, should be labelled as DD. However, this would be an error in the understanding of the question, not a perceived common error. In this particular question, the correct answer would be {AA,AB,AD,BC,BD,BE,CE,DD,DE}.

3.3.4 Edge Sets for Directed Graphs

The final question, as shown in Figure 3.12, in this set is similar to the previous question, but it makes use of digraphs instead. Note that the question type implemented is Responsive Word Input (RWI).

The important thing to note in this question is the direction of the edges. It is possible for a digraph to have all edges listed so that each pairing is in alphabetical order, such as $\overrightarrow{AB}$, $\overrightarrow{CD}$, and $\overrightarrow{EF}$. Therefore, one edge, at least, is deliberately set so that it will go in anti-alphabetical order, such as $\overrightarrow{EA}$; this is done by randomly selecting two unequal values from 1 to *n*, where *n* is the number of vertices in the question, and allowing $a_{j,i} = 1$, where $j > i$, $a_{i,j} = 0$.

In this question, the ordering of the letters for each edge is important because, for instance, $\overrightarrow{CF}$ does not appear in the graph because there are no edges going in that direction, but $\overrightarrow{FC}$ does appear because there is an edge that shows an arrow pointing from F to C. For this given question, the correct answer would be

$$\{\overrightarrow{AD}, \overrightarrow{AE}, \overrightarrow{BC}, \overrightarrow{BD}, \overrightarrow{BF}, \overrightarrow{CA}, \overrightarrow{CE}, \overrightarrow{CF}, \overrightarrow{DC}, \overrightarrow{ED}, \overrightarrow{EF}, \overrightarrow{EG}, \overrightarrow{FG}, \overrightarrow{GA}, \overrightarrow{GB}, \overrightarrow{GD}\}$$

Of course, when students answer this question, they will not have use of the vector arrows, so they will just input the letter pairs instead.

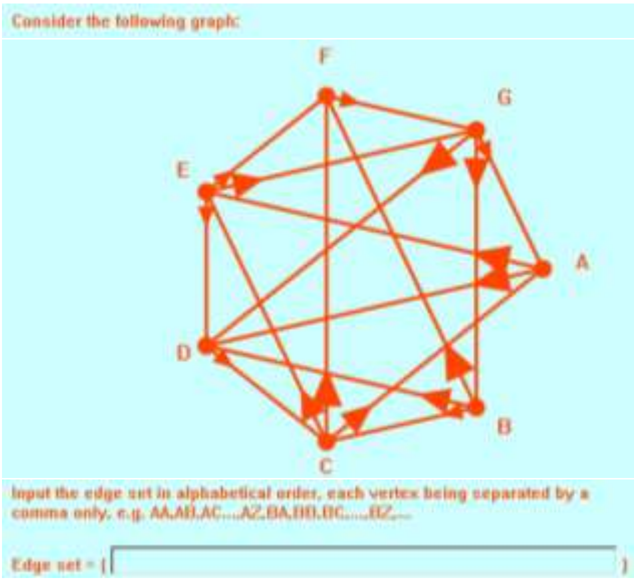


Figure 3.12 An example of a Responsive Word Input (RWI) question asking to determine the edge set of a digraph.

3.4 Simple and Connected Graphs

Due to the simpler nature of simple and connected graphs, both graph types are combined together in this section when writing questions on these topics. Also, because of the simple nature of the graphs, there are only three questions. The first question looks at finding a simple and connected graph among a list of four candidates, the second question looks at finding a simple and connected graph among a list of four adjacency matrices, and the third question combines graphs and adjacency matrices. However, in order to make the question slightly more difficult, a scenario is given in such a way that a student reading the question must determine what has to be found.

Distracters are easily determined using the definitions of simple and connected graphs: One distracter has a loop around one vertex and another distracter has a loop around a pair of vertices. A third distracter is not connected. A fourth distracter has loops everywhere.

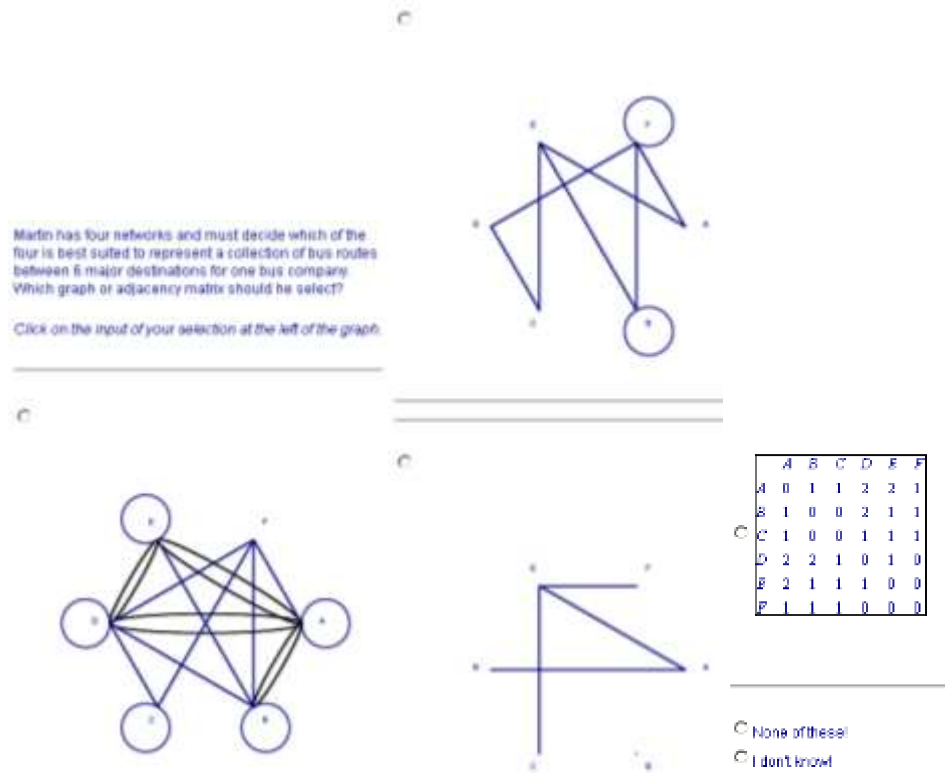


Figure 3.13 A Multiple-choice (MC) question asking to find the simple, connected graph that could be formed among a list of candidate adjacency matrices.

3.5 Hamiltonian and Eulerian Cycles

There are six questions that deal with these two special cycles, all of which are Multiple-choice (MC) questions. For each cycle, there is one question including the graphs themselves, one question including the adjacency matrices to the graphs, and one example including a random mixture of graphs and adjacency matrices. Even though these are all MC questions, the skills and understanding needed to answer these questions will become very challenging, in a positive way, for most students.

3.5.1 Finding a Graph or Adjacency Matrix that is Non-Hamiltonian

3.5.1.1 Finding a Graph that is Non-Hamiltonian

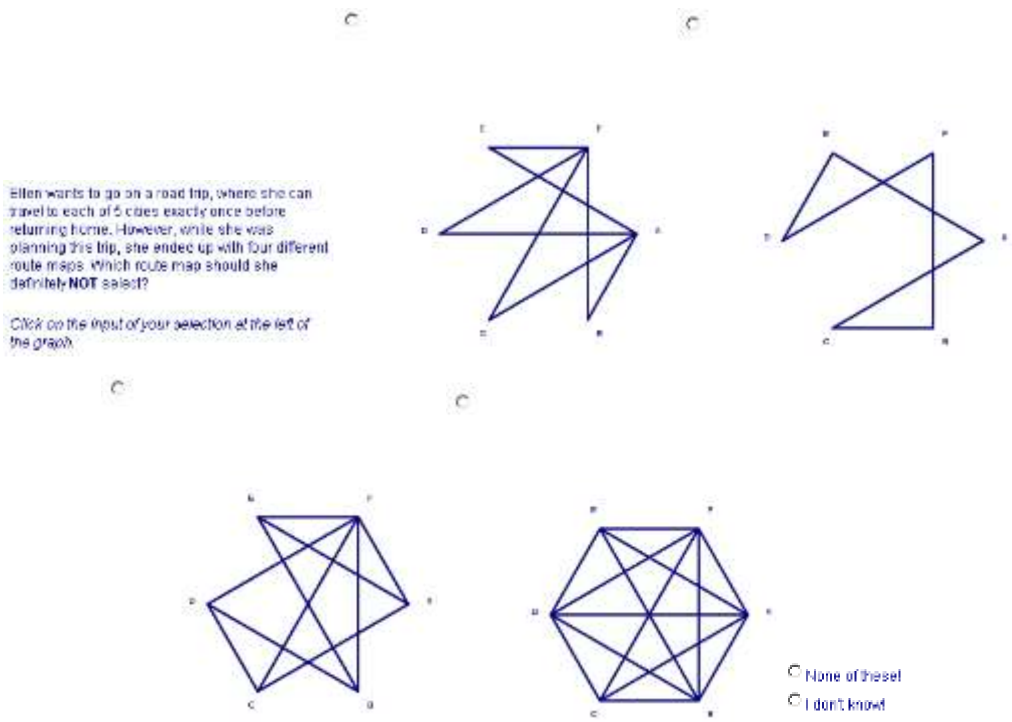


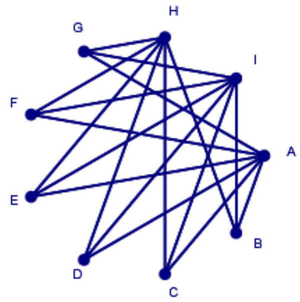
Figure 3.14 An example of a Multiple-choice (MC) question asking to find the graph that does not fit the properties of a Hamiltonian graph.

The first question gives a scenario of a person travelling to a number of cities exactly once before returning home. However, in the problem, the person involved has a set of different route maps (for different areas obviously). The problem at hand, though, is that one of the maps may not be sufficient for doing such a trip. Therefore, the student is asked to determine which of the maps the traveller should definitely not select.

For this question to work, a lot of mathematical theory is required. In order to obtain the correct answer (bottom left), the following theorem is needed:

Theorem 3.1 Let G be a connected, bipartite, undirected graph with the vertices, V, partitioned as $V = V_1 \cup V_2$. If $|V_1| \neq |V_2|$, then G cannot have a Hamiltonian cycle.[48]

Example 3.1 For the bipartite graph shown in the figure below, show why there cannot be a Hamiltonian cycle.



Note that the two partitions in this bipartite graph are $V_1 = \{A, H, I\}$ and $V_2 = \{B, C, D, E, F, G\}$, $|V_1| = 3$, $|V_2| = 6$, and that $|V_1| < |V_2|$. Suppose we start with vertex, A, in V_1. From vertex, A, we could travel to any vertex in V_2, so choose vertex, B. From vertex, B, we need to return to V_1, so choose vertex, H. From vertex, H, we need to return to V_2, so choose vertex, C. From vertex, C, we need to return to V_1, so choose vertex, I. From vertex, I, we need to return to V_2, so choose vertex, D. From vertex, D, we need to return to V_1, but we cannot do so as we have already visited every vertex in V_1 and as we have not yet visited vertices, E, F, or G, we cannot possibly obtain a Hamiltonian cycle starting with a vertex in V_1. A similar pattern would occur if we began with a vertex in V_2. □

In this question presented in Figure 3.14, the traveller, Ellen, wants to visit a total of eight cities on her road trip. She has four maps for four different areas. In two of the graphs, namely the second and fourth graphs, there is clear evidence of Hamiltonian cycles being present in each as the second graph (top centre) is only one Hamiltonian cycle (thus also making it an Eulerian cycle) and as the fourth graph (right side) is nearly complete, i.e. almost all vertices are connected by all other vertices. The third graph (bottom centre) can also be shown to have the Hamiltonian cycle, $A \to C \to D \to F \to B \to E \to A$. All that now has to be decided is whether or not the first graph has a Hamiltonian cycle. From looking at this graph, it is obvious to see that it is a **bipartite graph** of partitions of sizes, two and four. As these two partitions are not equivalent, it is not possible to form a Hamiltonian cycle with this graph[48].

With this question, it would not be sufficient to leave this as the only possible solution as students may discover the pattern through repeated attempts. Therefore, alternate solutions can be randomly generated, such as the graph shown below:

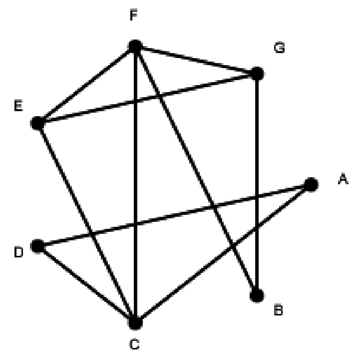


Figure 3.15 An example of an alternate solution to the problem involving Hamiltonian cycles with graphs given, using two separate cycles connected by one vertex.

This graph is formed by creating a Hamiltonian path from a starting vertex to an end vertex. From there, a set of edges is added, starting with the end vertex of the Hamiltonian path, where each vertex is connected to the vertex that is of length, two, away from it in the path. However, the second last edge in the Hamiltonian path is exempt from this, thus causing a potential Hamiltonian cycle to disappear. This causes two distinct Hamiltonian cycles to form, one of which being a triangle, connected by a joint vertex. However, together as one graph, no Hamiltonian cycles can be formed as any cycle would have to reach the joint vertex at least twice before it returns to the starting vertex (three times if the joint vertex is the starting vertex). In this example, a Hamiltonian path is given as $A \to D \to C \to E \to F \to G \to B$.

For this problem, a third alternate solution is provided so that students may not quickly recognize any patterns in the correct solutions.

With this graph, a Hamiltonian path is created by randomly arranging the order of the vertices, then creating a connecting path between adjacent vertices in the arrangement. In this example, a Hamiltonian path is

$$A \to E \to F \to \mathbf{G} \to \mathbf{B} \to C \to D \to H .$$

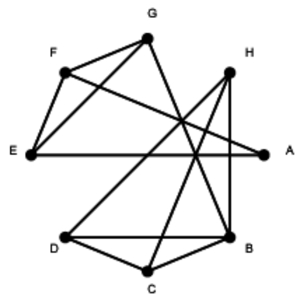


Figure 3.16 An example of an alternate solution to the problem involving Hamiltonian cycles with graphs given, using two separate cycles connected by one edge.

Two cycles are created by splitting the *n* vertices into two *equal* partitions (i.e. equal partitions when *n* is even and having one partition with an extra element when *n* is odd); in this example, the partitions are {A,E,F,G} and {B,C,D,H}. Each partition then has a Hamiltonian cycle added to it by creating a simple polygon from the first vertex in the partition to the last vertex. Extra edges are randomly added between vertices in a partition so that each partition of the graph (or **subgraph**) may be more unique. However, even though each partition has a Hamiltonian cycle, the graph itself does not as the two partitions are connected by one edge (from the Hamiltonian path) created between the last vertex in the first partition and the first vertex in the second partition; in this case, the edge is $\mathbf{G} \rightarrow \mathbf{B}$. Similar to the previous solution, these specific vertices will be used at least twice (or three times if one of them is a starting vertex) in order to make a cycle. Therefore, no Hamiltonian graphs can be formed.

This question does require some skill in finding the Hamiltonian cycles within the other graphs in order to eliminate them as candidate solutions. However, towards the end, it becomes progressively more difficult to find Hamiltonian cycles, especially if the above examples of two disjoint cycles were to appear. Also recall that although this is a MC question, the option, *None of these!*, is always available and is sometimes the correct solution, thus making this question all the more challenging.

3.5.1.2 Finding an Adjacency Matrix that is Non-Hamiltonian

The second question looks at the adjacency matrices rather than the graphs, but is otherwise identical to the previous question, where a student has to find the adjacency matrix that does not generate a Hamiltonian cycle. With this question, it is more difficult to find Hamiltonian graphs as the graphs are simply not given in the question.

3.5.1.3 Finding Either a Graph or an Adjacency Matrix that is Non-Hamiltonian

This question combines graphs and adjacency matrices within multiple-choice answers to create a more challenging question. This question forces students to use both their visual and logical intelligences in order to solve the problem correctly. In this question, the third option has been randomly chosen to be shown as a graph, whereas the other three options have been randomly chosen to be shown as adjacency matrices. With each option having its own "switch", it is possible, with a probability of $\frac{1}{8}$, for all four options to appear as graphs or for all of them to appear as adjacency matrices.

3.5.2 Finding a Graph or Adjacency Matrix that is Eulerian

The next three questions look at Eulerian cycles. In contrast to the Hamiltonian graphs questions, these questions ask to find a graph or adjacency matrix that is indeed Eulerian.

3.5.2.1 Finding a Graph which is Eulerian

The third question asks to find an Eulerian cycle from a list of graphs. Similar to the questions on Hamiltonian cycles, this question gives a road trip scenario, but instead of visiting several towns or cities, the traveller simply wants to go on a journey, possibly just to explore the sights. In doing so, the traveller does not want to travel the same road twice, so (s)he maps out his/her journey by making a note of every road travelled before going home along another route to maximize the amount of scenery witnessed.

An Eulerian graph appears if and only if there is, at most, one nontrivial component and (more importantly) all of the vertices have even degree[49]. Therefore, for some students, answering this question will be simple as they can easily eliminate three candidate solutions, at least, by just looking at the degrees of the vertices in each graph.

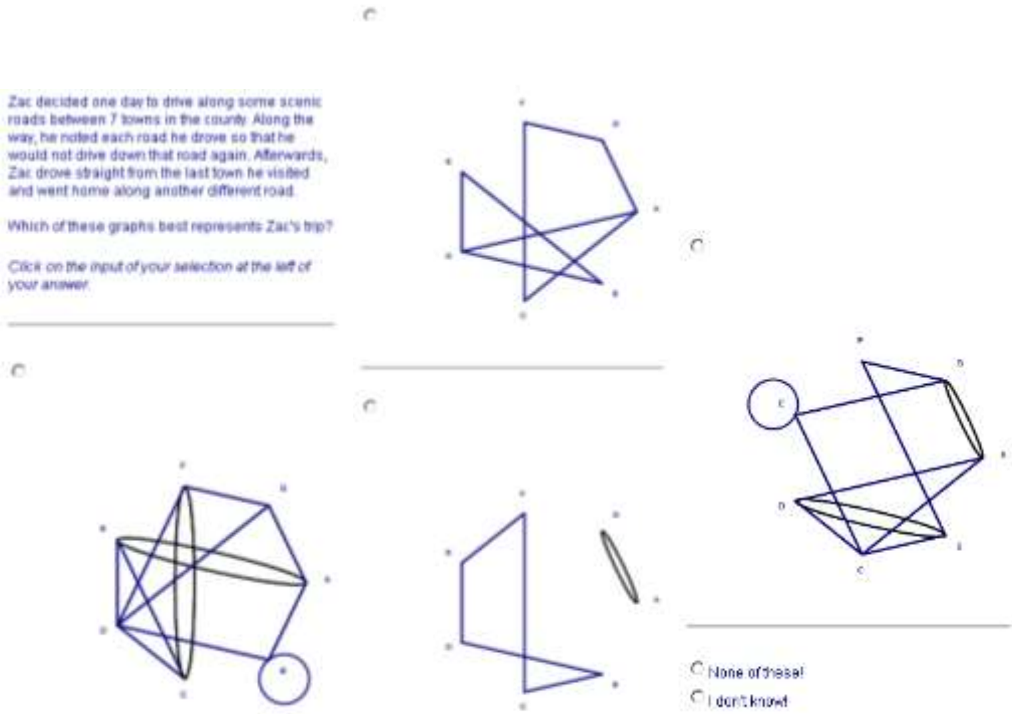


Figure 3.17 An example of a Multiple-choice (MC) question, asking to find the graph that best fits the properties of an Eulerian graph.

Technically speaking, Eulerian graphs are easier to study mathematically than Hamiltonian graphs[53]. This can be explained using the characterization of Eulerian graphs[50]:

The following statements are equivalent for a connected graph, G:

1. *G is Eulerian.*
2. *The degree of every vertex is even.*
3. *G is the union of edge-disjoint cycles*.

From this, it is clear that students just need to know that the degree of every vertex is even and that the graph is *not* one with no cycles (i.e. a **tree**) in order to see that the graph is Eulerian. As such, the marking scheme for the questions generated is reduced somewhat.

In the example shown in Figure 3.17, it can be easily shown that the fourth graph (right side) has an Eulerian cycle with the path,

$$E \rightarrow E \rightarrow G \rightarrow A \rightarrow G \rightarrow F \rightarrow B \rightarrow D \rightarrow B \rightarrow\rightarrow D \rightarrow A \rightarrow C \rightarrow E .$$

This question is very simple to answer, especially when the candidate solutions can be eliminated so easily. However, in order to eliminate each candidate solution, each graph must be analysed in order to find such imperfections.

3.5.2.2 Finding an Adjacency Matrix That is Eulerian

The next question is identical to the previous question, but again, involves adjacency matrices rather than graphs. However, finding an Eulerian cycle is more complex in this question as a student answering this question needs to realize that the graph must be connected, as well.

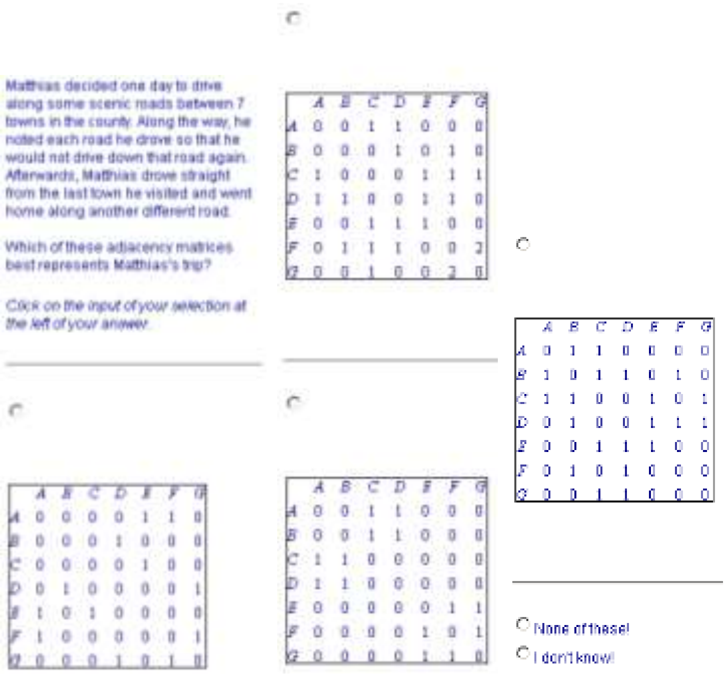


Figure 3.18 A Multiple-choice (MC) question asking to find the adjacency matrix that best fits the properties of an Eulerian graph.

This question is easier for students to answer if they can visualise the graphs by drawing them manually. However, simply counting the degrees of the vertices will not be enough as two disjoint subgraphs would suffice this property, but not be Eulerian. Therefore, students need to take more time to consider whether any adjacency matrix with all vertices of even degree is indeed Eulerian.

This question is worth 4 marks, but the other answers are worth either 2 marks, 0 marks, or -2 marks, depending on the adjacency matrices that appear in the questions.

Interestingly enough, the answer to the problem given in Figure 3.18 is:

	A	B	C	D	E	F	G
A	0	1	1	0	0	0	0
B	1	0	1	1	0	1	0
C	1	1	0	0	1	0	1
D	0	1	0	0	1	1	1
E	0	0	1	1	1	0	0
F	0	1	0	1	0	0	0
G	0	0	1	1	0	0	0

Figure 3.19 Correct answer to problem given in **Figure 3.18**.

What makes this answer interesting is that, at first glance, it does not appear to follow the second item that characterises Eulerian graphs (namely that t*he degree of every vertex is even*[50]) as the sum of the entries in row (or column) E is odd. However, looking more carefully, it can be seen that there is a loop around vertex E. Therefore, this can be ignored and each row sum is now even. Furthermore, the graph is connected and has an Eulerian cycle from

$$E \rightarrow E \rightarrow D \rightarrow F \rightarrow B \rightarrow A \rightarrow C \rightarrow G \rightarrow D \rightarrow B \rightarrow C \rightarrow E.$$

As such, the distracter is not a separate answer, but rather part of the correct answer itself. Of course, this does not always happen, but it is very likely nonetheless. Also, this distracter can only appear within adjacency matrices as the graphical equivalent would immediately give away the loop being present.

3.5.2.3 Finding a Graph or Adjacency Matrix That is Eulerian

The last question, given in Figure 3.20, looks at a combined problem, which views graphs and adjacency matrices together in the same question regarding Eulerian graphs. As noted earlier, a “switch” applied to each option decides whether or not they individually appear as graphs or adjacency matrices.

In the example of Figure 3.20, two graphs have been randomly created; again, note that it is possible for more or fewer graphs to appear in each question. The first adjacency matrix (left side) has odd degree at vertex G. The first graph (top of figure) has the bridge, $\overline{AH}$, which causes the vertices, A and H, to each have odd degree. The second graph (bottom centre) has even degree throughout, but contains two disjoint subgraphs. The second adjacency matrix (right side) has multiple loops, but once ignored, has even degree throughout and is connected. Therefore, the second adjacency matrix is the correct answer. However, notice that vertex C is disjoint from the remaining vertices; this is fine, though, since Eulerian graphs are defined by their edges rather than their vertices. Similar to Figure 3.19, it is possible in graph theory to “distract” students within the correct answer itself by adding a disjoint vertex.

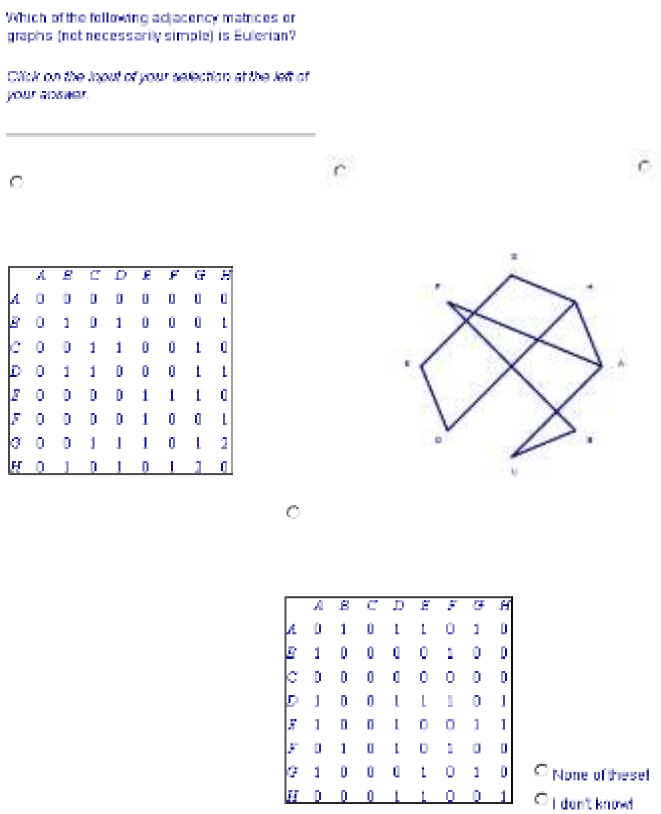


Figure 3.20 A Multiple-choice (MC) question asking to find the graph or adjacency matrix that best fits the properties of an Eulerian graph.

There are random distractions within the correct answer, as well as the distracters themselves. Furthermore, this question tests the visual and logical intelligences within a student, thus requiring them to carefully inspect each solution in a different way before deciding whether or not it is a candidate solution.

However, this is not where the pedagogical side of this question ends. Look again at the questions given in Figure 3.18 and Figure 3.20 and note that these two questions are completely different. The question in Figure 3.18 gives a story of a randomly selected person (in this case, it was Matthias), who goes for a scenic drive one day. Students are not directly told that his drive resembles that of an Eulerian cycle, but they are expected to figure this out on their own in order to answer the question. In comparison, the question in Figure 3.20 directly asks the question, “Which of the following is an Eulerian graph?”. This distinction of indirect and direct questioning is very important pedagogically and two definitions have therefore been made specifically for pedagogical use:

Def. 3.3 A **directed question** is a question that specifies what is required of the student, enabling him/her to immediately work towards answering the question.

Def. 3.4 An **indirected question** is a question that does not specify what is required of the student in order for him/her to answer the question.

The example of Figure 3.18 does not tell the student that Eulerian cycles are being tested in the question, even though (s)he is expected to determine this on his/her own in order to solve the problem. In order to generate such a question, it is most likely that some real-world scenario is given. Therefore, under this assumption, it is expected of the student to be able to translate the problem into a mathematical problem and hence, it is also necessary for the student to apply everyday mathematics into his/her thinking for solving the problem. Furthermore, although the contextualised area of the mathematical problem-solving process is being used, it is not necessary for interpreting the mathematical solution back into this area as such a question may still only require the student to find the initial, mathematical results.

This concept of directed and indirected questions for pedagogical use has been implemented into many graph theory topics, including simple and connected graphs and Hamiltonian cycles, which have already been discussed, and also in other topics, which are discussed later in this chapter.

3.6 Isomorphisms

For all of these questions, any one of the *five* answers programmed into these questions can be chosen as the correct answer (Note that this is obviously also implemented within the question as a randomised function and in the feedback with five different answers.). Also, the other four answers are randomly selected so that, after multiple attempts at practicing these questions, no obvious pattern emerges to the student. As such, this question type is labelled as **Random Selection Multiple-choice (RSMC)**.

3.6.1 Finding an Isomorphic Graph

In these questions, information can be given about the different graph types. This information is very helpful in trying to determine which of the graphs would represent the correct answer. However, if students were expected to trial these questions, then, during an examination, a harder question can be used in replacement, which does not include the additional information; this would force students to recall what they read and learned about the different graph types from trialling the easier question and would also (likely) cause a decrease in marks for students who did not follow the teacher’s instructions.

For all of these questions, each graph has its own unique feedback. In the feedback, students are given easier methods for determining which graph is which type. For instance, in the case of antiprisms, one of the main components is that the degree of each vertex is four. However, this cannot be seen with any of the other graphs for this question, so, although it appears students can now answer the next questions easily, this helps them to learn about the patterns they need to observe when looking for isomorphisms.

3.6.2 Finding an Isomorphic Adjacency Matrix

These questions use adjacency matrices instead of graphs. After numerous attempts, students can use the information acquired from looking at the degrees of each of the vertices in order to determine the graph type for each adjacency matrix. The selection of the matrices is randomised throughout so that students cannot determine any patterns of selection when they should be trying to solve the problems instead.

Each of the questions shown is worth two marks, but a mark of zero is given instead if incorrect. Note that there is no possibility of giving a negative score for these questions as it is very difficult to determine which graph will appear at any time and as it is not as necessary with such a low score for answering correctly.

3.6.3 Finding an Isomorphic Graph or Adjacency Matrix

Naba is a university student at University of Alberta. Recently, she has been learning the following:

n-prisms (n-prism) are graphs with 2n vertices and 3n edges that create two similar shapes of n vertices joined by the corresponding (matching) vertices between them.

n-antiprisms (n-antiprism) are graphs with 2n vertices and 4n edges that create two similar shapes of n vertices, where each vertex of the bigger shape is connected to two adjacent vertices in the smaller shape. For instance, if there are two squares, say ABCD and EFGH, with EFGH being smaller and contained within ABCD, and suppose vertex E is adjacent to vertices, A and B, from the larger square. Then two edges would form, namely AE and BE. This would apply for all vertices in the smaller square.

Wheel graphs (W_n), like that shown below, are graphs where n vertices are joined to its neighbouring vertices to form a cycle and one additional vertex is placed in the center. This vertex is connected to all of the n vertices, thus creating the *spokes* of a wheel.

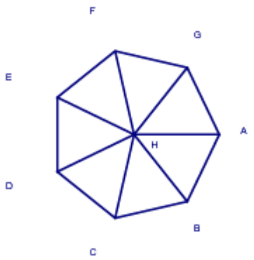
Complete bipartite graphs ($K_{x,y}$) are graphs with n vertices and composed in two sets of x and y vertices. Each vertex in set x connects with all vertices in set y and vice versa. However, there are no edges connecting vertices in one set to other vertices in the same set.

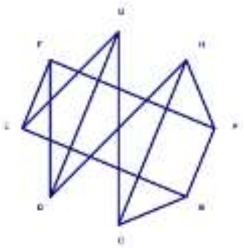
Ladder graphs (L_n) are graphs with 2n vertices and composed in two sets of n vertices each. Vertices in each set are connected only to one adjacent vertex within the same set. However, each vertex in one set is connected to a corresponding vertex in the other set, thus creating a rectangular structure composed of squares, almost like using matchsticks to draw squares connecting to each other.

Naba has been asked to find a graph that is isomorphic to W_7. However, she has found four potential candidates and is unsure which one to select.

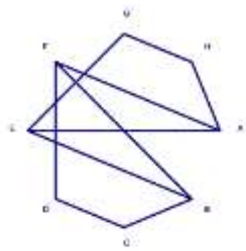
Which of these adjacency matrices or graphs should Naba select?

Click on the input of your selection at the left of the graph or adjacency matrix. You may also choose either None of These or I Don't Know, if you prefer.





	A	B	C	D	E	F	G	H
A	0	0	0	0	1	1	0	1
B	0	0	1	0	1	1	0	0
C	0	1	0	1	0	0	0	0
D	0	0	1	0	0	1	0	0
E	1	1	0	0	0	0	1	0
F	1	1	0	1	0	0	0	0
G	0	0	0	0	1	0	0	1
H	1	0	0	0	0	0	1	0



	A	B	C	D	E	F	G	H
A	0	1	1	1	0	1	1	0
B	1	0	0	0	1	0	0	1
C	1	0	0	0	1	0	0	1
D	1	0	0	0	1	0	0	1
E	0	1	1	1	0	1	1	0
F	1	0	0	0	1	0	0	1
G	1	0	0	0	1	0	0	1
H	0	1	1	1	0	1	1	0

None of these!

I don't know!

Figure 3.21 Example of a Random Selection Multiple-choice (RSCM) question, asking to find the adjacency matrix or graph that is isomorphic to the wheel graph, W_7, as shown in the question.

The last two questions both involve combining graphs and adjacency matrices within the same question. For some realisations, this will surely prove to be a challenge for the students. However, for these examples, as is the case for all questions in this topic, there is an additional feature in each case.

Figure 3.21 has included a visual image of a typical wheel graph to help the student to more easily find the correct solution; in Figure 3.22, the added image is a typical ladder graph. Since neither of these two graphs adheres to the conventions of the circular positioning of vertices, two new functions had to be created. The wheel graph takes form from using the circular positioning of the vertices, but by also including one more vertex in the centre, which joins all of the other vertices. The ladder graph, however, does not consider circular positioning whatsoever and so, another strategy had to be implemented in order to create this, involving a re-positioning of vertices.

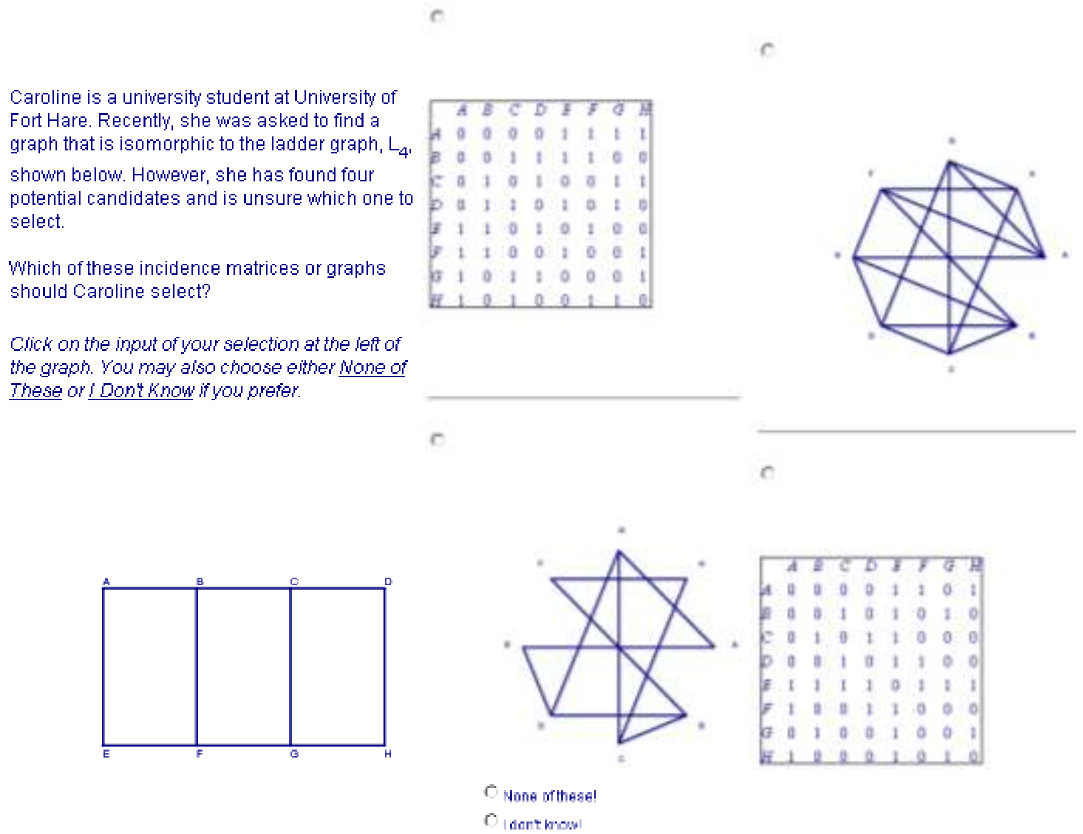


Figure 3.22 Example of a Random Selection Multiple-choice (RSMC) question, asking to find the adjacency matrix or graph that is isomorphic to the ladder graph, L_4, as shown in the question.

3.6.4 Functionality of Random Selection, Multiple-choice Questions

This question type is unique in that all distracters embedded in the code can appear as the correct solutions. For this to occur, five distracters are created and then one of them is randomly selected to be the correct solution to the generated problem. However, as there are five possible answers, the necessary code for generating the problem and the solution all need to be more flexible.

When generating the questions, simple keywords may be implemented by means of separate functions, thus reducing the number of characters needed in generating the full code. The feedback needs to specify the characteristics of each graph, though so that the students reading it may be able to understand how to watch out for these special graphs in future attempts. Therefore, there is no alternative but to create separate feedback for each of the five special graphs.

Finally, there is the issue of additional feedback that could show students what graphs they selected if answers are incorrect. Unfortunately, because the answer is randomly selected, as is the ordering of the other candidate solutions, it is not possible to create any such feedback to warn students of their errors. Nonetheless, students are able to learn about each graph type and their characteristics through multiple attempts and through the provided feedback.

3.7 Bipartite Graphs

In this section, there are four questions to consider. Two of the questions are virtual copies of each other as MC questions, one using graphs and the other using adjacency matrices. However, the third question is a simple, numerical input (NI) question that will test to see if students are paying close attention to the detail of the partitions and how they function in creating bipartite graphs.

3.7.1 Finding a Bipartite Graph

The first question is a MC question, asking to find a complete, bipartite graph with two partitions of unequal amounts. The objective of this question is to find a complete, bipartite graph such that the two vertex partitions are unequal.

3.7.2 Finding a Bipartite Adjacency Matrix

The second question is the copy of this question, using adjacency matrices instead of graphs. In both of these questions, recognising patterns in data is the essential element for solving the problems. For graphs, visualisation makes it far simpler to detect which of the graphs is bipartite.

For the adjacency matrix, the simplest way to detect whether a graph is bipartite is to find matrices with rows and columns of just zeros (or just ones) that match each other and to *Analyse* them first.

3.7.3 Finding a Bipartite Graph or Adjacency Matrix

The third question combines graphs and adjacency matrices randomly to test students using their visual and logical intelligences.

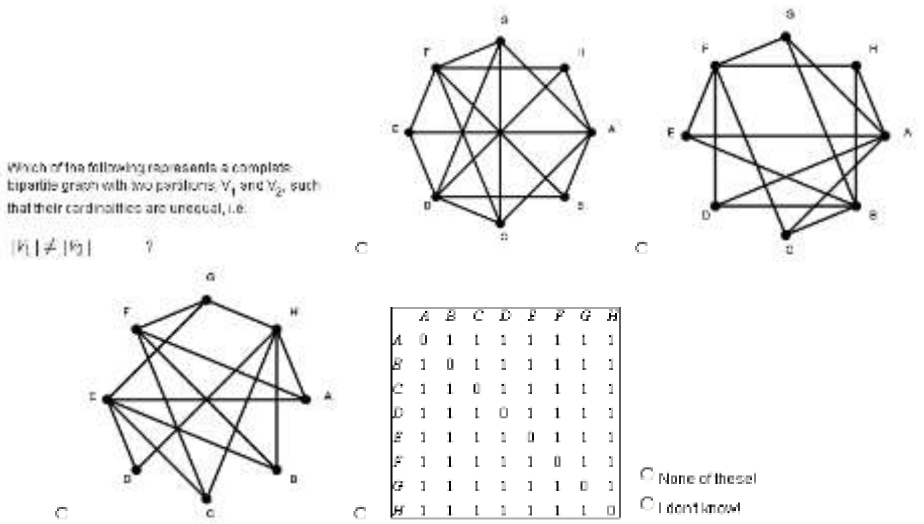


Figure 3.23 An example of a numeric input (NI) question asking to find the number of vertices in the larger partition of a bipartite graph.

These questions are worth 3 marks each, mainly because the questions ask for such specific graphs and because some distracters were initially correct, but then edits were made, causing them to appear *almost correct*; as such, some of the distracters will award partial credit, but none of the answers will give a negative score.

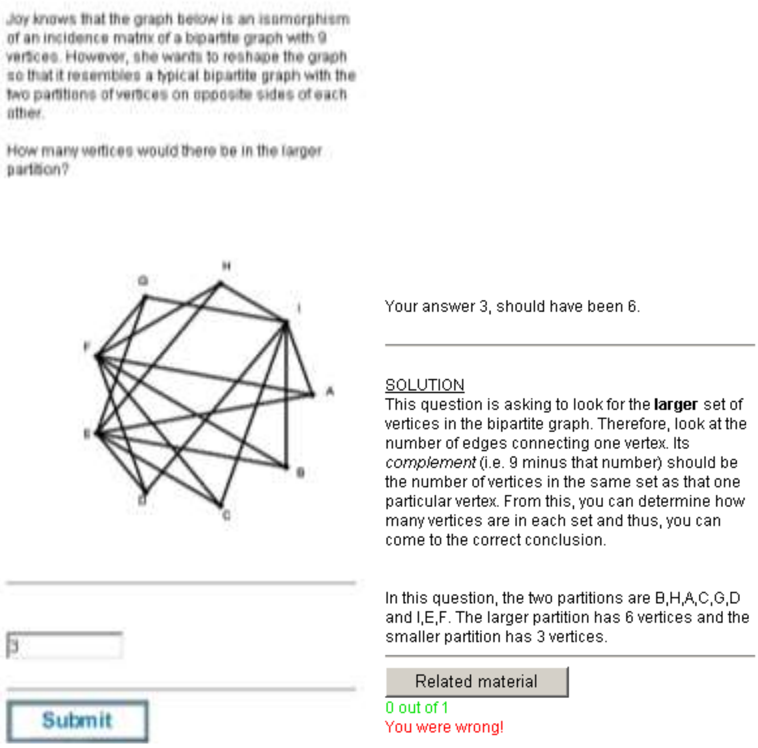


Figure 3.24 An example of a numeric input (NI) question asking to find the number of vertices in the larger partition of a bipartite graph.

3.7.4 Finding the Number of Vertices in a Partition of a Bipartite Graph

The last question in this section looks at the partitions to see how many vertices are in each partition. This is done to ensure students are noticing the graphs, how they function, and how they are connected. To keep with this theme, the bipartite graphs are set up so that they have unequal partitions. Furthermore, the question could ask the student to determine the number of vertices in either the *smaller* partition or the *larger* partition.

In Figure 3.24, the question asks to find the number of vertices in the larger partition of the graph. However, in Figure 3.25, the question asks to find the number of vertices in the smaller partition. In either case, the bipartite graph is created by randomly separating the set of vertices, *V*, into two components, each of different lengths. From this, any vertex in one set is automatically connected to all vertices in the other set; similarly, any vertex in one set is automatically disconnected to all other vertices in the same set.

Notice that this is not a RNI question, but rather just a NI question. As this question should be simple to answer, there are no partial marks to award and thus, there is no great need to use the RNI question type here; instead, the feedback highlights the key word, either *smaller* or *larger*, in bold print to remind the student what (s)he was supposed to solve. As either keyword can appear in this question, it allows for more randomness to appear as the total number of possible realisations doubles to conform to both keywords. This question, which should be easy to solve, is worth 1 mark.

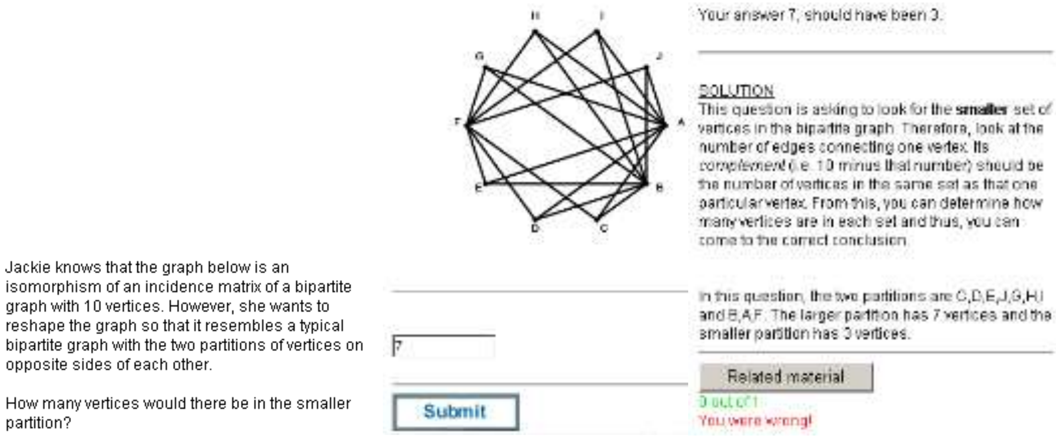


Figure 3.25 An example of a numeric input (NI) question asking to determine the number of vertices in the smaller partition of a bipartite graph.

3.8 Planar Graphs

There are four multiple-choice questions for this section; one uses graphs, one uses adjacency matrices, and one uses both graphs and adjacency matrices. All of these first three questions are similar, but, as will be shown, two of these questions are far more difficult because of such changes. The fourth question uses pertinent information about them in order for students to draw conclusions and make an appropriate decision.

3.8.1 Use of Quantitative Information to Prove Planarity

The first question, which does not use graphs, looks at the students' awareness of the following theorem[51]:

Theorem 3.2 Let $G = (V, E)$ be a planar graph that is simple and connected and with $|V| \geq 3$. Then $|E| \leq 3|V| - 6$.

To answer this question, students need to look at the numbers of edges and vertices present and, using this, determine for which of the possibilities, if any, does the corresponding graph become non-planar. An example of this is shown in Figure 3.26.

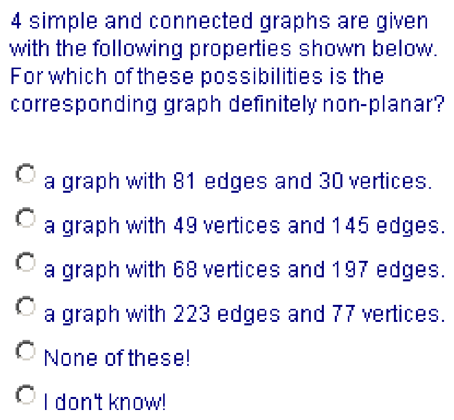


Figure 3.26 An example of a MC question, asking to determine which set of data corresponds to a graph that is non-planar.

Note that the number of edges and the numbers of vertices are randomly positioned in each line; for each entry, a random "switch" variable determines the ordering of the number of vertices and the number of edges.

3.8.2 Finding a Planar Graph

The second question looks at the graphic form of the MC question. In the example in Figure 3.27, eight student teachers are asked to throw a ball of yarn to each other as part of an icebreaker game. After the last throw, the student teachers need to unravel themselves so that the graph formed using themselves as vertices and the yarn as edges is planar.

Even knowing the theorem (and even its proof), finding the correct answer here is quite tricky, unless the randomised graphs that appear make it obvious to deduce whether or not K_5 or $K_{3,3}$ appear. With this question, though, there are also three different scenarios that can randomly appear.

The scenario in Figure 3.28a details a maze that must not have any overlapping edges. However, all of the options will (normally) have overlapping edges. Therefore, each of these graphs would have to be “edited” first in order to attempt to get a planar “maze”. In Figure 3.28b, the key information from Kuratowski’s Theorem is given. However, in this case, the information is detailed to show that K_5 and $K_{3,3}$ are special graphs themselves; an online reference is also included as part of the detailing.

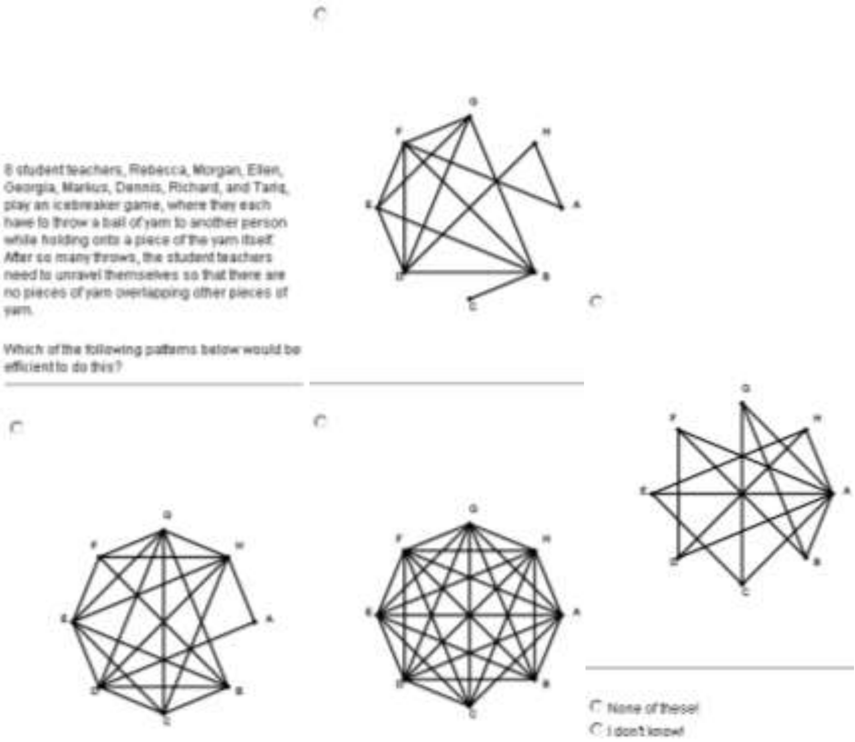


Figure 3.27 A question relating to an icebreaker game that is asking to find a planar graph.

This question, regardless of which scenario exists, is still somewhat difficult as students have to find one of the two key subgraphs in order to eliminate it as a candidate for being a planar graph. What makes this even more difficult for some students is the uncertainty factor, where students who cannot find neither a K_5 nor a $K_{3,3}$ subgraph may think they have made an error and instead of assuming they have found the correct answer, will continue looking for the subgraphs to appear in the last of the four choices.

This question is worth 4 marks, but if a student replies with “I don’t know!”, then (s)he will receive one mark; note, though, that students should ***not*** be made aware of this as they may attempt to fast-track the question otherwise.

A **utility graph** is the complete, bipartite graph, $K_{3,3}$, that originates from attempting to connect three utilities from the same sources to three houses. Also, the complete graph, K_5, is helpful in explaining the **Golden Ratio** in The Fibonacci Sequence as the pentagram inside the graph exhibits properties relating to the Fibonacci Sequence. (NOTE: More information on this can be found at http://mathworld.wolfram.com/GoldenTriangle.html .).

a) Jess wants to create a maze between 8 key points such that, when designing it, there are no overlapping edges. Jess has 4 candidates, as shown below. Which of these graphs can be edited to form such a maze?

b) These two types of graphs are embedded within, at least, three of the graphs shown below. However, which of them, if any, does not have either of these graph types embedded within it?

Figure 3.28 Examples of the two additional scenarios for the MC question on finding planar graphs: a) Maze creation, and b) Utility graph and Golden Ratio.

3.8.3 Finding a Planar Graph when Adjacency Matrices are Included

It is clear that visualisation may be a key factor in determining the planarity of a graph as it is easier to see the connections that make a K_5 or $K_{3,3}$ graph appear. Therefore, these questions are more challenging as students first need to draw the graphs corresponding to each of these adjacency matrices before determining whether or not any of them are planar.

Also, as with the previous question, 1 mark will be awarded if a student replies with “I don’t know!”, but again, they should not be made aware of this in case they decide not to attempt the question altogether.

3.9 Spanning Trees

There are nine questions involving spanning trees. Similar to previous topics, these questions include copies of questions with graphs changed either into adjacency matrices or with graphs and adjacency matrices appearing together. Therefore, only three examples are provided, but they are more independent than many of the questions in the previous topics as each question provides different information, requiring students to think differently each time about how to solve these problems.

3.9.1 Finding a Spanning Tree

3.9.1.1 MC Questions on Finding Spanning Trees

Four MC questions have been created involving finding a spanning tree for a particular graph. In each instance, either a graph or an adjacency matrix is given and the student is required to look at four candidate solutions to see which of them resembles a spanning tree for the given graph or adjacency matrix. Similar to previous questions, these combinations require the student to remain aware of visual and logical skills needed in graph theory so that (s)he may perform better on assessments. The four questions created have the following combinations of graphs and adjacency matrices:

- Graph is given. Candidates are graphs.
- Graph is given. Candidates are adjacency matrices.
- Adjacency matrix is given. Candidates are adjacency matrices.
- Adjacency matrix is given. Candidates are graphs.

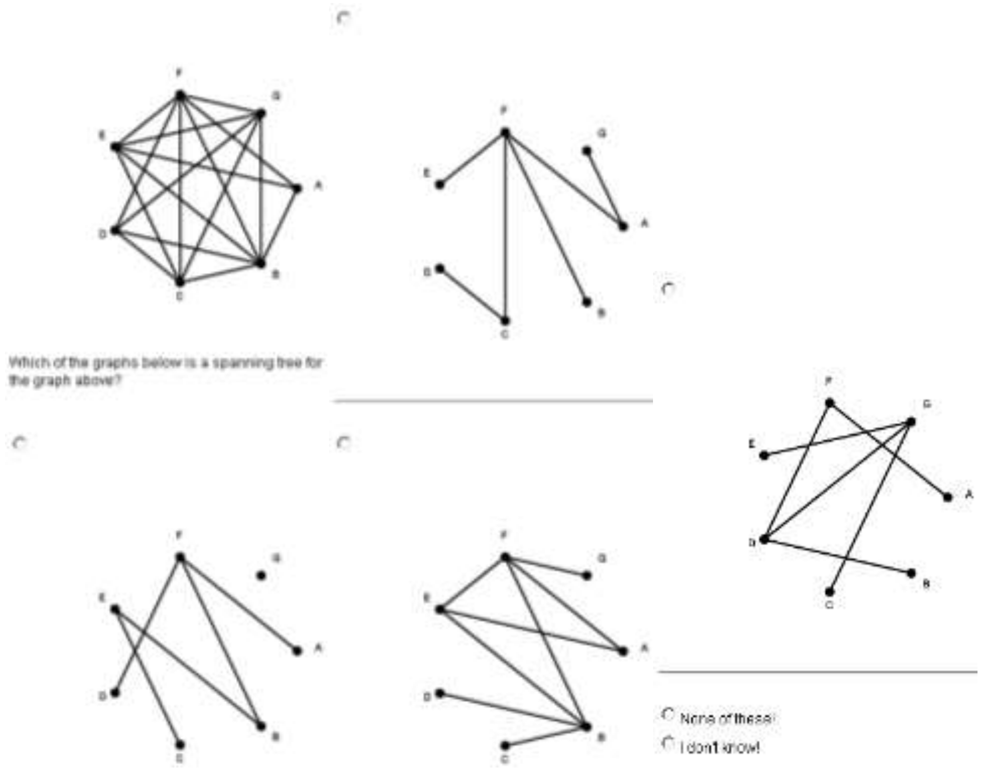


Figure 3.29 A directed MC question using only graphs and asking to find a spanning tree for the given graph.

The example given in Figure 3.29 is a directed question involving only graphs. For each combination, there are four possible scenarios; the other three scenarios are all indirected and are shown in Figure 3.30. With four scenarios in each question and four combinations of intelligences also, this question has a lot to offer to students in terms of mathematical and pedagogical assistance. Furthermore, the additional, indirected scenarios help to see how this topic is more useful in real world situations.

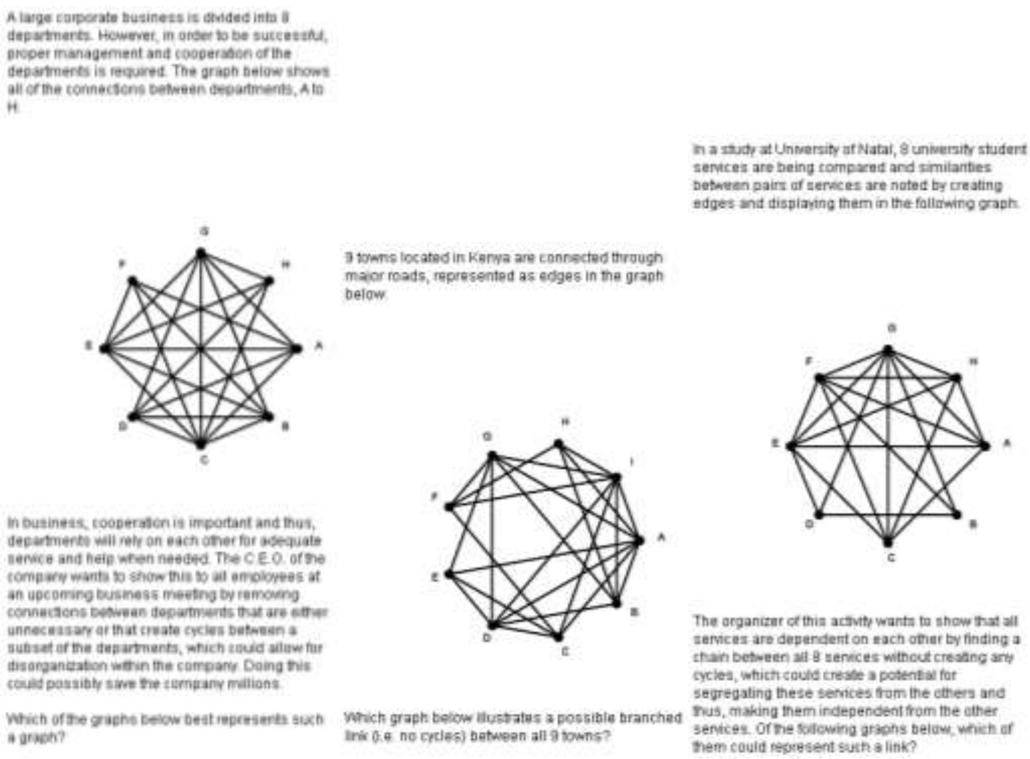


Figure 3.30 Examples of the three indirected scenarios for the MC question on finding a spanning tree for a given graph. From left to right, the scenarios are: business departments, link between towns, university student services.

3.9.2 Determining The Number of Spanning Trees in a Graph

The number of spanning trees in a graph can be calculated using the determinant of the difference between a graph’s corresponding degree matrix and its corresponding adjacency matrix. Therefore, it is possible, using this method, to use the numerical input (NI) question type rather than just using multiple-choice (MC) questions. However, it is better to use responsive, numeric input (RNI) as this not only allows for additional feedback to be given when a distracter is used, but also, partial marking can be implemented with RNI questions, unlike NI questions, where the answer is either correct or incorrect (i.e. all-or-none marking). The problem now is to determine what makes a useful distracter for this question.

3.9.3 Number of Spanning Trees for Graphs with Bridges

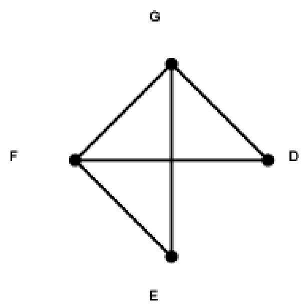
Three questions have been generated, asking the student to determine the number of spanning trees in a graph given a selection of branched subgraphs, along with a given number of copies for each subgraph. As with other topics, one question uses only graphs, one question uses only adjacency matrices, and one question uses both graphs and adjacency matrices. However, unlike the previous, like questions, which were all MC questions, these are all RNI questions. An example of the mixed scenario is given in Figure 3.31.

A professor at Stellenbosch University asks the students to look at a particular graph. The professor then asks them to determine which subgraphs are connected together, either by bridges or by single vertices, to create the larger graph. The students correctly determine that the subgraphs are as follows:

2 copies of

	A	B	C
A	0	1	1
B	1	0	1
C	1	1	0

3 copies of



2 copies of

	H	I	J	K	L
H	0	1	0	1	0
I	1	0	1	1	1
J	0	1	0	0	1
K	1	1	0	0	1
L	0	1	1	1	0

The professor then points at you, a student in his class, and asks you to determine the number of spanning trees in the original graph. What will be your answer to him?

Figure 3.31 Example of an RNI question, asking to determine the number of spanning trees in a given graph involving copies of branched subgraphs.

3.10 Minimal Spanning Trees

There are twelve questions created in this question set created by Zaczek. There are six questions for each algorithm. For each set of six questions, there are three questions each which use a graph of 5 – 6 vertices and three questions each which use a graph of 7 vertices. Within each subset of three questions, there are three different questions being asked, namely:

- Is a named edge included (or even considered) within the algorithmic process?
- What is the minimal spanning tree for the given graph?
- What is the n^{th} edge considered in the algorithmic process?

As there is a significant amount of sub-categorisation involved and since it is important to highlight these key features, it is sufficient for the purpose of this thesis to show just three examples that highlight all of the available features for all twelve questions.

One problematic issue in the programming of these questions is determining how to alert when a cycle is formed in the algorithmic process. Therefore, the graphs presented do not have randomised parameters embedded in the coding. Also, these questions are non-responsive (i.e. NI or WI questions).

Questions shown in this section illustrate the use of changing background colours, as mentioned in Section 1.3.1.

3.10.1 Kruskal's algorithm with a graph of 5 – 6 vertices, asking if an edge was considered

The example provided in Figure 3.32 shows an example of a question on Kruskal's algorithm involving a graph of six vertices to determine if a particular edge was considered in the algorithmic process.

The inclusion of the weighted matrix is important as some of the edges are overlapping with the weights in the graph. Clear instructions are provided for answering this question, showing students exactly how to type in their responses. Feedback to this question is provided in Figure 3.33. Feedback is detailed with step-by-step instructions on how the algorithm should have been implemented in the question. SVG graphics illustrate the step-by-step instructions nicely, illustrating the inclusion of edges without forming a cycle until all vertices have been connected.

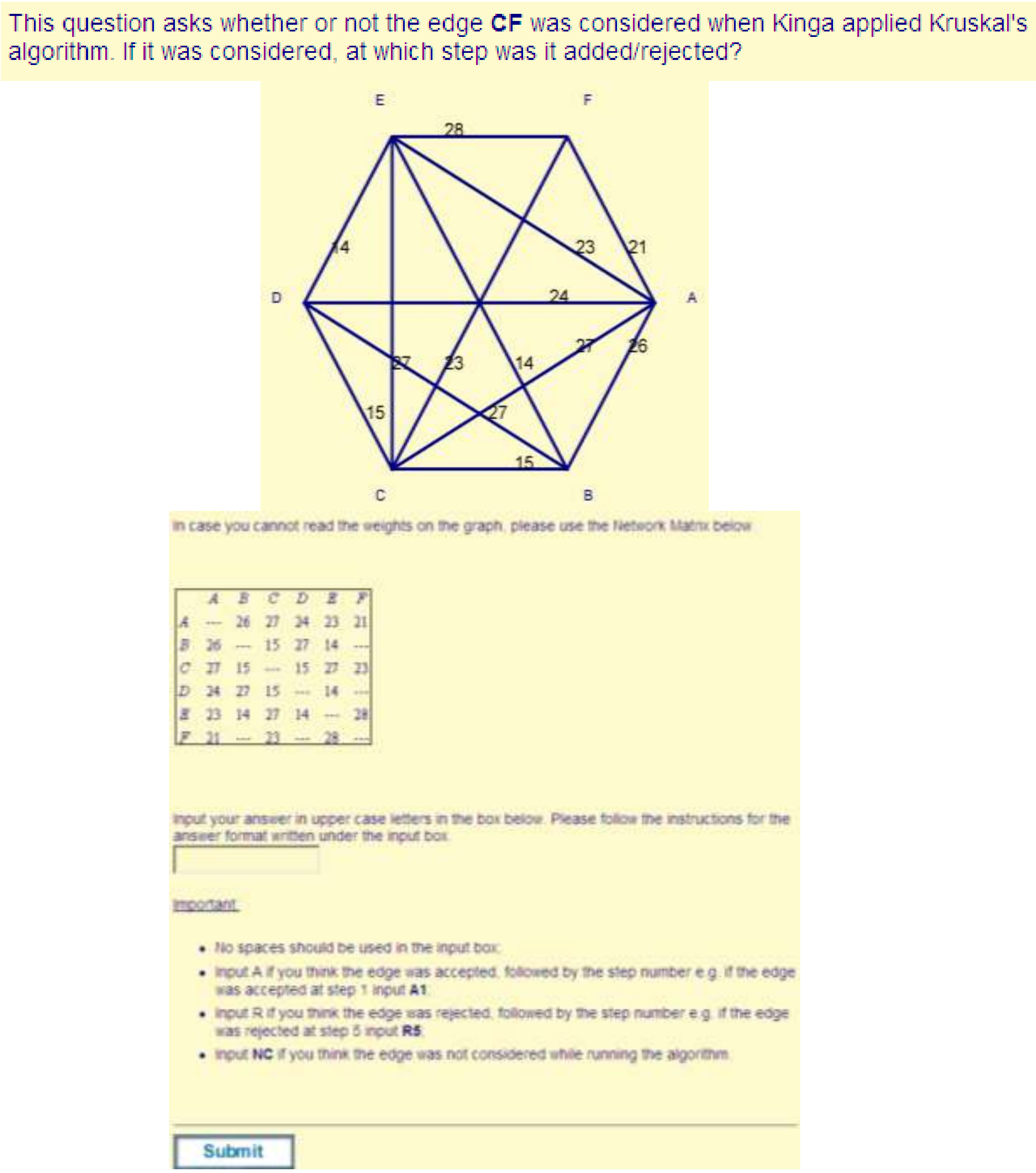


Figure 3.32 An example of a question on Kruskal's algorithm, asking to determine if the edge, $\overline{CF}$, was considered in the implementation of Kruskal's algorithm. Different background and text colours were used, highlighting the possibility of accommodating different students, as discussed in Section 1.3.1.

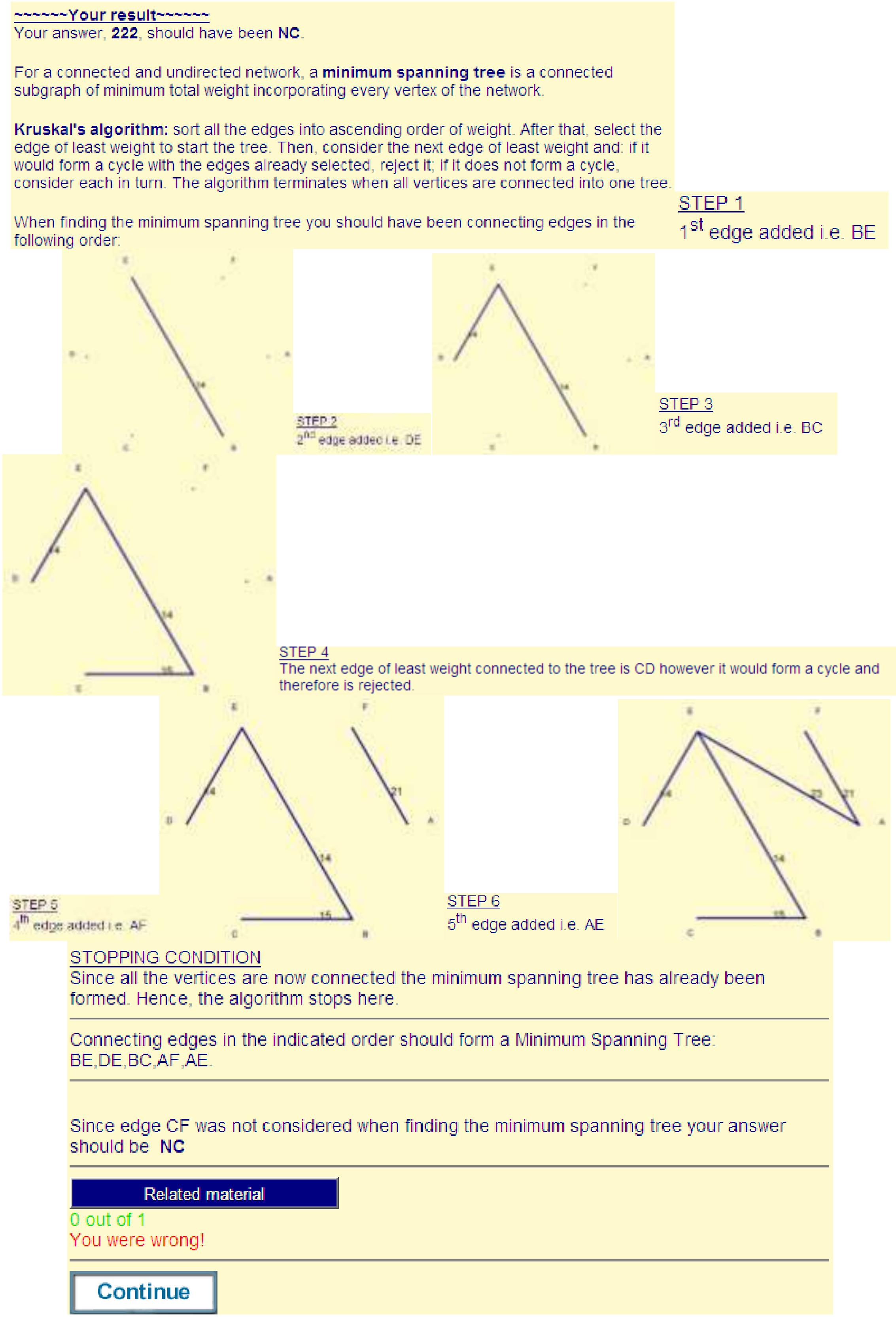


Figure 3.33 Feedback to the question presented in Figure 3.32.

3.10.2 Kruskal’s algorithm with a graph of 7 vertices, asking to find the minimal spanning tree

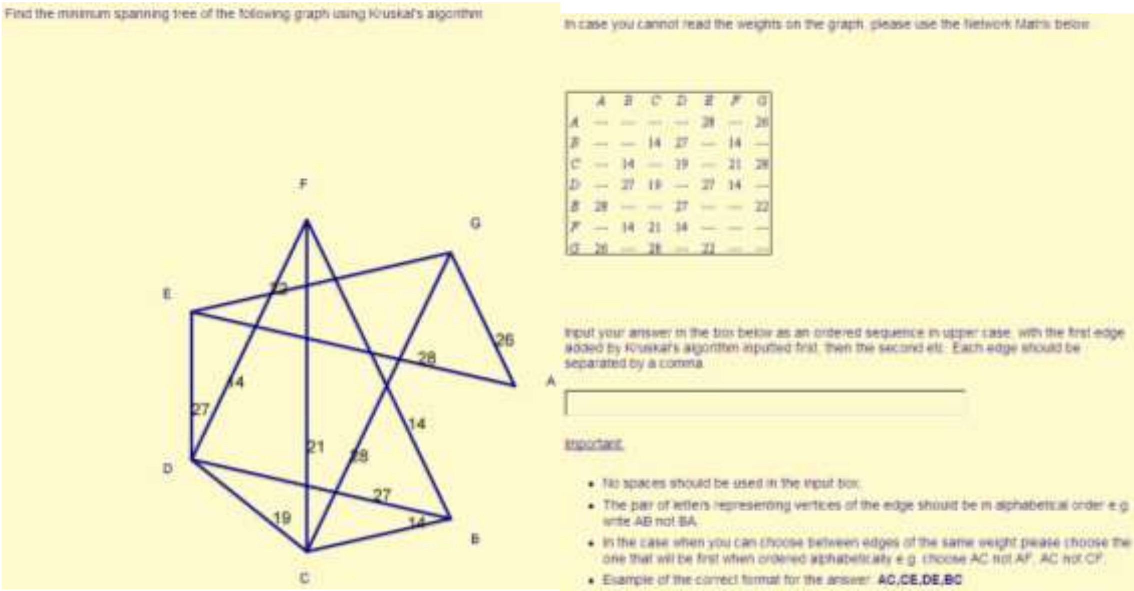


Figure 3.34 An example of a question using Kruskal’s algorithm on a graph of seven vertices, asking to determine the minimum spanning tree for the given graph.

The example provided in Figure 3.34 asks to find the minimal spanning tree for a given graph of seven vertices. Again, instructions are provided to remind students how to answer questions properly. The weighted matrix is again provided to ensure students can see the respective weights of the edges shown in the graph. Similar feedback to that shown in Figure 3.33 is provided, showing students how to use Kruskal’s algorithm properly in answering this question; upon completion of the algorithm, the answer is provided.

3.10.3 Prim’s algorithm with a graph of 7 vertices, asking to find the n^{th} edge added in the algorithmic process

The example provided in Figure 3.34 asks to find the sixth edge added in the implementation of Prim’s algorithm for a graph of seven vertices. Because Prim’s algorithm has a particular focus on vertices, the starting vertex is provided in the question. Again, instructions are provided on how to format answers and a weighted matrix is also provided to help students read the weights more efficiently. There is additional randomisation in choosing the edge which students need to find (i.e. 6^{th} edge) in the question.

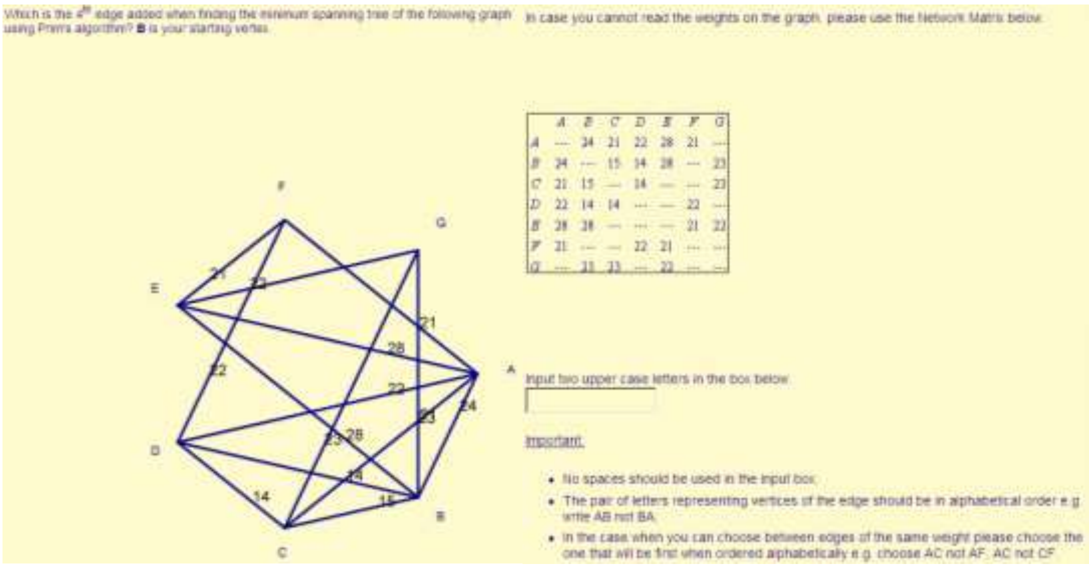


Figure 3.35 An example of a question using Prim's algorithm on a graph of seven vertices, asking to determine the sixth edge added as part of the algorithmic process.

The feedback is provided in Figure 3.36. Similar to the questions involving Kruskal's algorithm, feedback on Prim's algorithm is detailed, using SVG graphics to illustrate the algorithmic process to completion. A worded explanation discusses how the algorithm works throughout.

3.10.4 Challenges to Kruskal's and Prim's Algorithms

As previous noted, there are a couple of notable issues with the creation of these questions. These are WI questions and not RWI questions; especially in the case of determining the minimal spanning tree, the lack of use of a pop-up window may become an issue for students; this will be analysed later in Chapter 7 and Chapter 8. Also, there is less randomisation than usual, limiting the use of graphs to just two fixed structures. There is some randomisation elsewhere in these questions, but students will be familiar with the two set structures if they practise these questions in advance prior to an assessment taking place.

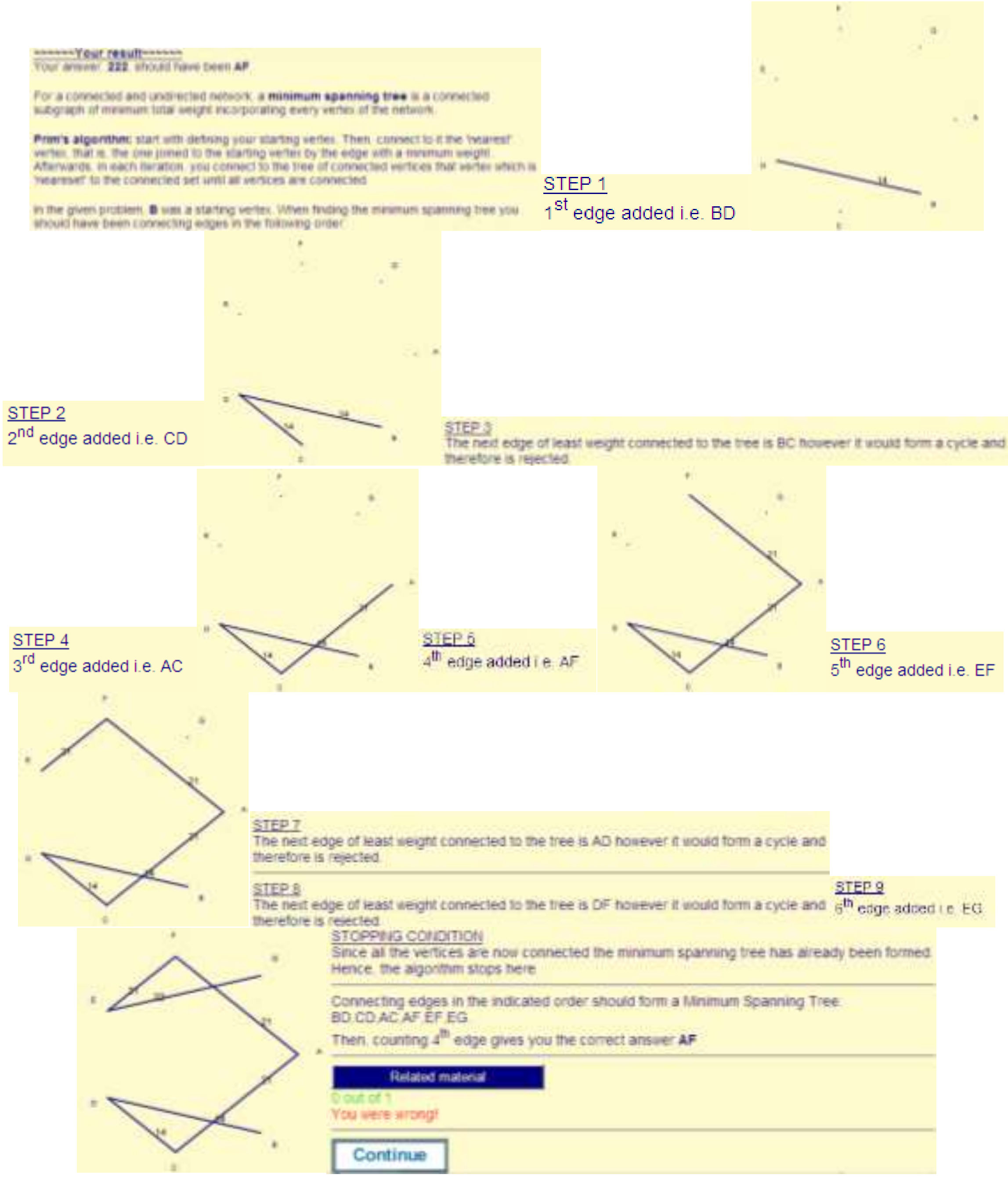


Figure 3.36 Feedback to the question presented in Figure 3.35.

3.11 Shortest Path Algorithm

All questions in this section involve directed edges, each pointing in the *forward* direction. If *backwards* edges were included, then the procedure shown in Example A.4 may not work since any backwards edge would cause previous calculations to then be verified and possibly re-assessed.

3.11.1 Calculating the Shortest Path

The shortest path problem is very interesting mathematically, but is also very time consuming computationally as it involves the use of complete enumeration[52], calculating all possible paths and their distances. With a 32,000 character limit and “pedagogical” time constraints (e.g. time-limited testing), creating a suitable code for Dijkstra’s algorithm must involve accommodations.

First, it is important to note that the design of questions involves a careful manipulation of directed edges and vertices so that the “starting vertex” is located to the left, with the vertex labelled O, and all movement of directed edges goes from left to right, with vertices labelled in alphabetical order from top to bottom, left to right, and the terminating vertex being labelled T; additional information about this can be found in Section A.11. Knowing these issues helps to better understand the technical side of this algorithm. However, the graphical display of the network is not the only technical element to this question because the solution requires a delicate method for calculating path lengths and determining the shortest path.

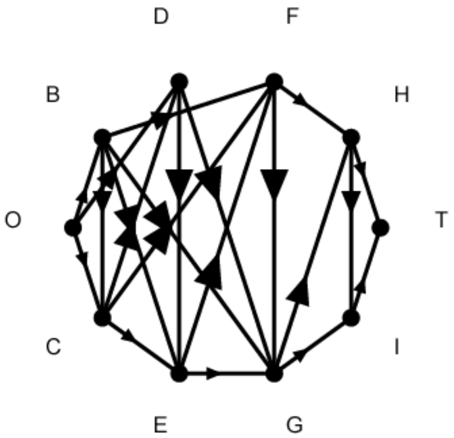
3.11.2 Distracters in a 2RNAI Question on The Shortest Path Algorithm

The other problem posed in this section is identical to the previous problem, but asks more questions, requires more from students, and provides much to students in terms of conditions for earning a better grade. With this problem, two questions are given. The first problem always asks for the minimum distance to be travelled from the origin, *O*, to the terminal, *T*. However, the second question requires the answer to the first question as it asks to either find the amount of fuel needed to travel between these two destinations or for the cost of the fuel used to travel between them. As such, this is a sequential, 2-Responsive Numeric Input (sequential 2RNI) question. An example with the second question asking for the amount of fuel needed is given in Figure 3.37.

In addition to this, though, because of the numerical data given to help answer the second problem, rounding errors may occur. As such, special attention needs to be given to this and so, this is also a Numeric Approximation Input (NAI) question. Therefore, this question is a sequential 2-Responsive Numeric Approximation Input (sequential 2RNAI) question.

Helen wants to travel between two cities in Ghana. However, the direct route involves a rather costly toll station and so, she wants to therefore find a different route by travelling via non-toll routes. A map she purchased provides all of the non-toll routes between 10 cities and towns, including the departure and destination cities, as shown below, along with a distance matrix representing the distances (in km) between the communities.

	O	B	C	D	E	F	G	H	I	T
O	---	51	64	80	---	---	---	---	---	---
B	---	---	57	---	59	58	90	---	---	---
C	---	---	---	51	55	63	---	---	---	---
D	---	---	---	---	55	---	80	---	---	---
E	---	---	---	---	---	51	60	---	---	---
F	---	---	---	---	---	---	51	64	---	---
G	---	---	---	---	---	---	---	55	60	---
H	---	---	---	---	---	---	---	---	55	69
I	---	---	---	---	---	---	---	---	---	58
T	---	---	---	---	---	---	---	---	---	---



Avoiding the toll route, if the average fuel consumption for the vehicle is 8.695652 kilometers per litre and the cost of fuel is equivalent to £1.9 per litre, then...

1. What is the minimum distance to be travelled from O to T?

Shortest distance = 259 km.

2. How much fuel would Helen use, at least, to drive along the route with the distance, as measured, in question 1?

Give your answer to, at least, 2 decimal places.

29.79 L

Figure 3.37 An example of a sequential, 2-Responsive Numeric Approximation Input (2RNAI) question, asking to find the minimum distance from the origin (labelled O) to the terminal (labelled T), along with the amount of fuel needed to travel this distance.

This question is worded so that after each question, an input box appears; this clearly shows the student which input corresponds to which question. Also, information is given to six decimal places regarding the vehicle's fuel consumption and the price is given as an "equivalency" value in British pounds sterling, which suggests that a given question could be representative of any set of locations in the world. Also, the value for the average fuel consumption is deliberately given to a large number of decimals as part of this problem involves testing the students' abilities in rounding properly when calculating.

For the problem in Figure 3.37, the answer given to the first question is incorrect; the correct answer should be 242 km. However, for the second question, with which the correct answer should be 27.83 L, the correct procedure was used, although it involved using the incorrect answer to the first question. For this problem, correct answers to both questions is worth 7 marks, but for the answers given in this problem, 5 marks are awarded as 2 marks are only removed for the incorrect answer in the first question.

The provision of additional feedback and partial marking is important for this problem as not only are there two questions to answer, but also, the pairing is sequential and one of the questions may be answered incorrectly due to improper rounding, thus causing approximations to be considered. Due to all of this, there are eight cases for warranting partial credit and/or additional feedback in this problem:

Given the following variables,

- **COR1** is the correct solution to the first question
- **COR2** is the correct solution to the second question
- **ANS1** is the student's answer to the first question
- **ANS2** is the student's answer to the second question
- **DIS1** is the first distracter (for the first question)
- **DIS2** is the second distracter (for the first question)
- **DIS3** is the third distracter (for the second question)

1. $ANS1 = COR1$, but $\left|\frac{ANS2 - COR2}{COR2}\right| \leq 0.01$. 6 marks are awarded. The error involved is a simple rounding error in the second question, although just minor.
2. $ANS1 = DIS1$, and $\left|ANS2 - DIS3\right| < 0.05$. 3 marks are awarded. The first answer is wrong, but triggers a distracter. The second answer is also wrong, but it, too, triggers a distracter.
3. $ANS1 = DIS1$, but $\left|ANS2 - \left(\frac{ANS1}{COR1} \times COR2\right)\right| < 0.005$. 5 marks are awarded. The first answer triggers a distracter, but based on this answer, the

second answer given appears to be correct within 0.5%. Therefore, the second answer is deemed to be correct.

4. $ANS1 = DIS2$, but $\left|ANS2 - \left(\frac{ANS1}{COR1} \times COR2\right)\right| < 0.005$. 5 marks are awarded.

   The second answer triggers an alternate distracter, but using this answer, the second answer appears to be correct within 0.5%. Therefore, the second answer is deemed to be correct.

5. $ANS1 = DIS2$, but $\left|ANS2 - \left(\frac{ANS1}{COR1} \times COR2\right)\right| \le 0.01$. 4 marks are awarded.

   The student's first answer triggers an alternate distracter. However, based on this information, the student's second answer appears to be correct within 1%. Therefore, the second answer is deemed to be "almost correct".

6. $ANS1 = COR1$, but $\left|\frac{ANS2 - DIS3}{DIS3}\right| < 0.005$. 6 marks are awarded. The first answer is correct, but the second answer triggers a distracter.

7. $ANS1 = COR1$, but $ANS2 \ne COR2$. 5 marks are awarded. The first answer is correct, but the second answer is incorrect and is not even close to the correct solution.

8. $ANS1 \ne COR1$, but $\left|\frac{ANS2 - COR2}{COR2}\right| < 0.005$. 2 marks are awarded. The first answer is completely incorrect. However, somehow, the student's second answer appears to be correct.

Note that *DIS1* and *DIS2* are distracters that relate to the appearance of other entries in the last column of the resulting matrix using the shortest path algorithm. These values may not appear; if this is the case, then the distracters are ignored. Also, *DIS3* involves placing the average fuel consumption in the wrong place during the calculation of the second answer. As there are two possibilities for the second part of this problem, there are two variations of *DIS3*:

- **(average_fuel_consumption/cost_of_fuel)*minimum_weight[n]**, if asking for the cash needed to travel the shortest distance.
- **(average_fuel_consumption/minimum_weight[n])**, if asking for the amount of fuel needed to travel the shortest distance.

As noticed from the eight cases, a correct answer to the first question is worth 5 marks and a correct answer to the second question is worth 2 marks. However, due to rounding issues with the software, it could be possible for a student's answer to the second problem to be close, but not enough to award him/her the 2 marks associated with it. Also, note that to answer the second question correctly, the only mathematical operations needed are multiplication and division. Therefore, since the equation, $\left|ANS2-\left(\frac{ANS1}{COR1}\times COR2\right)\right|$, looks at a student's wrong answer to the first question and then compares the student's second answer to the product of the correct answer to the second question and ratio of the student's first answer to the correct answer, it can be used to see if the student's second answer is, at least, following the proper methodology.

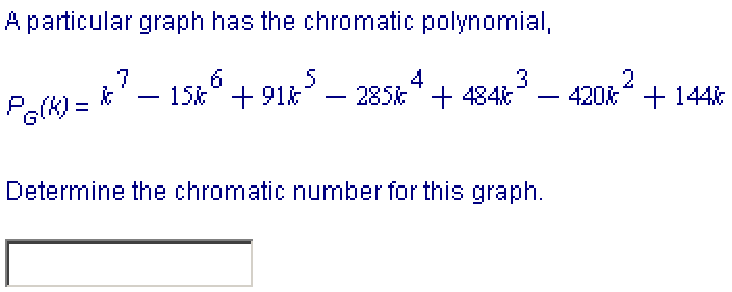
3.12 Vertex Colouring

There are four questions available in this topic, but unlike other topics, these questions only use graphs and do not ask similar questions. Each question therefore requires students to think differently about what it is they are trying to solve.

3.12.1 Chromatic Numbers using Chromatic Polynomials

The first question in the set is somewhat unique as there are neither graphs nor adjacency matrices in it. Instead, a polynomial is given and students are required to use this to determine the chromatic number of the corresponding graph. An example of this question is shown in Figure 3.38.

Recall that a chromatic polynomial determines the number of ways with which to colour a graph using so many colours. Therefore, for each value of $k\in\mathbb{Z}^{+}$, the value, $P_G(k)$, will determine the number of different colourings using *k* colours that can be created. For this particular question, starting at $k=1$, we obtain the results shown in Table 3.1. Notice that for $1\leq k\leq 4$, $P_G(k)=0$. Therefore, it is not possible to properly colour the corresponding graph with these numbers of colours. However, for $k=5$, $P_G(k)=720$, which implies that with five colours, the corresponding graph may be coloured in any of 720 different ways. As this is the smallest number of colours with which to colour the corresponding graph, the chromatic number must be five. However, do note that with six colours, there are 8,640 different combinations available for colouring the graph.

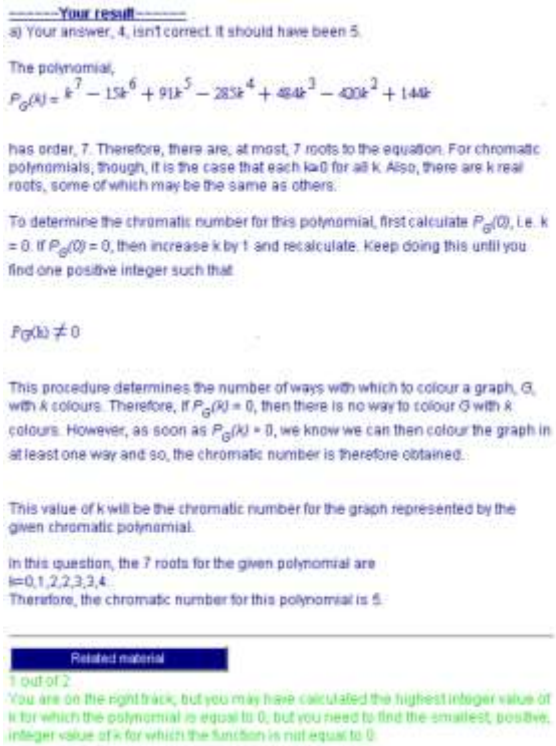


A particular graph has the chromatic polynomial,

$$P_G(k) = k^7 - 15k^6 + 91k^5 - 285k^4 + 484k^3 - 420k^2 + 144k$$

Determine the chromatic number for this graph.

Figure 3.38 An example of an RNI question, asking to find the chromatic number of a graph using its corresponding chromatic polynomial.



--------Your result--------

a) Your answer, 4, isn't correct. It should have been 5.

The polynomial,

$$P_G(k) = k^7 - 15k^6 + 91k^5 - 285k^4 + 484k^3 - 420k^2 + 144k$$

has order, 7. Therefore, there are, at most, 7 roots to the equation. For chromatic polynomials, though, it is the case that each k≥0 for all k. Also, there are k real roots, some of which may be the same as others.

To determine the chromatic number for this polynomial, first calculate $P_G(0)$, i.e. k = 0. If $P_G(0) = 0$, then increase k by 1 and recalculate. Keep doing this until you find one positive integer such that

$$P_G(k) \neq 0$$

This procedure determines the number of ways with which to colour a graph, *G*, with *k* colours. Therefore, if $P_G(k) = 0$, then there is no way to colour *G* with *k* colours. However, as soon as $P_G(k) \neq 0$, we know we can then colour the graph in at least one way and so, the chromatic number is therefore obtained.

This value of k will be the chromatic number for the graph represented by the given chromatic polynomial.

In this question, the 7 roots for the given polynomial are k=0,1,2,2,3,3,4.
Therefore, the chromatic number for this polynomial is 5.

Related material

1 out of 2
You are on the right track, but you may have calculated the highest integer value of k for which the polynomial is equal to 0, but you need to find the smallest, positive, integer value of k for which the function is not equal to 0.

Figure 3.39 Feedback and scoring provided for answering the question in **Figure 3.38** and triggering a distracter in the process.

This question is worth 2 marks and has one simple distracter in that it looks instead for the highest value of *k* such that $P_G(k) = 0$; this distracter will award students with 1 mark and an example of this, following from Figure 3.38, is shown in Figure 3.39. This distracter may be performed because students are accidentally looking for this rather than the smallest value (i.e. first value they will probably count if counting from 1 onwards) of *k* such that $P_G(k) \neq 0$.

k	$P_G(k)$		*k*	$P_G(k)$
1	0		*4*	0
2	0		5	720
3	0		6	8640

Table 3.1 Table of values for $P_G(k)$ for chromatic polynomial in **Figure 3.38**.

3.12.2 Finding the Chromatic Number using a Given Procedure

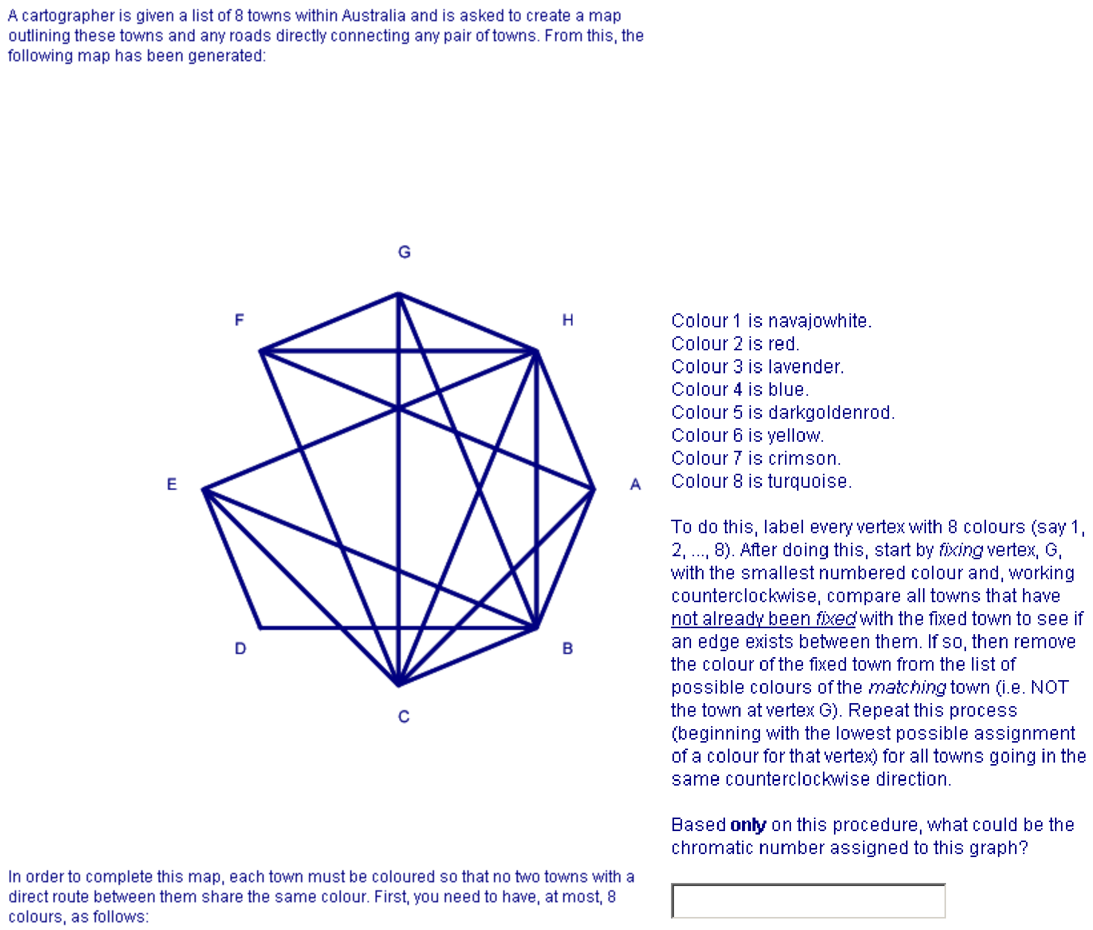


Figure 3.40 An example of an RNI question, asking to find the chromatic number of a graph using a given procedure.

The next question asks to find a candidate chromatic number for a graph, given a specific procedure for selecting vertices. Any solution obtained would automatically constitute an upper limit to the true chromatic number for the graph. Also note that the procedure employed is identical to that of Example A.5.

Note the procedure; it is asking to start at vertex, *G*, and then move counterclockwise (or anticlockwise). Just like the graph itself, the procedure in this question is randomised so that a student can be expected to start at any vertex and move in one of three ways, namely clockwise, counterclockwise, or in a particular sequence (which is given to students as part of the question). Additionally, for *n* vertices, *n* colours are also given in each question and are randomly generated from a default set of colours. The question itself is also randomised, but among a set of two different wordings; the other wording is: *Using this procedure, what is the upper bound for the chromatic number for this graph?*

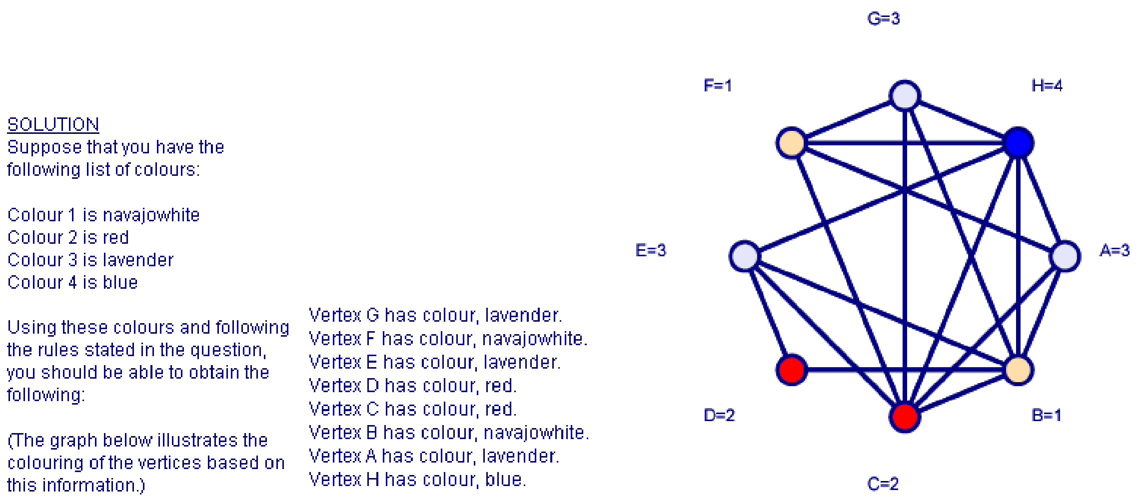


Figure 3.41 Feedback to the question presented in **Figure 3.40**, which includes a fully coloured graph and new labels.

The thing that makes this question special, however, is not what happens in the question, but rather what happens in the feedback, as shown in Figure 3.41.

In the feedback, the graph has changed dramatically to show the colouring of the vertices based on this procedure. The sizes of the vertices themselves have been enlarged to “boldly” show the colouring of the vertices. The labels of the vertices have changed, too, but this is especially important to consider as some students may have some degree of colour blindness, causing them to misinterpret two colours as being identical. To accommodate this likely possibility, each vertex label identifies which colour in the list, as noted in the feedback, is used to represent it. Being able to accommodate students' needs is very important and also, it helps to show the flexibility of the software to adapt to such needs.

The code to generate this special graph type simply uses the original graph code and attaches colours to the vertices from within the code. However, as the sizes of the vertices had to change in the feedback to show the colours properly, a new function was created to ensure all features worked properly.

3.12.3 Labelling a Vertex with a Particular Colour

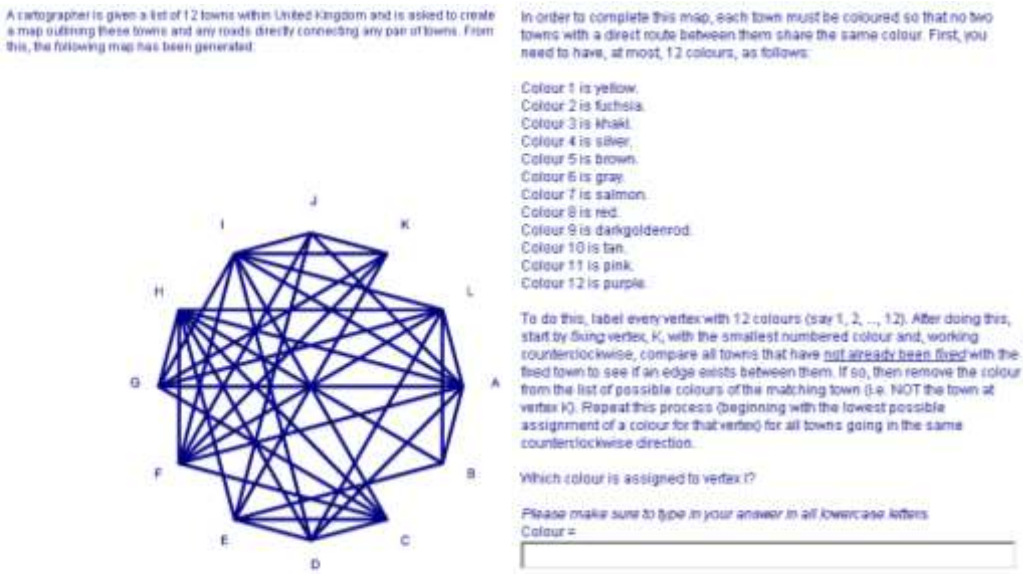


Figure 3.42 An example of a Responsive Word Input (RWI) question, asking to find the colour associated with vertex, *I*, using a given procedure.

The next problem is similar to the previous problem, but asks a different question. An example of this question, which asks to find the colour associated with a particular vertex, can be seen in Figure 3.42.

This is a Responsive Word Input (RWI) question and asks the student to find the colour that would be given to a particular vertex. As this is a RWI question and similar to previous RWI questions, an alert box will appear, asking the student to double-check his/her answer and then to hit the submit button again if (s)he is happy with it or otherwise, edit the answer and re-submit, knowing that the alert box will reappear.

When performing this algorithmic process for colouring vertices, order is important. However, in terms of getting students to attempt this question properly, it does not help to ask for the colour of a vertex that the student will encounter early in the algorithm. For instance, for the example in Figure 3.42, it would not take as much effort to determine the colours of the vertices, G to J, as in comparison to the other vertices. However, as noted in the previous section, the ordering of vertices is randomly chosen from one of three patterns. Therefore, to modify the question to the benefit of the assessor (e.g. teacher or lecturer), some accommodations have been made:

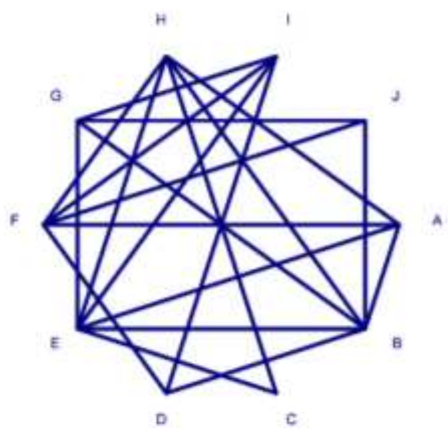
- For a completely random ordering, a vertex towards the end of the list is selected for which students need to determine the colour.
- For either a clockwise or counterclockwise (i.e. anticlockwise) colouring, if the distance between the target vertex (i.e. the vertex with which the student is to determine its colour) and the starting vertex is less than three, then the target is shifted down the list by a factor of one-third. For instance, if there are twelve vertices and the target vertex is the third vertex to appear in the list, then a new target vertex is chosen to be the seventh vertex (i.e. $3 + 4 \left(\frac{1}{3}(12)\right) = 7$).

3.12.4 Colouring all Vertices of a Graph

The final question in this set is, again, identical to the question on determining the number of colours needed to colour a graph using a given procedure. This is a RWI question and an example of this question is found in Figure 3.43.

The most important thing in this question is the ability to read the question properly. Instructions are given immediately above the answer box, similar to the other RWI question, telling students how to input their answers. Also, upon submitting this answer the first time, an alert box will appear, asking students to double-check their answers. However, also important is knowing the order in which the sequence of colours is to be generated as it is not necessarily in alphabetical order or in reverse order as both of these are actually distracters hidden in the question. If either distracter is triggered, then the student who triggered this distracter will receive 2 marks out of 4 for this question. Other than these differences, all other features of this question are similar to the previous questions (except the question on chromatic polynomials).

A cartographer is given a list of 10 towns within Pakistan and is asked to create a map outlining these towns and any roads directly connecting any pair of towns. From this, the following map has been generated:



In order to complete this map, each town must be coloured so that no two towns with a direct route between them share the same colour. First, you need to have, at most, 10 colours, as follows:

Colour 1 is silver.
Colour 2 is yellow.
Colour 3 is chocolate.
Colour 4 is darkgoldenrod.
Colour 5 is lime.
Colour 6 is maroon.
Colour 7 is blue.
Colour 8 is gray.
Colour 9 is pink.
Colour 10 is brown.

To do this, first label every vertex with all 10 colours (say 1, 2, ..., 10). After doing this, you then generate the following list:

C,H,F,B,D,G,I,A,E,J

Starting with the first vertex in the list, which you assign the smallest numbered colour in its list, compare all other towns following it in the list (i.e. NOT preceding it) to see if any edges exist between them. If so, then remove the colour from the list of possible colours of the matching town (i.e. NOT the town at vertex C). From there, go to the next vertex in the list and follow the same procedure (beginning with the lowest possible assignment of a colour for that vertex) until you reach the end of the list. Once finished, give the list of colours, starting with the colour number of the first vertex in the given list of vertices, followed by the second vertex, etc.

*Please make sure to type in your answer in all lowercase letters and leave **only** a comma between each colour (i.e. NO spaces).*

Coluring set = ( )

Figure 3.43 An example of a Responsive Word Input (RWI) question, asking to generate a particular colour sequence for the corresponding graph.

3.13 Research Question: Question Features

This section answers the research question:

What question features exist that could change how students interact with questions?

Chapter 3 discussed relevant features that can be implemented within questions in Mathletics on the subject of graph theory.

Graph theory makes excellent use of graphs and adjacency matrices within topics. The implementation of both graphs and adjacency matrices within similar questions created a wider range of questioning that can compel students to better study and understand the relationship between graphs and adjacency matrices within graph theory.

Different question types provided different techniques for answering questions. Multiple-choice (MC) questions provided opportunities to select a given answer, whereas Numerical Input (NI) and Word Input (WI) questions required solving and typing in answers. WI questions involved entering text in a very precise format in order to be evaluated fairly. Additional instructions on how to format answers within questions is required to avoid possible conflicts with answers that could be assessed unfairly. However, some questions were also created so that a pop-up window could remind students to double check their answers before hitting the Submit button a second time around. Not all word input questions provided the pop-up window and so, brought about an additional feature that could be assessed later. Although some research suggested that there was no difference in the assessment of MC questions and NI/WI questions, this had not been assessed for questions in graph theory, so it was helpful to see if a pattern change existed for this subject.

Questions were designed with some questions directly asking students to answer the questions and other questions providing word problems with students then being required to interpret the word problems into mathematical problems that could then be solved. Similar questions were designed in some topics so that a variety of question wordings could be provided; a teacher or lecturer could be more interested in asking students to solve word problems in context to show students the practicality of the learning material outside the classroom environment.

Chapter 4 Distracters in Online Assessment

4.1 Introduction

In the design of questions, it is important to consider how students may make mistakes in answering questions; this helps to remind question designers of the importance of the question structure to avoid further issues from occurring within online assessments. Some mistakes students make are minor and when questions are worth multiple marks, it is fair to award some partial credit. However, in online assessment, any such opportunities to award partial credit need to be carefully programmed into the question coding.

In this thesis, incorrect answers are called **distracters**. Not all distracters warrant being included in the coding of a question; for instance, if a distracter is to be credited with zero marks, then it is usually better to not mention the distracter in the question coding at all, especially as character limits within question coding may interfere with the possibility of including other distracters. If a distracter appears to be used by many students, then it may be worth considering this within the question coding. Also, distracters that could be worth partial credit may be worth considering within question coding, provided that partial credit can be awarded at all.

An issue that may occur when creating distracters is accidentally allowing correct responses to occur, although student methodologies may have been incorrect; as an example of this, substituting $x = 2$ into $2x$ instead of x^2 would still yield a correct response, even though the wrong formula was used in substitution. This is an issue that can occur in graph theory, too; for example, a question could ask a student which graph in a MC question has a Hamiltonian cycle and a graph with an Eulerian cycle as a distracter (to catch out for issues understanding the differences between the two cycles), but if the graph with an Eulerian cycle has not been checked properly, then it could, too, have a Hamiltonian cycle.

This chapter looks at various strategies implemented for finding distracter answers that could be used in the design of online questions for Mathletics. The list of strategies provided in this chapter is not exhaustive, but it does provide some insight into searching for distracters and why they are important when designing questions. It is expected that other strategies that may exist could provide more insightful information about why students select particular distracters over others; however, what is important for this thesis is simply the use of distracters within online assessment and not how distracters are chosen by students. Distracters are carefully considered to avoid problematic issues from occurring so that all distracters are unique. This chapter will explore the use of distracters using the graph theory questions designed in Mathletics, looking at specific questions to see how different distracters are considered. Some of the methods explained in this chapter take considerably more effort to research, but may be more valuable than other methods, which may appear to take less effort to research, but are considerably more difficult to obtain.

4.2 Comparable Questions

The first strategy looks at comparable questions to see what differences may appear in answering questions. Any significant differences may result in different marking schemes being used, but more importantly, using wrong strategies could trigger distracters and depending on how different the strategies are between comparable questions, partial marking could be awarded and additional detailed feedback may also be given.

From Section 3.1.3, the question on directed graphs is comparable to the questions on undirected graphs and the addition of weights. Unlike the other questions, this RNI question is worth three marks. Also, there are three distracters in place. The first distracter is simply misinterpreting indegree and outdegree and thus, calculating the number of edges along the row of the corresponding adjacency matrix when they should be using the column instead or vice versa; a student entering in an answer with this distracter will receive two marks. The second distracter uses the appropriate row or column, but instead calculates the sum of the entries from the given network matrix; as students are expected not to use the network matrix (from properly reading the question), a student suspected of using this distracter will receive one mark. However, the third distracter combines both of the previous distracters by using the network matrix and using a row instead of a column or vice versa. This could easily represent a student who obviously did not study beforehand and therefore, a student caught using this distracter will receive no marks.

4.3 Multiple-choice Questions

Multiple-choice questions require great detail in considering alternative answers to be presented as options. If a question has four possible answers and three of the possible answers are clearly incorrect, then a student may correctly answer a question without properly thinking about the learning material that led to understanding how to obtain the correct answer.

In Mathletics, multiple-choice (MC) questions go further, always providing “None of these” as an option. Within the question coding, it is randomly determined with a specified probability of occurrence (usually one time in eight occurrences) that “None of these” will be the correct answer. To design a MC question effectively in Mathletics, five answers need to be provided, namely one correct answer and four carefully designed distracters.

The question noted from Section 3.2.2, which looks at finding an appropriate adjacency matrix, is a MC question. The design of this question is simple in that the programming did not require much effort, aside from the already created functions for the graphs and adjacency matrices; in fact, all that is needed is one graph, its corresponding adjacency matrix (which is included in the programming of the graph itself), and a few other adjacency matrices. However, the problem with this question is finding suitable, *common* errors that students could make. This question only requires students to match up the connections to the adjacency matrices, all of which have the corresponding labels already attached. However, errors still need to be deliberately created for the distracters and thus, care is needed in designing randomised algorithms that give unique distracters in all realisations.

One error involves using ones to signify a “true/false” reaction to finding the adjacency matrix. A student may perceive anything connecting a pair of vertices to be “true”, thus entering the number, one, in the corresponding position of the adjacency matrix, even if there are multiple edges connecting the vertices.

The second error in this question places twos in the diagonals of the adjacency matrix where loops occur. This error has validity because although there is only one edge involved, the two ends of the edge are both connected to the vertex. Therefore, some students may input twos in these diagonal positions, thus creating the errors. This is a problematic issue as some people recognise the importance of using 2s in the diagonal as the Handshake Theorem works well using this convention, so teachers / lecturers need to discuss this first in class.

The third error is to create an asymmetric matrix, such as the third matrix shown in Figure 3.8. Even though there are no arrows to represent digraphs, it may be possible for students to be inclined to assume that the graph in question is indeed asymmetric, especially if they have been practising many questions on this topic. However, to create this error, the adjacency matrix needed to be the correct answer. Following this, two random, symmetric positions within the adjacency matrix, say $a_{i,j}$ and $a_{j,i}$, are made to be unequal, thus creating the error.

The fourth error has adjustments made to every entry in the adjacency matrix. However, for students who make honest attempts at the questions, they should never select this to be their answer. Therefore, if a student selects this to be his/her answer, then the reason for doing so can simply be because they just guessed the answer without looking at the question. In order to discourage students from doing this, a mark of -2 will be given for selecting this answer.

4.4 Notation Issues

It is possible within Mathletics for students to have issues with notation of answers. Word Input (WI) questions, in particular, may create problems for students if not designed carefully. Additionally, especially within the subtopic of directed graphs, alphabetical order may play a role in creating distracters.

The question described in Section 3.3.4 looks at edge sets for directed graphs. Upon attempting the other questions or simply by force of habit, students may still enter all of the edges with each entry being in alphabetical order. Therefore, for this question, a distracter answer is given with each edge listed in alphabetical order (e.g. $\overrightarrow{CF}$ instead of $\overrightarrow{FC}$). If a student enters his or her answer in that format, then a partial credit is given as many of the edges would still be correct.

Another possibility, although may seem far-fetched at first glance, would be for the students to input all of the answers in the wrong direction. As an example, in the above question, not only would a student input $\overrightarrow{CF}$ instead of $\overrightarrow{FC}$, but (s)he would also input $\overrightarrow{BA}$ rather than $\overrightarrow{AB}$ and so on. One reason why a student may do this would be that (s)he is getting confused with the understanding of notation when directions are included. Remember that students will not have the vector-like notation when inputting their answers, so they may think that CF could represent "going to C from F", thus convincing them to list this as part of the solution.

4.5 Making Use of Theorems

One method for creating distracters for a question is to look at mathematical theories related to the topic of the question. The use of theorems, corollaries, lemmas, etc. can help to create distracters that may be more challenging to notice.

From Section 3.5.1, a series of questions looks at finding non-Hamiltonian graphs. In generating these questions, it is important to consider what makes a good distracter. In the case of Hamiltonian graphs, there are some theorems that help by showing what properties can give away a graph as being Hamiltonian and as such, these theorems were implemented into the coding, either as distracters or as the correct solution.

First, it is important to note that the correct solution for the MC questions on Hamiltonian cycles is randomly selected amongst a list of three candidates. One of these candidates, however, was created based upon the following theorem[48]:

Theorem 4.1 Let G be a connected, bipartite, undirected graph with $V = V_1 \cup V_2$ and let $|V|$ be the number of vertices in G. If $|V_1| \neq |V_2|$, then G cannot have a Hamiltonian cycle.

There are also useful corollaries that may be implemented and in this instance, one of them helps to create a distracter, looking at the degrees of the vertices, *deg(v$_i$)*, of a graph[53].

Corollary 4.1 If $G = (V, E)$ is a loop-free, undirected graph with $|V| \geq 3$ and if $\deg(v) \geq \frac{|V|}{2}$ for all $v \in V$, then G has a Hamiltonian cycle.

Another useful corollary is the following, which looks at the number of edges in a graph rather than the number of vertices[48]:

Corollary 4.2 If $G = (V, E)$ is a loop-free, undirected graph with $|V| \geq 3$ and if $|E| \geq \binom{n-1}{2} + 2$, then G has a Hamiltonian cycle.

There are two other distracters used in these questions on Hamiltonian cycles, but they do not need any special distinctions as theorems, corollaries, or other terms. One of the distracters is simply a Hamiltonian cycle generated by creating an Eulerian cycle that uses all of the vertices in passing. The other distracter is a **wheel graph**, which will obviously be Hamiltonian as it is formed by creating a circle of vertices with one additional vertex in the center of the circle and connecting all of the other vertices. More on wheel graphs can be seen in an upcoming topic, which looks at isomorphisms.

From Section 3.8, a series of questions looked at finding planar graphs. Obtaining the distracters for this topic is easy to do as Kuratowski's Theorem allows two different subgraphs to interfere in the attempt to make a graph planar. For one distracter, only the K_5 subgraph needs to be included; in another, only the $K_{3,3}$ subgraph needs to be included. For a third distracter, both subgraphs are included. However, this does leave one additional distracter open. Therefore, special graphs are needed to create a correct answer.

Looking back at the questions on isomorphisms, it is easy to notice that four of the five special graphs mentioned, namely wheel graphs, prism graphs, antiprism graphs, and ladder graphs, are all planar graphs. Using these special graphs, one of four possible, correct answers may be generated upon a question's appearance. If a correct answer is to appear, then the computer coding will randomly select which of the four special graphs will represent this answer.

Returning to the distracters, the fourth distracter is a complete graph, but with one edge removed. In most cases, removing one edge will not make any difference. However, if there are only five vertices used in each answer, then, according to the theorem, this distracter will also be linear. Therefore, it is necessary that in every case, $|V| \geq 6$, where *V* is the set of all vertices used.

In Section 3.9.2, the questions designed looked at calculating the number of spanning trees in a graph. The essential calculation to be performed using the determinant method for finding the number of spanning trees is a difference between the $n \times n$ degree (*D*) and adjacency (*A*) matrices for the graph. However, do note that although $D - A \neq A - D = -(D - A)$, students will most likely still give the correct answer if they determine the determinant of a cofactor of *D* – *A* as either $\det^*(A - D) = \det^*(-(D - A)) = \det^*(D - A)$, where $\det^*(D - A)$ represents the determinant of any cofactor of the matrix, *D* – *A* and when *n* is even or $\det^*(A - D) = -\det^*(D - A)$, which will undoubtedly prompt students to input the absolute value, $|-\det^*(D - A)| = \det^*(D - A)$, as they will know that this value, which is essentially a counting variable, will be nonnegative. However, instead of subtraction, it is possible for students to accidentally perform an addition of these matrices. In this instance, $\det^*(D - A) \neq \det^*(D + A)$ and so, a distracter can be created.

Another essential feature of this calculation is knowing to calculate the cofactor of the matrix, *D* – *A*. However, it does seem possible for students to easily overlook this and instead of calculating $\det^*(D - A)$, they could calculate $\det(D - A)$. However, there is a problem, as Proposition 4.1 explains, but first, one term needs to be defined[48]:

Def. 4.1 Let *A* be an $n \times n$ matrix. Then, for $1 \le i, j \le n$, the **minor** associated with an entry, $a_{i,j}$, is the $(n-1)\times(n-1)$ determinant obtained by removing the i^th^ row and the j^th^ column of *A*.

Now, the proposition is given.

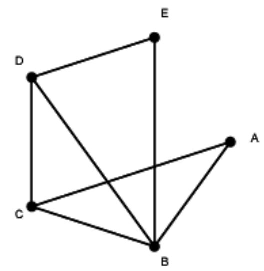
Proposition 4.1 For any simple, undirected $n \times n$ graph, *G*, let *D* represent its corresponding degree matrix and *A* represent its corresponding adjacency matrix. Then $|D - A| = 0$.

So, it is not reasonable to use this as a distracter, but it is possible to look at a particular minor of the adjacency matrix, *A*, and calculate the absolute value of its determinant. Therefore, this instead can be used as a distracter.

Following suit to the previous (and now proven to be faulty) distracter, another possible distracter can be formed by simply calculating the determinant of a cofactor of the adjacency matrix itself, especially at the diagonals, forgetting completely about the other, relevant parts of the calculation. The reasoning behind this distracter is because it has been observed at Brunel University that some students are still expecting a “quick solution” to mathematical problems, even at the postsecondary level. Although this may not be generic for all students, for this particular group of students, who are to be tested using these questions, it seems appropriate to use this distracter. Perhaps by using this distracter, too, other students in other institutions may be caught using it. Do note, though, that not all determinants of cofactors will be positive and therefore, the absolute value of each determinant has been obtained and each new result is used as a separate distracter. Additionally, it is possible that any of these results will duplicate another distracter in the code. However, based on how the coding of the distracters works, if a student gives an answer that could trigger multiple distracters, then only the first distracter triggered will be noted in the feedback. As such, it is always important to ensure a proper ordering of distracters in the code by allowing those that are more likely to occur by a student to appear in the code first, followed by the less likely distracters.

The three distracters noted have been implemented into two RNI questions, one involving a graph and the other involving an adjacency matrix. In both questions, students are simply asked to determine the number of spanning trees for either the graph or the adjacency matrix. An example of this question, along with a response that triggers one of these distracters is given in Figure 4.1.

How many spanning trees can be made from the following graph?



Your answer, 1, was incorrect. Your answer should have been 21.

SOLUTION
Recall that a **spanning tree** is a tree that connects all vertices of a graph. In this problem, to see the number of spanning trees visually may take some effort. However, there is a computational approach to doing this. Basically, you create the incidence matrix for this graph, then you create a new matrix from this as follows:

Each entry in the incidence matrix will be its opposite in the new matrix. For instance, when $A_{4,3}$ = 1, the corresponding value in the new matrix will be -1. Next, each diagonal value in the new matrix becomes the sum of the values either in the corresponding row or in the corresponding column for that entry (**from the original incidence matrix**). As an example, for position, (1,1), its value is the number of edges connected to vertex A, which is 2. Once completed, take the minor using the (2,2) entry and compute the determinant. In the event that the result is negative, take the absolute value of this number to get the required result, 21.

This is shown as follows below:

The incidence matrix for this graph is

	A	B	C	D	E
A	0	1	1	0	0
B	1	0	1	1	1
C	1	1	0	1	0
D	0	1	1	0	1
E	0	1	0	1	0

The corresponding matrix to be generated from this incidence matrix is

	A	B	C	D	E
A	2	-1	-1	0	0
B	-1	4	-1	-1	-1
C	-1	-1	3	-1	0
D	0	-1	-1	3	-1
E	0	-1	0	-1	2

Taking the determinant of the (2,2) minor gives

$$\begin{vmatrix} 2 & -1 & 0 & 0 \\ -1 & 3 & -1 & 0 \\ 0 & -1 & 3 & -1 \\ 0 & 0 & -1 & 2 \end{vmatrix} = 21.$$

Please note, however, that when performing this calculation, you may take any minor you prefer.

Related material

2 out of 4
It appears that you simply took the absolute value of a cofactor of the matrix. Please remember that you must first calculate the difference of the degree matrix (diagonal matrix of the degrees of the vertices) and the corresponding adjacency matrix, then you take the absolute value of any cofactor, i.e. calculate the determinant of any minor of the matrix.

Figure 4.1 Example of an RNI question, asking to determine the number of spanning trees in a given graph, along with feedback for responding and triggering a distracter.

The distracter used in Figure 4.1 involves taking the absolute value of the (3,3)-cofactor of the adjacency matrix. However, it should be obvious to any student that the correct answer will not be three as the number of edges present suggests there are many more spanning trees possible and therefore, a mark of two out of four may seem inappropriate. However, what is more important here is not the difference between a student's answer and the correct answer, but rather how the student may have answered the question in the first instance. Therefore, although the answer is obviously incorrect, the predicted procedure by which the student obtained this result shows that the student had some idea, at least, about how to solve the problem and so, two marks have been awarded based on this assumption.

In Section 3.12.2, a question on determining the chromatic number of a graph using a given procedure is presented. This question has three distracters, but two of them are minor. Similar to the questions on the shortest path problem, there is not much to work with in terms of researching distracters for this topic. However, based upon one of the theorems stated earlier, one reasonable distracter can be created.

The first two distracters simply take the correct answer and either add or subtract 1 from it so that two new values are generated. With this topic, it is quite possible to implement a procedure in the correct way, but then "tweak" the final result, as if there was something wrong with it; this idea is based on the material shown in Section 3.12.1.

The third distracter looks back at Theorem A.3. Recall that Brooks' Theorem gives an upper bound for the chromatic number of a graph. However, the procedure may give a different (or an even better) upper bound. As an example, the graph in Figure 3.40, according to Brooks' Theorem, has an upper bound of 9 (since vertex *H* has maximum degree, 9). However, as was shown in Figure 3.41, a new upper bound of 5 was obtained using the suggested procedure. Therefore, under the assumptions that Brooks' Theorem is taught to students learning material on vertex colouring and that the upper bounds will differ between Brooks' Theorem and the suggested procedure, this is a valuable distracter to have included in the question code.

In Section 3.12.3, the question asks for the colour to be associated with a particular vertex. Although there are no formal distracters for this problem, one “impromptu” distracter has been formed by selecting the colour that appears either immediately before it or immediately after it in the list of colours. Also, although an alert box will appear, reminding students to double-check their answers before submitting them, it may still be likely that a student will give the correct answer, but then spell it incorrectly. In such a case, nothing can be done for the student as (s)he will be told the answer is incorrect and so, no marks will be awarded (as it is not a distracter, either). However, as this material is primarily for undergraduate students in university, they should be reminded to carefully read all questions and to follow all instructions. At Brunel University, it has been noted for online, mathematics tests for other courses/modules that students often fail to read questions carefully and so, submit answers that are not formatted properly and thus, they, in turn, lose several marks, even though their answers are theoretically correct otherwise. For university students, if they cannot read a question fully and carefully enough, then strict penalties, such as the loss of all marks for that particular question, could be warranted.

4.6 Reverse Engineering

Another method for creating distracters is to design the answers and then create the question around the answers; this is known as **reverse engineering**. In Section 3.9.1, a MC question is presented on finding a spanning tree from a particular graph. In order to create the answers, the original graph needs to be created in such a way so that multiple features may appear within the graph. The distracters below highlight the features that can appear as a result of reverse engineering.

There are four distracters used in creating the MC questions on spanning trees. With these questions, though, distracters are mostly obvious, but with one exception. The first distracter is a spanning tree of the given graph, but with one edge included so that a cycle is formed. Upon looking at this candidate solution, students should easily find the cycle and eliminate it as a candidate. The second distracter removes an edge from a spanning tree, causing it to no longer be a spanning tree. The removal of this edge should create an unconnected graph and students should see this relatively easily. The third distracter implemented is an unconnected union of two subtrees of a graph. Although all vertices have connecting edges and although there are no cycles, this union is unconnected and thus, it cannot be a candidate solution to represent a spanning tree.

The fourth distracter is rather interesting in that it actually is a spanning tree, but due to the question wording, it does not constitute a spanning tree for the given graph and therefore, it is no longer a candidate solution. This candidate solution is not easy to catch as the other three solutions and this has the potential to catch many students off guard, especially with three obvious, incorrect solutions already in use. If the correct answer does not appear in a particular question, then it might be likely that a student will not select “None of these!”, but rather this distracter as it has all of the characteristics of a spanning tree.

In each distracter, a candidate spanning tree was introduced, but then it was “mutated” to destroy it as a candidate solution. In each case, though, there was an initial possibility of each answer being correct as most of the edges included matched up to the given graph. However, for these questions, the process behind how this works does not use a forward approach, but rather uses reverse engineering to allow the initial features of each distracter to combine together to create the graph that is seen as the “given graph” in each question. To do this, each of the four initial spanning trees, some before being mutated, along with the correct solution, were combined together to create something similar to a “layering effect”, where repeating edges were removed so that only one edge could join any pair of vertices and all other edges remained intact.

The concept of reverse engineering is very useful in generating randomised questions as the answers can be used to manipulate how the question is to be worded and how much information may be provided within the question. This concept appears again later in another topic, but in that instance, it will help to show how this concept may be used in other question types.

4.7 Distracters Created by Students

There are multiple perspectives from which to create distracters. However, distracters created from theories, strategies, etc. may not be the most trustworthy in catching out student errors. However, it is possible in some cases to review previous students’ answers in order to investigate errors and thus determine what distracters to create for a question. This strategy is ideal for determining distracters that are used more frequently, but it is trickier to obtain such information as ethics committee groups may block people from obtaining previous students’ work; in this case, it is ideal for anyone wanting to research common distracters to be directly involved with the modules in which assessments may be later created using Mathletics.

From Section 3.9.3, the number of spanning trees is to be calculated from a graph with a bridge connecting two disjoint subgraphs. This question is somewhat different in that additional strategies can be implemented beyond those already created from the previous question set.

The question in Section 3.9.3 has four distracters. The first distracter involves multiplying the number of spanning trees of one subgraph by the number of copies of that subgraph; in other words, the number of spanning trees for each copy is added together instead of multiplied. This distracter was generated based upon past examination results of students in the 2nd year course / module, MA2920: Algebra and Discrete Mathematics[43], at Brunel University, where it was seen that 17.74% of students (i.e. 47 out of 265 students) who answered the question on spanning trees from 2004 – 2008 willingly decided to multiply the number of spanning trees by the number of its copies rather than perform an exponential calculation of the number of spanning trees to the power of the number of its copies. In most of these cases, students performed such errors throughout the entire problem, including at the very end, when they added the numbers of different spanning trees together instead of multiplying them. A student who triggers this distracter will receive an overall score of $\frac{3}{6}$.

This question has been widely used in past examinations for MA2920, so past examination papers have also been analysed to assess student errors for this question; this can be shown in Appendix D. Students’ examination results show that 13 students skipped this question for all tests between 2004 – 2007, but then, 14 students skipped it on the 2007 – 2008 examination. Discounting these students, the percentage is now $\frac{47}{238} = 19.75\%$. Also, most of the other errors made by students who attempted this question had little commonality, aside from those who used the deletion-contraction recurrence[54]. However, the deletion-contraction recurrence is difficult to code in such a way as to catch a student making an error, so this was not considered in creating a distracter.

Another distracter for this question originally saw a student calculating the difference, *A* – *D*, instead of *D* – *A*; in other words, all of the values in the matrix are opposite of what they should be. For $n \times n$ matrices, where *n* is even, this has no effect. However, when *n* is odd, this causes the determinant to be negated. Nonetheless, it is expected that if a student then performs all other calculations correctly, then (s)he will omit the negative sign, knowing that the question is asking for a quantity of items and hence, must be positive. Doing this will give the student the correct answer and therefore, is not an official distracter for this question. However, if a student were to perform the same error as in the first distracter, then this could cause a major problem for the student. Considering the likelihood of the first distracter to occur, a student who commits this error may not notice it when (s)he then adds the numbers of spanning trees together. As such, this has been included as a second distracter. A student who triggers this distracter will receive a score of $\frac{2}{6}$.

The third distracter considers the possibility of students trying to rush to an immediate solution by simply taking the determinants of the adjacency matrices corresponding to the subgraphs, but then performs the necessary calculations using these values. Much of the work seen in the MA2920 examination scripts has students jotting down little pieces of information, but then they try to rush to an immediate solution. Although much of the work performed appeared to involve rather random procedures, if a student is, at least, somewhat aware of the overall procedure, then it is possible for the student to perform this error. For this distracter, a student who triggers it will receive a score of $\frac{1}{6}$.

The final distracter in this question has the student not only using the determinants of the adjacency matrices, but also adding the numbers of spanning trees. If this error does occur, then it will be apparent that the student has no clear indication of the procedure itself and so, no marks will be given for the attempt. However, at least some additional information will be provided to warn them of these errors, as is the case with all other distracters for all questions.

4.8 Additional Remarks about Distracters

Distracters can provide much insight into the commonality of errors made by students when answering questions. However, determining which distracters can occur can be problematic. Looking at errors students have made previously seems reasonable and easy to do, but there are ethical issues that can occur in doing this. Other strategies mentioned in this chapter provide insight on how to seek distracters, but this list is not necessarily exhaustive as other strategies may exist which also help in finding distracters.

In Appendix D, it is shown that some errors made by students have been categorised to form a more generalised basis for explaining the errors students are making in answering questions. However, when forming distracters, it is preferable not to generalise errors, but instead look carefully at the methodology to see where errors can occur in order to create suitable distracters. As an example of this, the question in Figure 3.40 has two calculation distracters where the answers are either one more or one less than the correct solution; although you can generalise the categorisation of the distracters, two separate distracters would warrant two separate sets of feedback.

Chapter 5 Structuring Graph Theory Questions in Computer-Aided Assessment using Mathletics

5.1 Mathletics On-Screen Features

In the Introduction, relevant features of Mathletics were discussed to explain the importance of the features that exist within the online assessment software. This section will look at how this works in practice with the questions designed for use in graph theory through a worked example, as shown in Figure 5.1.

Figure 5.1 shows a question on the topic of graph colouring. The question is presented in a default background colour, text size, colour, and font. This can be changed to suit a student's accessibility in viewing questions. Questions can also be printed in case students prefer to read it on paper whilst attempting the question online. There is only one question presented at a time on the screen in this example; this is the standard preference due to technical problems that infrequently occur with multiple questions appearing on the screen, especially when SVG graphics are included in questions and answers.

The question mentions "13 towns within Canada". The name, Canada, is chosen randomly from a template list of world countries. The inclusion of randomising text within questions helps in presenting individualised questions to students, which may help in deterring them from colluding during an assessment. The number, 13, is a random parameter designed for a suitable range of numbers of vertices. The corresponding graph uses SVG graphics using the same number of vertices as given by the random parameter and also selects edges to connect vertices using some randomisation, but also any preferences embedded into the coding by the question designer (e.g. a complete graph would need a subset of either K_5 or $K_{3,3}$ to be included as part of the graph and this feature can be embedded into the graph function if desired). The list of colours is also randomly chosen from a template list of colours.

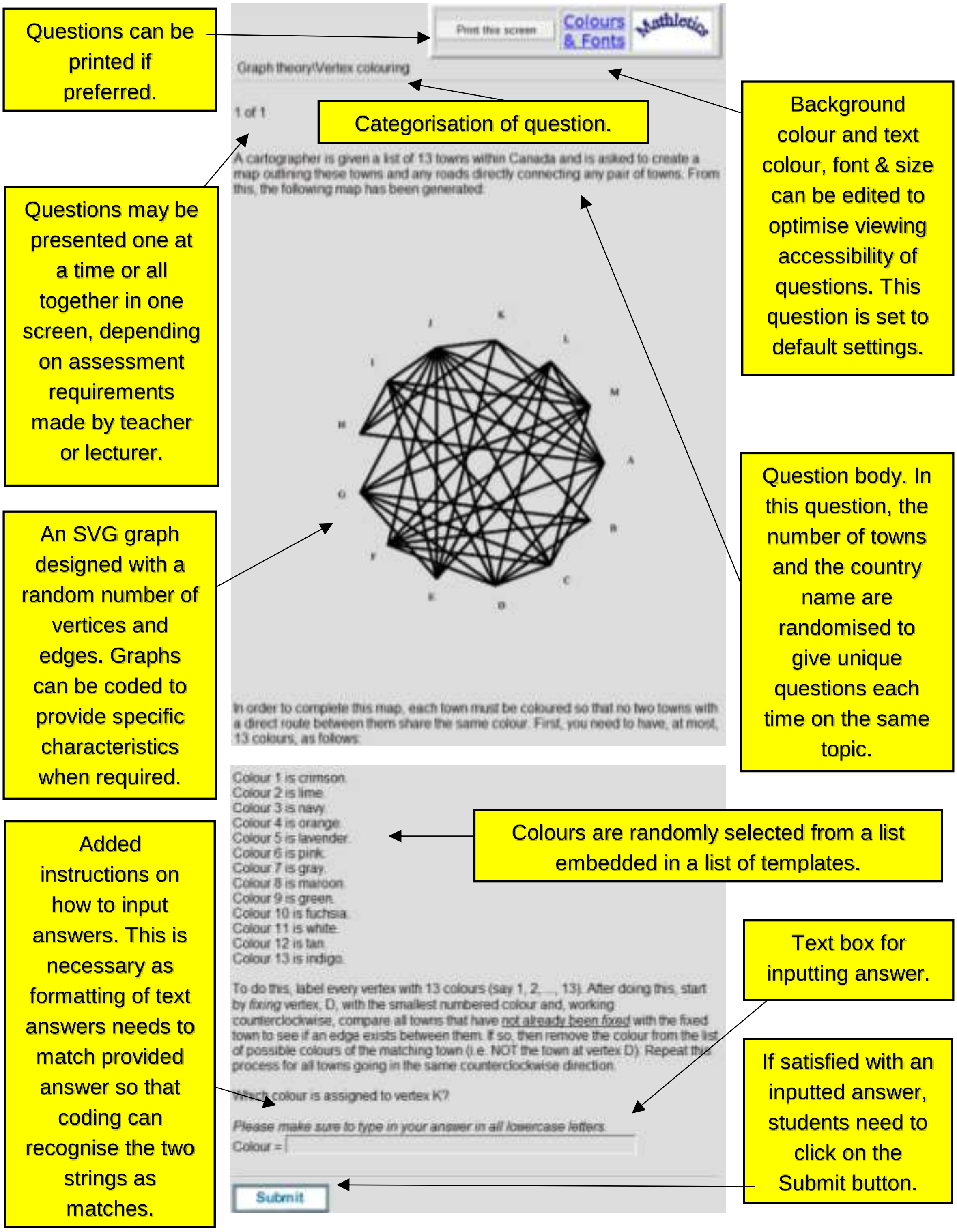


Figure 5.1 Example of a question designed in Mathletics on the topic of graph colouring within the subject of graph theory. Various properties of the question design are highlighted to detail features of the design of questions.

The topic of vertex colouring does not have specific algorithms which can easily be tested to determine the minimum number of colours needed to colour a random graph. However, strategies can be implemented to determine a possible minimum, so, in this question, a strategy is detailed. It is important that this strategy is well detailed in order to avoid any complications in understanding by the student. Also, the required answer is to be a colour, which is to be inputted by name, i.e. as text, in the text box provided. Inputted text is seen as a **string** by the code and strings are case-sensitive; for example, to choose the colour, pink, the texts, “pink” and “Pink”, would be seen as different strings and, unless otherwise coded within the question code, could result in an unnecessary loss of marks. Therefore, an added instruction is given after the input box to remind students to input their answers in all lowercase letters. Additionally, as shown in Figure 5.2, a pop-up window will appear to remind students to verify the formatting of their answers before clicking on submit a second time.

Figure 5.2 Upon submitting an answer to the question presented in Figure 5.1, a pop-up window appears, asking students to verify their answers are written in the correct format. Upon clicking OK, if students are satisfied with their answers and click Submit again, then the pop-up window will not re-appear; if a student changes the answer, then the pop-up window will re-appear.

Answer screens appear after submitting answers to questions. Answers provide detailed summative and formative feedback to questions and can include SVG graphics, as shown in Figure 5.3. Scores for an assessment are tabulated and will appear in a final screen after submitting all answers and receiving all feedback to questions.

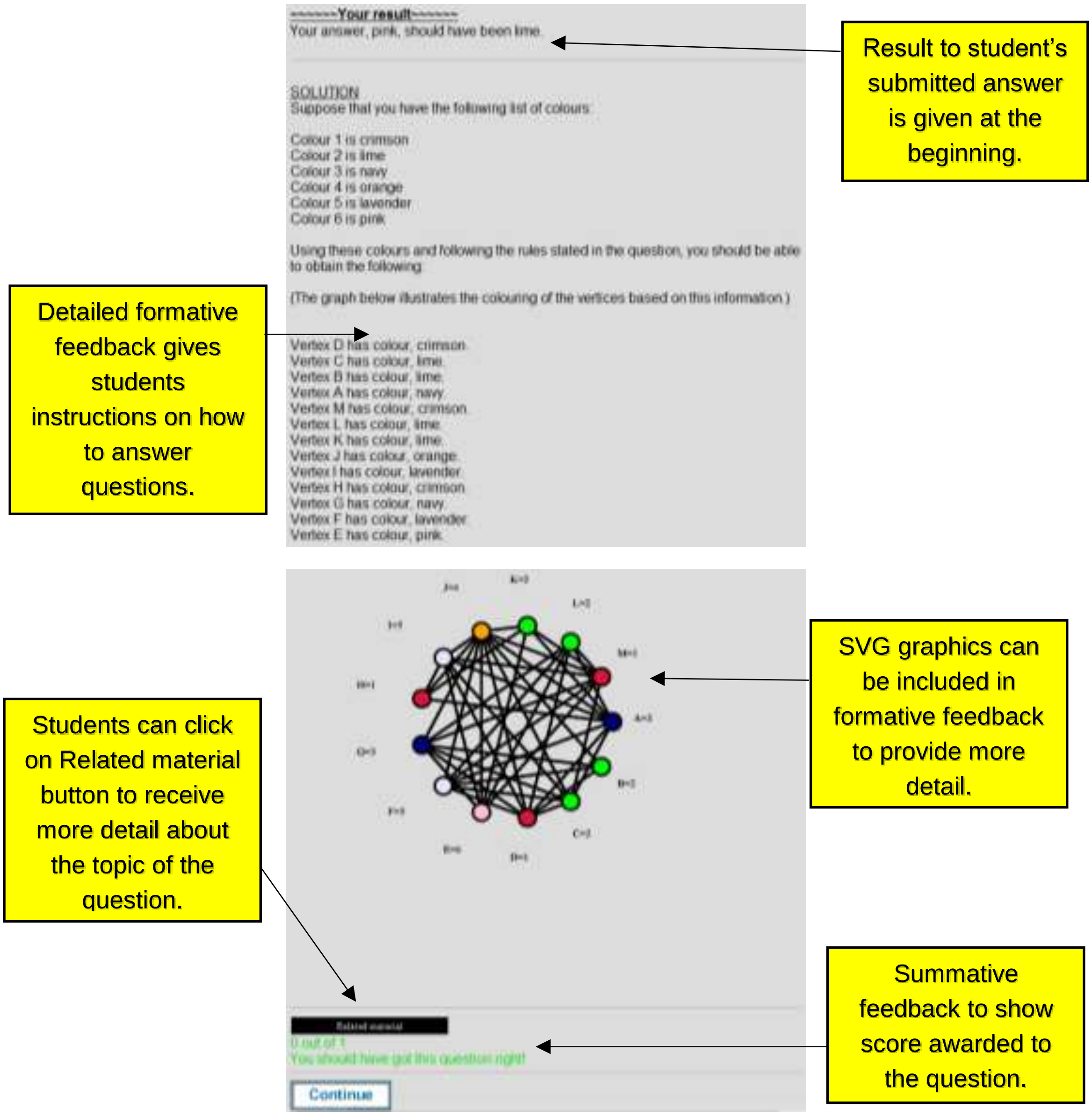


Figure 5.3 Feedback to answer given to question provided in Figure 5.1.

5.2 Generating Graphs for Use in Mathletics

There are many questions to ask when trying to generate a graph for use in Mathletics. In the case of graph theory:

1. How do you create a graph of n vertices such that they all appear regardless of the value of n?

2. How are the edges added and how is it decided which edges go where?
3. How do you create loops around a vertex and loops between two vertices?
4. How do you create arrows in the case of directed graphs?
5. If weights are needed, then should they be attached to the graphical image?

To answer the first question, vertices are arranged evenly around in a circular formation, using the mathematical concept of finding the roots of a complex number to place points evenly on a coordinate plane that remains invisible throughout.

Next, the edges have to be included. Doing this, however, requires one key component, namely the corresponding adjacency matrices. The SVG function that calls all such graphs require the corresponding adjacency matrix in order to determine which edges go where. In the case of an undirected graph, the adjacency matrix needs to be symmetric; otherwise, the graph will not appear properly. For a directed graph, however, the matrix does not need to be symmetric; in fact, educationally (or "pedagogically") speaking, it is preferable for the matrix to remain asymmetric as any symmetry may confuse students when answering questions. To do this involves editing the matrix within the question code itself and ensuring that if $a_{ij} = 1$, then $a_{ji} = 0$. However, if a graph is directed, then arrows are needed to show the directionality of the edges. This is done by attaching an arrow at a fixed ratio along a corresponding edge, going from the starting vertex and pointing towards the destination vertex. Each arrow is sized according to the overall size of its corresponding edge so that any unusual sizing issues do not arise.

Following from this, loops need to be created, whenever necessary. In the case that a loop is required around a vertex, then a circle is created so that its center is located just slightly further away from the center of the large circle than the corresponding vertex label. It is important that the distance from the vertex to the center of this circle is equivalent to the radius for that circle so that the circle will connect to its corresponding vertex. However, loops around a pair of vertices are more difficult as this is essentially two edges instead of one. Creating such a loop requires an ellipse rather than a circle, which is not symmetrical from all directions, and so, any ellipses that need to be included have to be rotated so that they may attach themselves to their corresponding vertices. The “skinniness” of any ellipse is also an issue as larger ellipses may interfere with the overall design of the graph. However, it is usually known that if a loop is required, then if the graph is directed, then the edges are automatically going in opposite directions. Nonetheless, though, since it is difficult to attach two arrows to one graphical element, it is usually best to either avoid the use of loops around a pair of vertices in digraphs or to include a “disclaimer” in each question, warning students of this.

One essential problem with creating these graphs is the labelling of edges whenever weights are included. All of the previously mentioned components to creating graphs are added in a “layering effect” so that one element is layered on top of another element. As each graph is randomly created, determining the location of labels for edges is incredibly difficult as any potential overlapping based on the layering effect can cause a question to become “unanswerable” due to unreadable information. Therefore, if weights are needed, then it is preferable to create a weighted matrix and attach it within a question and also, preferably adjacent to its corresponding graph.

With all of these properties, any graph for use in most graph theory topics can be generated. The template code for this function is shown in Code 2.4. Variations of this graph are explained in Chapter 2 and Chapter 3, detailing the technical and pedagogical issues that appear throughout.

5.3 Provision of Feedback

For all questions designed in Mathletics, it is possible to provide additional feedback after students answer questions to help them better understand what they have done correctly or incorrectly. Feedback can be simple or “generic”, as shown in Figure 5.3, but this can easily be enhanced, as shown in Figure 3.39 regarding chromatic polynomials and Figure 3.33 regarding Kruskal’s Algorithm, for instance.

Since questions are created with the intention of including random parameters throughout so that numerous realisations of the same question can appear, different strategies and formulae have to be programmed into the coding so that feedback is appropriate. Since feedback can be programmed into the question code directly, it can call upon the same functions used in the question to generate appropriate feedback.

Feedback can be used to explain how to answer a question correctly if a student answered it incorrectly. If a student answers a question correctly, then the methodology for solving the problem does not need to be included. Additionally, following the feedback, the question is scored and additional information, perhaps referring to the students' answers if preferred, is given; the use of distracters, as highlighted in Chapter 4, is very helpful here as partial credit and more detailed information about what the student may have done incorrectly in answering a question can all be provided.

5.4 Research Question: Versatile and Robust Questions in Graph Theory

This section answers the research question:

How can the potential of computer-aided assessment be exploited to set versatile and robust questions in graph theory?

As was shown in Chapter 4, Mathletics is very helpful in creating an organised library of questions within graph theory with additional tagging to allow teachers and lecturers to easily search for and design assessments based on their own requirements. The questions designed in Mathletics for the subject of graph theory have random parameters embedded within them so that numerous realisations of the same question can be generated. This creates individualised assessments for each student.

Graph theory relies heavily on the use of graphs and networks in order to illustrate problems. Graphs and networks needed to be drawn with a random number of vertices and edges, along with weights and coloured vertices, so that they can be flexibly used in multiple topics within graph theory. It was shown in Section 5.2 that a clear layout of the graph can be achieved by placing vertices equally round a circle. Sometimes, the centre of the graph could also be used as a vertex and so, another graph function was created to include this. The image needed to be embedded within an image frame on the question screen and this was carefully managed so that technical errors would not come up when loading a question on the screen.

A variety of topics within graph theory were visited in Chapter 3. Different question types were used to allow a wider range of questioning to be used. Questions used graphs or adjacency matrices to provide students with a better understanding of the relationship between the two items within graph theory and how they can easily be interchanged within topics. Different assessment schemes were implemented to ensure added flexibility in designing an assessment structure suitable to each assessment.

Chapter 6 Analysis of Graph Theory Questions using Mathletics

6.1 Methodology for 2007 – 2008 Analyses

From 2007 – 2008, graph theory questions designed using Mathletics[27] were tested on students enrolled in Brunel University's MA2920: *Algebra and Discrete Mathematics*[43] module. The testing of questions on students was designed in three parts:

- Practice test with all questions using graphs where available and not using adjacency matrices when possible.
- Practice test with all questions using adjacency matrices where available and not using graphs when possible.
- Invigilated test with all questions using a combination of graphs and adjacency matrices.

For the two practice tests, students were provided access to complete tests as often as they wished for a specified period prior to the invigilated test. The invigilated test was scheduled during a lecture session two months prior to the students sitting their final examination for the module. Students were in a controlled environment and were given a maximum of five attempts to complete the invigilated test. The maximum score achieved in the invigilated test would be the recorded score received and all scores counted toward their overall assessment scores for the module.

Questions were designed to have partial marking included so that students could receive partial credit if a predictable error (known as a **distracter**) was given in the students' answers. Students had to answer one question at a time, receiving a score and feedback after answering each question. Spare paper was provided to students in case they needed it to help them answer questions.

Results were summarised by the system software and formatted into spreadsheets, detailing the questions, topics in which they were categorised, number of attempts, facility, i.e. mean score between all respondents, and discrimination, i.e. correlation between each individual question score and the individual’s overall test score.

Two months after sitting the invigilated test session, students in MA2920 sat their final examinations for the module; this was a paper-based examination. As per university procedures, the lecturer assessed the examinations and the papers were then clerically checked, usually by Ph.D. students or other lecturers within the department. After these procedures were completed, the papers were individually analysed and were compared to previous final examination papers for the same module to determine if students who attempted the online questions progressed in their learning and understanding of the module better than those students who had not had access to the online questions at the time. Comparisons were made possible due to the fact that previous final examination papers for MA2920 were similar in structure and questioning to the final examination paper sat by the students who had been exposed to the online assessment material in graph theory.

The first analysis, conducted in 2008, involved students from the Brunel University mathematics module, MA2920. Students completed two sets of practice tests, namely a “visual test” using graphs and a “logical test” using adjacency matrices, prior to sitting an invigilated test, which combined graphs and adjacency matrices in each question. The analysis will determine if the designed questions have the potential to be effective in the assessment and learning of graph theory, but they will also highlight any patterns that may cause a significant change in overall assessment scores. An analysis of the students’ final examination results will also help to determine if they have performed better as a result of using the online software.

6.1.1 Statistical Analyses

Each online test is analysed independently, investigating the specific answers students gave for each question to determine the effectiveness of the pre-chosen distracters. A further analysis looks at discrimination values to determine the effectiveness of the question in relation to the overall assessment. Additional analyses will investigate comparable questions for significant differences in assessment scores based on various characteristics, such as question style (e.g. direct question or word problem), type (e.g. multiple-choice, numeric input, etc.), and presentation (e.g. use of graphs or adjacency matrices).

For the two practice tests, an additional analysis of correlations will explore the effectiveness of the assessments from one practice test to the next. A similar analysis will be reviewed for the invigilated test, but keeping in mind that the number of attempts at the invigilated test was limited to five attempts.

To analyse questions further, each set of identical questions for each topic were evaluated together using a two-factor ANOVA test without replication[55], which implies that there are no possible interactions between the independent factors of student outcomes and the question designs. This experiment design is used to determine if there are significant effects between elements in either factor and further analysis is performed when this is the case. However, note that much has already been discussed in relation to distracters using the quantitative analysis already performed in previous sections. This factor should have numerous differences throughout and this is verified using the two-factor model.

If differences occur between the question designs, then further testing will be needed to determine for which combinations of factors these differences appear. To do this, student t-testing[56] is performed. However, because the quantitative data collected is matched to the student outcomes, any t-testing to be performed must be a paired, two-sample test for means. The one-tailed and two-tailed effects for all paired comparisons are determined at the $\alpha = 0.05$ level of significance.

To test for significance using ANOVA, there are two methods that can be employed, but to describe these methods involves the following definitions[57]:

Def. 6.1 In a statistical model, the **F-ratio** is the ratio of the mean square value for a source to the mean square value for the calculated error.

Def. 6.2 In statistics, the **F-distribution** is a distribution of two independent, chi-squared random variables, say χ_u^2 and χ_v^2**,** with *u* and *v* degrees of freedom respectively. The ratio to determine a critical value for this distribution at a given level of significance, say α, is given by the equation,

$$F_{u,v} = \frac{\chi_u^2 / u}{\chi_v^2 / v}.$$

Def. 6.3 In statistics, the **P-value** is the probability that a given statistic can be used to determine the conclusion to a given experiment for any level of significance.

The methods are now explained:

First, set up a statistical experiment with a **null hypothesis**, H_0. Based off the null hypothesis, create an **alternative hypothesis**, H_1, that is mathematically opposing H_0. The analysis of variance (ANOVA) or a t-test can then be used. Upon determining all of the quantitative results, use the following[57] to determine the appropriate conclusion:

1. Determine the experimental value for the analysis and compare it with the critical value for the distribution at the $\alpha = 0.05$ level of significance. If the experimental value is greater than the critical value, then the null hypothesis is **rejected** and the *alternative hypothesis is therefore accepted*; otherwise, the null hypothesis is **not rejected**. Note that the statements, “The null hypothesis is not rejected” and “The null hypothesis is accepted” are not mathematically equivalent and therefore, it is not possible to say that “H_0 is accepted” by way of the experiment.
2. Determine the P-value. If it is lower than the set value for α, then the null hypothesis is **rejected**. If the P-value is higher than the set value for α, then the null hypothesis is **not rejected**.

For the tests that are performed on the quantitative results from the graph theory trials, the following hypotheses are formed:

$$H_0 : \mu_{o1} = \mu_{o2} = \ldots = \mu_{o8}$$

H_1: *At least one of the means of the outcomes differs from the others.*

$$H_0 : \mu_{q1} = \mu_{q2} = \mu_{q3} = \mu_{q4}$$

H_1: *At least one of the means of the question styles differs from the others.*

To test comparisons using t-testing[56], the critical values for the **one-tailed** and **two-tailed**, **paired, two-sample t-tests** are performed at the α level of significance using Microsoft Excel, along with the t statistic. The reason for choosing the paired, two-sample t-tests is because the data collected in each topic correspond to each distracter uniquely in comparison with other distracters and therefore, whenever comparing question styles, data must remain paired.

To perform the t-test, the mean, variance, and number of observations for each item in the comparison are required. Once determined, the experimental value for the distribution is calculated using the formula[58],

$$t_0 = \frac{\overline{d}}{S_d / \sqrt{n}},$$

where $\overline{d}$ is the sample mean of the differences between each pairing, S_d is the sample standard deviation of the differences, and *n* is the number of observations. From this, a comparison is then performed with an expected value using *n – 1* degrees of freedom and an appropriate degree of certainty (i.e. using α). However, this degree of certainty depends on the nature of the trial. If a one-tailed test is performed, then the hypotheses could be

$$\boxed{\begin{array}{l} H_0 : \mu_1 = \mu_2 \\ H_1 : \mu_1 < \mu_2 \end{array}} \text{ or } \boxed{\begin{array}{l} H_0 : \mu_1 = \mu_2 \\ H_1 : \mu_1 > \mu_2 \end{array}},$$

depending on the directionality expected. For either set of hypotheses, α remains as 0.05 and the tests are to determine if $t_0 < -t_{\alpha,n-1}$ and $t_0 > t_{\alpha,n-1}$ respectively; if so, then H_0 is rejected. However, if a two-tailed test is performed, then the hypotheses are

$$H_0 : \mu_1 = \mu_2$$
$$H_1 : \mu_1 \neq \mu_2,$$

where the value of α is cut in half to represent the equal possibilities of the distribution drifting to either side and the test is to determine if $|t_0| > t_{\alpha/2,n-1}$; if so, then H_0 is rejected.

Additionally, it should be noted that any t-testing performed has the risk of creating an experimental error, as explained in these definitions[57,59]:

Def. 6.4 In a statistical experiment, if the null hypothesis is rejected, but it is actually true, then it is said a **Type I error** has occurred. The probability of such an event occurring is denoted using the Greek letter, α, and can be referred to as the **significance level** for an experiment.

Def. 6.5 In statistics, the **familywise error rate** (FER) or **experimental error rate** (EER) is the probability of committing a Type I error in performing a set of experiments. If *n* experiments are performed, each with a **significance level**, α, then the probability of an error occurring is given by the equation, $P = 1 - (1-\alpha)^n$, where *n* is the number of comparisons performed.

The more comparisons that have to be made, the larger the EER can become. Because of this, it becomes more likely that one Type I error will be made, at least, when conducting t-tests. Therefore, when analysing the quantitative analysis of the questions and the answers selected, it is important to note the number of t-tests needed.

Questions in the analyses combine practice questions with the test questions to check for significant differences and what effects they could have on the promotion of learning in graph theory.

6.2 Methodology for 2008 – 2014 Analyses

From 2008 to 2014, introductory graph theory questions designed using Mathletics[27] were tested on students enrolled in Brunel University's MA0422: *Discrete & Decision Mathematics*[43] module. All sessions were for practicing purposes only, as arranged by the lecturer with all questions evaluated so that each student would score one mark if correct and zero marks if incorrect. Results were summarised by the system software and formatted into spreadsheets, detailing the questions, topics in which they were categorised, number of attempts, facility, i.e. mean score between all respondents, and discrimination, i.e. correlation between each individual question score and the individual's overall test score.

When someone attempts to answer a question using a multi-step process, it is possible to make a mistake at any step in the process, causing the submitted answer to be incorrect. It can be reasonably assumed that the more steps that exist in the method by which a question is answered, the less likely it is for the submitted answer to be correct. However, this may not be the only factor: Prerequisite knowledge may play a role, causing a question requiring only a few steps in solving to be more difficult than an easier question without the need for prerequisite knowledge requiring more steps in solving. It would be very helpful when designing online, objective questions to understand these factors better in order to produce more efficient questioning with appropriately detailed feedback; this can help students to learn from their attempts in a practice-based environment in order to better understand the learning material in preparation for invigilated assessments. Therefore, the objective is to determine what, if anything, caused the facility values for some questions to be higher than others and to implement an approach by which we can better design online, objective questions in the future.

6.2.1 Statistical Analyses

During the investigation of the data, it was revealed that there was a significant variation in questions being analysed between the 2008 – 2011 academic years and the 2011 – 2014 academic years:

- 12 questions on Prim's and Kruskal's algorithms were presented to students from 2011 – 2014.
- The topic of edge sets removed questions involving digraphs from 2011 – 2014.
- Multiple questions were prepared on the topic of degree from 2011 – 2014, using previously designed questions on degree, indegree, and outdegree, but isolating possible realisations to ensure that different realisations would be tested each time.
- There were no questions on simple and connected graphs from 2011 – 2014.
- There were no questions on bipartite graphs from 2011 – 2014.

To ensure a fair analysis of the data, two groupings were formed with two separate analyses conducted. Any questions that appeared in both academic year groups were extracted so that results from one academic year group would not affect analysis in the other academic year group. Also, it was noted that the question topic changed for some questions; because this did not have any effect on the questions presented and because of the significance of the academic year groupings, the topic category was ignored in the analysis. The focus for the analysis was on the facility and discrimination of questions to identify which questions were more challenging or discriminating. The following chapters seek to explain the results on the basis of understanding why some questions were more challenging to answer as opposed to other questions.

Each question could be identified by multiple characteristics, i.e. topic, question type, and use of graphs compared to use of adjacency matrices; however, the objective is to determine if facility and discrimination are affected by the number of reasonable steps required to solve a problem. Therefore, it was important to compare all questions as a collective unit. Also, summary data for all questions from each academic year group is provided for facility to show overall mean scores for each academic year.

Since all questions were marked with either zero or one, standard deviation of scores would only be representative of the proportion of correct answers given, which can be easily noted from the facility value. Therefore, standard deviation is not considered in the analysis of these questions. Discrimination values are presented for each academic year in which questions were made available and attempted online. Also, these questions were conducted in “practice conditions”, implying students were not obliged to complete all questions in the assessment, nor did the assessments necessarily have to go through all questions in each sitting. Therefore, overall discrimination values do not appear in the analysis; however, each academic year’s question discrimination values do appear as they were made available through the statistical analysis provided by the software after each academic year’s testing.

Since all questions were evaluated with scores of either 0 or 1, statistical tests were conducted to test the differences between proportions using a normal distribution. Let μ_i^α be the proportion of correct answers for question set i. One-tailed tests were set with a significance level of $\alpha = 0.05$ ($Z_{crit} = \pm 1.645$). Any values, $Z_{test} \ni Z_{test} > |Z_{crit}|$, would result in a rejection of the null hypothesis, H_0, and an acceptance of the alternative hypothesis, H_1, depending on the value of Z_{test}. The null and alternative hypotheses for these statistical tests are given as follows:

$$\boxed{\begin{array}{l} H_0: \mu_1^\alpha = \mu_2^\alpha \\ H_1: \mu_1^\alpha < \mu_2^\alpha \end{array}} \quad \text{or} \quad \boxed{\begin{array}{l} H_0: \mu_1^\alpha = \mu_2^\alpha \\ H_1: \mu_1^\alpha > \mu_2^\alpha \end{array}}$$

(if $Z_{test} < -1.645$) (if $Z_{test} > 1.645$)

To determine what makes one question have a higher facility over another question, we need to first group similar questions together so that a valuable comparison can be made between them. When viewing these groups, we need to look at the specific characteristics that make them different, e.g. question type, question topic, comparable features in similar questions. From viewing these characteristics, we can have an understanding as to the number of steps that should be required to answer the question correctly and, furthermore, they will be useful in identifying any relevant differences in student understanding. If, however, it is not clear what characteristics made a question seemingly more challenging than another question, then it can be concluded that there is no clear evidence to conclude how this occurred. This, however, will all occur generally from two perspectives: the year-by-year analysis will look at any significant patterns between academic years with different cohorts of students and the overall facility values will look at a general analysis of the results for all three academic years combined, as if all students were part of the same cohort.

Chapter 7 Analysis of Advanced Graph Theory Questions using Mathletics

7.1 Hypotheses

In the 2007 – 2008 assessments, students were allowed to re-attempt questions as often as they wished in a practice environment with two test sets having been designed for questions involving adjacency matrices and questions involving graphs; later, in the invigilated assessment, students were given up to three attempts to answer questions from a test set that included questions involving both graphs and adjacency matrices. Since students repeatedly attempted similar tests, it is hoped that they will learn from any previous mistakes made in order to improve in future attempts; therefore, it is being hypothesized that the correlation matrix will show that the test-retest coefficients representing correlations from one test to the next test will be positive.

For these assessments, it was preferred for questions to have a facility of 0.5 and a discrimination of 1; however, this cannot be expected throughout, especially in a finite number of attempts. Where questions have lower facility values, positive discriminations will still be encouraging as this will indicate these questions had some academic value to overall assessments; alternatively, questions with high facility values and low discrimination values may not be so worthwhile in overall assessments. Therefore, it is being hypothesized for each question analysed that facilities will be ideally close to a value of 0.5, but regardless of the facility, discrimination values will remain significantly positive.

When analysing answers students gave to questions, it is ideal for students to give correct answers and therefore, the number of times a distracter is chosen ought to remain small. However, it is still expected that in attempts, students will eventually select an incorrect answer. Additionally, as explained earlier, some consideration towards the creation of distracters has been made so that they are not obviously incorrect. Therefore, it is being hypothesized that the proportion of distracters chosen remains small, yet is still significant enough to have been selected by students during their attempts.

As noted earlier, in practice sessions, students were given two test sets to review, one involving questions using graphs and the other involving questions using adjacency matrices. With a subject that includes visual and numerical elements separately, it may be possible that students performed better in one set of tests versus the other. Additionally, since the invigilated test involved questions using both adjacency matrices and graphs, students who practised the earlier tests should have been able to work suitably well with both question styles. However, it may not be expected that one question style dominates the other as different topics may provide more advantageous opportunities to use one of the two question styles. Therefore, it is being hypothesized that there will be significant differences between question styles in comparison, but knowing that this will vary from one topic to the next.

7.2 Quantitative Analysis of Results Before the Invigilated Test Session for Visual Components

The quantitative results determined in this section look at the series of questions that focused on the visual components of the topics in graph theory. Adjacency matrices were not used in any of these questions, but rather just graphs. The data retrieved reflects upon the students' performance and abilities to handle this material.

7.2.1 Discrimination and Other Quantitative Results

There were eight questions given in the visual question set for graph theory. Details of each question and their respective, quantitative results are given in Table 7.1.

Six of the eight questions were multiple-choice (MC) and two were responsive, numeric input (RNI).

Question description	Question Type	Times answered	Max. score	Mean score	Standard deviation of score	Facility	Correlation
Find the simple and connected graph	MC	320	1	0.787	0.41	0.787	0.343
Bipartite graph search	MC	301	3	1.987	1.337	0.662	0.436
Find the non-Hamiltonian cycle	MC	285	3	1.253	1.482	0.418	0.437
Find the Eulerian cycle	MC	269	5	3.208	2.35	0.642	0.623
Find the planar graph	MC	255	5	1.686	2.369	0.337	0.555
Number of spanning trees (with branches)	RNI	80	6	0.388	1.326	0.065	0.437
Number of spanning trees using graphs	RNI	68	4	0.75	1.53	0.188	0.455
Find the correct spanning tree	MC	69	3	1.275	1.293	0.425	0.37

Table 7.1 Table of quantitative results from practice questions looking only at graphs.

7.2.2 Finding the Simple and Connected Graph

The question asking to find the simple and connected graph among a list of candidate solutions generated the results shown in Table 7.2.

320 attempts were performed on this question. The facility for this question was 0.787, implying that many students found this question to be somewhat easy. The index of discrimination, noted as *correlation* in Table 7.1 and given as 0.343, shows that this question somewhat helped to measure the same skills as the test overall. 78.75% of the students who attempted this question answered it correctly, further suggesting the simplicity of this question. The diminished numbers of students who triggered the given distracters additionally illustrates the simplicity of this question.

Outcome name	Times answered	Percentage of times answered
Correct	252	78.75%
Loops around one vertex.	11	3.44%
Loops around a pair of vertices.	0	
Unconnected.	14	4.38%
Loops (almost) everywhere.	9	2.81%
None Of These	23	7.19%
Did Not Know	11	3.44%
Not Answered	0	

Table 7.2 Table of responses given by students for the MC question asking to find the simple and connected graph.

7.2.3 Finding the Bipartite Graph

The question asking to find the bipartite graph among a list of candidate solutions generated the results shown in Table 7.3.

Outcome name	Times answered	Percentage of times answered
Correct	179	59.47%
Not complete	26	8.64%
Not bipartite	42	13.95%
Complete graph	9	2.99%
Wheel graphs	23	7.64%
None Of These	14	4.65%
Did Not Know	8	2.66%
Not Answered	0	

Table 7.3 Table of responses given by students for the MC question asking to find the bipartite graph.

301 attempts were performed on this question. The facility for this question was 0.662, implying that many students found this question to be somewhat easy. The index of discrimination, given as 0.436, shows that this question somewhat helped to measure the same skills as the test overall. 59.47% of the students who attempted this question answered it correctly, further suggesting the likelihood of simplicity for this question. However, 13.95% of the respondents triggered the “not bipartite” distracter and 8.64% triggered the “not complete” distracter when attempting this question. The question is given so that although the vertices appear in a cyclic formation, the two disjoint subsets that make the bipartite graph are formed by a random selection of vertices to appear in each set. Therefore, it is likely that students may have been deceived when deciding to choose either of these distracters.

7.2.4 Finding the non-Hamiltonian Graph

The question asking to find a graph that was non-Hamiltonian among a list of candidate solutions generated the results shown in Table 7.4.

285 attempts were performed on this question. The facility for this question was 0.418, implying that some students, although not a majority, found this question to be easy. The index of discrimination, given as 0.437, shows that this

question somewhat helped to measure the same skills as the test overall. It is interesting to note, though, that 19.30% of the respondents triggered the distracter regarding the degree of every vertex. This distracter is based upon the theorem[60] that any loop-free graph with more than three vertices and with $\deg x \geq \frac{n}{2}, \forall x \in V$ must have a Hamiltonian cycle. This distracter warranted a partial credit of one mark for the question.

Outcome name	Times answered	Percentage of times answered
Correct	119	41.75%
Hamiltonian and Eulerian cycles	23	8.07%
Loop-free graph with $\lvert V \rvert \geq 3$, where $\deg(x) \geq \frac{n}{2}$, $\forall x \in V$	55	19.30%
Loop-free, undirected graph with $\lvert V \rvert \geq 3$ and $\lvert E \rvert \geq \binom{n-1}{2} + 2$	26	9.12%
Wheel graph	27	9.47%
None Of These	22	7.72%
Did Not Know	13	4.56%
Not Answered	0	

Table 7.4 Table of responses given by students for the MC question asking to find the graph that is not Hamiltonian.

7.2.5 Finding the Eulerian Graph

The question asking to find a graph that was Eulerian among a list of candidate solutions generated the results shown in Table 7.5.

Outcome name	Times answered	Percentage of times answered
Correct	166	61.71%
Almost a cycle, but the two ends are not connected	11	4.09%
Two cycles connected by one edge.	22	8.18%
Edge added to an already Eulerian cycle.	42	15.61%
Two unconnected subgraphs.	5	1.86%
None Of These	19	7.06%
Did Not Know	4	1.49%
Not Answered	0	

Table 7.5 Table of students' for the MC question asking to find the Eulerian graph.

269 attempts were performed on this question. The facility was 0.642, implying that a good number of students found this question to be easy. The index of discrimination, given as 0.623, shows that this question helped well to measure the same skills as the test overall. Additionally, it can be argued that this question is a somewhat good indicator for determining how well students performed overall in their attempts. It is interesting to note, though, that 15.61% of the respondents triggered the distracter regarding an edge being connected to what would otherwise have been an Eulerian cycle. This distracter was an unlikely candidate for being such a good distracter as students only needed to determine if the graph was connected and if the degree of each vertex was even; the addition of one edge would easily cause two of the vertices to have an odd degree and so, this distracter should easily be caught by students. Nonetheless, 42 students were caught triggering it and so, it is worth further consideration as a distracter.

7.2.6 Finding the Planar Graph

The question asking to find a graph that was planar among a list of candidate solutions generated the results shown in Table 7.6.

Outcome name	Times answered	Percentage of times answered
Correct	86	33.73%
K_5 subgraph	24	9.41%
$K_{3,3}$ subgraph	38	14.90%
Both subgraphs included	31	12.16%
Complete graph minus one edge	23	9.02%
None Of These	28	10.98%
Did Not Know	25	9.80%
Not Answered	0	

Table 7.6 Table of responses given by students for the MC question asking to find the planar graph.

255 attempts were performed on this question. The facility for this question was 0.337, implying that a smaller portion of students, compared to the previous question, found this question to be easy. Again, notice that the percentage of students who answered this question correctly is equivalent to the facility, but this could simply be coincidental to the spread of the incorrect answers chosen. The

index of discrimination, given as 0.555, shows that this question helped well to measure the same skills as the test overall. It is interesting to note, though, that all of the distracters, along with the response, “Did Not Know”, were each triggered by relatively equivalent percentages of students. It is actually surprising that 23 students out of 255 actually triggered the distracter of a complete graph, minus one edge. This question always selects a minimum of eight edges, so any such graph would obviously have the K_5 and $K_{3,3}$ subgraphs. However, it is welcoming to see that a fair number of students who did not know the correct answer actually admitted it by selecting “Did Not Know”.

7.2.7 Spanning Trees

7.2.7.1 Determining the Number of Spanning Trees when Branches are Given

The first RNI question in the set, which asks to determine the number of spanning trees in a graph given a set of subgraphs as branches, generated the results shown in Table 7.7.

Outcome name	Times answered	Percentage of times answered
Correct	4	4.60%
Wrong	0	
Multiplied numbers of spanning trees in subgraphs by their corresponding numbers of copies.	0	
Initial matrix calculation was opposite of what it should have been.	0	
Calculated determinants of adjacency matrices for each subgraph.	7	8.05%
Calculated determinants of adjacency matrices for each subgraph, then multiplied numbers of spanning trees in subgraphs by their corresponding numbers of copies.	0	
Not Answered	76	87.36%

Table 7.7 Table of responses given by students for the RNI question asking to determine the number of spanning trees in a graph given a set of subgraphs, along with their respective numbers of copies within the graph.

It is important to note that only 80 attempts were performed on this question when over 250 attempts were made on the previous questions. The reason for this is because the setup for each attempt involved selecting one of three spanning trees questions to be selected for the student to trial; the other two questions appear in the next subsections. The facility for this question was

0.065, implying that virtually nobody answered this question correctly, which can obviously be noted in Table 7.7, where it notes that 87.36% of students did not answer the question at all. Clearly, students did not feel comfortable approaching this question. However, it should also be noted that this question was created based upon the fact that a similar question has appeared in the final examinations for MA2920 in the past four years. Therefore, students should have made more valuable efforts in attempting this question.

The index of discrimination, given as 0.437, shows that this question helped to measure the same skills as the test overall. Although 87.36% of the students who saw this question did not attempt it, 63.64% of those who did attempt it triggered the distracter that is created by multiplying the number of copies of a subgraph by the number of spanning trees in it. Based on simple combinatorics, the correct answer involves calculating the number of spanning trees of a subgraph *to the exponent* of the number of copies of that subgraph and then multiplying each of the results together. This distracter was created based upon the viewing of exam results for MA2920 in previous years; more details about the previous exam results are given later in this unit.

7.2.7.2 Determining the Number of Spanning Trees in a Graph

The second RNI question in the set, which asks to determine the number of spanning trees in a graph, generated the results shown in Table 7.8.

Outcome name	Times answered	Percentage of times answered
Correct	12	17.14%
Wrong	0	
All values in the matrix were kept positive before continuing with the calculations.	0	
Calculated the determinant of the difference of the degree matrix and the adjacency matrix.	3	4.29%
Calculated the absolute value of a cofactor of the adjacency matrix.	3	4.29%
Not Answered	52	74.29%

Table 7.8 Table of responses given by students for the RNI question asking to determine the number of spanning trees in a given graph.

It is important to note that only 68 attempts were performed on this question when over 250 attempts were made on the previous questions. Again, the reason for this is because the setup for each attempt involved selecting one of three spanning trees questions to be selected for the student to trial. The facility for this question was 0.188, implying that not many people answered this question correctly, which can obviously be noted in Table 7.8, where it notes that 74.29% of students did not answer the question at all. Although this question is not directly based upon a question from previous exams, the material is similar to what does appear and so, students should have made a more valuable effort in attempting this question.

The index of discrimination, given as 0.455, shows that this question helped well to measure the same skills as the test overall. Although 74.29% of the students who saw this question did not attempt it, 20% of those who did attempt it triggered the distracter that is created by calculating the determinant of the difference of the degree matrix and the adjacency matrix; this result is always zero[61] and so, even though the graph appeared on the screen and even though it should have been obvious that there would be multiple spanning trees, students still gave zero as their answer. An equal number of students chose the distracter created by calculating the absolute value of a cofactor of the adjacency matrix. When students attempted this question, the only way they could trigger this distracter would have been to take the (1,1)-cofactor of the adjacency matrix. However, since then, the code behind the question has allowed for other diagonal cofactors to be selected; this is because these values will usually be larger than other values in the matrix involving the difference of the degree and adjacency matrices.

7.2.7.3 Finding a Proper Spanning Tree for a Particular Graph

The last question in the set, a MC question that asks to find a proper spanning tree for a given graph, generated the results shown in Table 7.9.

69 attempts were performed on this question. The facility for this question was 0.425, implying that a good number of students answered this question

correctly. The index of discrimination, given as 0.37, shows that this question helped somewhat well to measure the same skills as the test overall.

It is interesting to note the distracters here, as well. 14.08% of the respondents chose the distracter that has a cycle embedded within it. Obviously, if the question is asking to find a spanning tree, then surely this answer would not be selected by many students, but nonetheless, this has occurred. The distracter, "Disconnected subtrees", refers to a disconnected subgraph that is formed by a random selection of edges that appear in the graph, but do not create a connected spanning tree. Connectedness is important for obtaining a spanning tree and so, it is worth noting that nearly 10% of the respondents used this distracter. Most importantly, though, is the distracter of a spanning tree that does not appear in the given graph. This distracter was created to see if students are paying enough attention when attempting these questions. Unfortunately, 26.76% of those who responded to this question triggered this distracter and so, it can be noted that a fair number of students do not pay close enough attention to the question and to the possible answers given. However, this distracter is clearly helpful for catching students doing similar things in future attempts.

Outcome name	Times answered	Percentage of times answered
Correct	23	32.39%
Cycle created.	10	14.08%
Not connected.	4	5.63%
It is a tree, but it does not correspond to the graph.	19	26.76%
Disconnected subtrees.	7	9.86%
None Of These	0	
Did Not Know	6	8.45%
Not Answered	2	2.82%

Table 7.9 Table of responses given by students for the MC question asking to find a spanning tree for a given graph.

7.3 Quantitative Analysis of Results Before the Invigilated Test Session for Logical and Mathematical Components

The quantitative results determined in this section look at the series of questions that focused on the logical / mathematical components of the topics in graph theory. Graphs were not used as the primary focus for any of these questions, but rather adjacency matrices. The data retrieved reflects upon the students' performance and abilities to handle this material.

7.3.1 Discrimination and Other Quantitative Results

Question description	Question Type	Times answered	Maximum score	Mean score	Standard deviation of score	Facility	Correlation
Find the simple connected graph given the adjacency matrices	MC	328	1	0.677	0.468	0.677	0.417
Bipartite adjacency matrix search	MC	151	1	0.523	0.501	0.523	0.475
Number of vertices in a partition of a bipartite graph	NI	155	1	0.587	0.494	0.587	0.431
Find the non-Hamiltonian cycles using adjacency matrices	MC	287	3	0.794	1.326	0.265	0.547
Find the correct Eulerian cycle in an adjacency matrix	MC	293	5	2.672	2.41	0.534	0.686
Find the planar adjacency matrix	MC	126	6	2.19	2.685	0.365	0.709
Which combination of properties does not yield a planar graph	MC	139	2	2	0	1	-1
Number of spanning trees (with branches) using adjacency matrices	RNI	76	6	0.276	1.04	0.046	0.286
Number of spanning trees using adjacency matrices	RNI	78	4	1.269	1.8	0.317	0.592
Spanning trees using adjacency matrices	MC	76	3	0.816	1.262	0.272	0.42

Table 7.10 Table of quantitative results from practice questions looking only at adjacency matrices.

Eight questions were given in the visual question set for graph theory. Details of each question and their respective, quantitative results are given in Table 7.10.

In the logical / mathematical question set, there are ten questions: seven are MC questions, one is a NI question, and two are RNI questions. Two additional questions in this question set include a NI question asking to find the number of vertices in either the larger or the smaller partition of a bipartite graph, as well as a MC question asking to determine which paired quantities of edges and vertices do not yield a planar graph.

7.3.2 Finding the Simple and Connected Adjacency Matrix

The question asked to find the simple and connected adjacency matrix among a list of candidate solutions and generated the results shown in Table 7.11.

328 attempts were performed on this question. The facility for this question was 0.677, implying that many students found this question to be somewhat easy. The index of discrimination, given as 0.417, shows that this question somewhat helped to measure the same skills as the test overall. $\frac{2}{3}$ of the students who attempted this question answered it correctly, further suggesting the simplicity of this question.

Outcome name	Times answered	Percentage of times answered
Correct	222	66.67%
Loops around one vertex.	29	8.71%
Loops around a pair of vertices.	0	
Unconnected.	17	5.11%
Loops (almost) everywhere.	9	2.70%
None Of These	25	7.51%
Did Not Know	26	7.81%
Not Answered	5	1.50%

Table 7.11 Table of responses given by students for the MC question asking to find the adjacency matrix corresponding to a simple and connected graph.

7.3.3 Bipartite Graphs

7.3.3.1 Finding the Bipartite Adjacency Matrix

The question was one of two looking at bipartite graphs. The MC question asked to find the adjacency matrix corresponding to a bipartite graph among a list of candidate solutions and the generated results are shown in Table 7.12.

151 attempts were performed on this question. The facility for this question was 0.523, implying that this was a relatively fair question for students to answer as it is close to the optimal value of 0.5. The index of discrimination, given as 0.475, shows that this question somewhat helped to measure the same skills as the test overall. Just over half of the students who attempted this question answered it correctly, further suggesting the simplicity of this question.

Outcome name	Times answered	Percentage of times answered
Correct	79	50.97%
Not complete	10	6.45%
Not bipartite.	6	3.87%
Complete graph	11	7.10%
Wheel graphs.	15	9.68%
None Of These	30	19.35%
Did Not Know	0	
Not Answered	4	2.58%

Table 7.12 Table of responses given by students for the MC question asking to find an adjacency matrix corresponding to a bipartite graph.

The chosen distracters are triggered relatively well, but the option, "None of These", was more popular with 19.35% of students incorrectly selecting this as the answer. This suggests that the correct answer is well hidden among the choices, which is very helpful since this question would otherwise be even easier for students to answer.

7.3.3.2 The Number of Vertices in a Partition of a Bipartite Adjacency Matrix

The NI question looking at bipartite graphs asked to determine the number of vertices in a specific partition of the adjacency matrix corresponding to a bipartite graph. The results generated can be seen in Table 7.13.

Outcome name	Times answered	Percentage of times answered
Correct	91	57.96%
Wrong	64	40.76%
Not Answered	2	1.27%

Table 7.13 Table of responses given for NI question asking to find the number of vertices in a partition of an adjacency matrix corresponding to a bipartite graph.

155 attempts were performed on this question. The facility for this question was 0.587, implying that this was a relatively fair question for students to answer. The index of discrimination, given as 0.431, shows that this question somewhat helped to measure the same skills as the test overall. About 58% of students who attempted this question answered it correctly, further suggesting the simplicity of this question. This question was only worth one mark and feedback provided did not require responsive input.

In total, 306 attempts were made at questions involving adjacency matrices corresponding to bipartite graphs. The facilities suggested the questions were fair to students, although not easy. The indices of discrimination suggested in each case that the questions somewhat helped to measure the same skills as the test overall. Both questions are seen as being valuable to this question set.

7.3.4 Finding the non-Hamiltonian Adjacency Matrix

This question asked to find the adjacency matrix that did not correspond to a Hamiltonian graph. The results generated can be seen in Table 7.14.

287 attempts were performed on this question. The facility for this question was 0.265, implying that this question was somewhat difficult for students to answer. However, the index of discrimination, given as 0.547, shows that this question helped well to measure the same skills as the test overall.

Outcome name	**Times answered**	**Percentage of times answered**
Correct	76	26.30%
Hamiltonian and Eulerian cycles	41	14.19%
Loop-free graph with $\lvert V\rvert \geq 3$, where $\deg(x) \geq \frac{n}{2}$, $\forall x \in V$	53	18.34%
Loop-free, undirected graph with $\lvert V\rvert \geq 3$ and $\lvert E\rvert \geq \binom{n-1}{2} + 2$	33	11.42%
Wheel graph	17	5.88%
None Of These	22	7.61%
Did Not Know	45	15.57%
Not Answered	2	0.69%

Table 7.14 Table of responses given by students for the NI question asking to find the adjacency matrix that does not correspond to a Hamiltonian graph.

Each of the chosen distracters was significant in catching students making errors. The distracter, “Hamiltonian and Eulerian cycles”, was surprisingly good as it was triggered 14.19% of the time. However, this distracter should have been obviously seen as being Hamiltonian as it is essentially a cycle graph. It is possible that students may have misread the question, thinking they were expected to find a Hamiltonian graph, but it is equally possible that students may have rushed through this question only to see the feedback. The distracter, “Loop-free graph with $|V| \geq 3$, where $\deg x \geq \frac{n}{2}, \forall x \in V$”, was triggered 18.34% of the time, suggesting that it is a valuable distracter for this question. The response, “Did Not Know”, was triggered 15.57% of the time, which is surprising, but very good to see as it shows students are willing to admit they are not confident about selecting any particular answer, including “None of These”.

7.3.5 Finding the Eulerian Adjacency Matrix

This question asked to find the adjacency matrix that corresponded to an Eulerian graph. The results generated can be seen in Table 7.15.

293 attempts were performed on this question. The facility for this question was 0.534, implying that this was a fair question for students to answer. The index of discrimination, given as 0.686, shows that this question helped very well to measure the same skills as the test overall.

Outcome name	Times answered	Percentage of times answered
Correct	144	48.32%
Almost a cycle, but the two ends are not connected	21	7.05%
Two cycles connected by one edge.	20	6.71%
Edge added to an already Eulerian cycle.	20	6.71%
Two unconnected subgraphs.	16	5.37%
None Of These	49	16.44%
Did Not Know	23	7.72%
Not Answered	5	1.68%

Table 7.15 Table of responses given by students for the question asking to find the adjacency matrix corresponding to an Eulerian graph.

Unlike the visual question, the distracter, “Edge added to an already Eulerian cycle” was not triggered as often. Due to the logical / mathematical nature of this problem, students may have been able to better detect that two vertices each had odd degree, causing the corresponding graph to be non-Eulerian. However, the percentage of students who triggered the distracter, “Two unconnected subgraphs”, nearly tripled, which could suggest they were not paying enough attention to the nature of the corresponding graph, but rather just the vertex degrees in each adjacency matrix.

Again, it is surprising to see such a large percentage of students selecting “Did Not Know” as although this did happen with some students, the percentages for selecting this within the logical / mathematical question set have increased.

7.3.6 Planar Graphs

7.3.6.1 Finding an Adjacency Matrix Corresponding to a Planar Graph

There were two MC questions that looked at planar graphs. The first question asked to find the adjacency matrix that corresponded to a planar graph. The results generated can be seen in Table 7.16.

126 attempts were performed on this question. The facility for this question was 0.365, implying that this was somewhat difficult for students to answer, yet not extremely difficult. The index of discrimination is 0.709, which shows that this question helped tremendously to measure the same skills as the test overall.

Outcome name	Times answered	Percentage of times answered
Correct	41	31.78%
K_5 subgraph	9	6.98%
$K_{3,3}$ subgraph	14	10.85%
Both subgraphs included	9	6.98%
Complete graph minus one edge	8	6.20%
None Of These	15	11.63%
Did Not Know	30	23.26%
Not Answered	3	2.33%

Table 7.16 Table of responses given by students for the NI question asking to find the number of vertices in a partition of an adjacency matrix corresponding to a bipartite graph.

Similar to the “visual” clone of this question, percentages of distracters triggered are somewhat similar. However, there is one exception with the response, “Did Not Know”, where over 23% of students who saw this question selected this answer.

7.3.6.2 Finding Which Combination of Properties Does Not Correspond to a Planar Graph

The second question asked to find the adjacency matrix that corresponded to a planar graph. The results generated can be seen in Table 7.17.

Outcome name	Times answered	Percentage of times answered
Correct	139	98.58%
Wrong answer 1	0	
Wrong answer 2	0	
Wrong answer 3	0	
Wrong answer 4	0	
None of these	0	
Did Not Know	0	
Not Answered	2	1.42%

Table 7.17 Table of responses given by students for the question asking to determine for which combination of properties the corresponding graph would definitely be non-planar.

139 attempts were performed on this question and all of them answered the question correctly, causing the facility to be 1. The index of discrimination for this question is -2, suggesting that this question was not useful in measuring the same skills as the test overall. Therefore, this question should only be used in practice mode and never used for an online, invigilated assessment.

7.3.7 Spanning Trees

7.3.7.1 Determining the Number of Spanning Trees when Branches are Given as Adjacency Matrices

There were three questions in the logical / mathematical question set that looked at spanning trees; two of these are RNI questions and one is a MC question. The first question asked to determine the number of spanning trees that are in a graph given a set of subgraphs as branches and appearing as adjacency matrices. The results generated are shown in Table 7.18.

QuestionMark's statistical results state that 76 attempts were made for this question, but clearly only 9 attempts were actually performed. The facility for this question was 0.046, implying that very few students were able to answer it at all. The index of discrimination, given as 0.286, shows that this question only helped partially to measure the same skills as the test overall.

Outcome name	Times answered	Percentage of times answered
Correct	2	2.11%
Wrong	0	
Multiplied numbers of spanning trees in subgraphs by their corresponding numbers of copies.	1	1.05%
Initial matrix calculation was opposite of what it should have been.	0	
Calculated determinants of adjacency matrices for each subgraph.	6	6.32%
Calculated determinants of adjacency matrices for each subgraph, then multiplied numbers of spanning trees in subgraphs by their corresponding numbers of copies.	0	
Not Answered	86	90.53%

Table 7.18 Table of responses given by students for the question asking to determine the number of spanning trees in a graph given a set of subgraphs as adjacency matrices, along with their respective numbers of copies within the graph.

Similar to the visual question set, an overwhelming number of students chose not to answer this question at all, when they should have made some effort in attempting it. This question is important for students to attempt as it shows the importance of dealing with calculations within graph theory and so, more emphasis should be placed on this question.

7.3.7.2 Determining the Number of Spanning Trees of a Graph, Given Its Adjacency Matrix

The second question asked to determine the number of spanning trees that are in a graph, given its adjacency matrices. The results generated are shown in Table 7.19.

QuestionMark's statistical results state that 78 attempts were made for this question, but clearly only 33 attempts were actually performed. The facility for this question was 0.317, implying that although the question was difficult, a good number of students were able to answer the question. The index of discrimination, given as 0.592, shows that this question helped well to measure the same skills as the test overall.

Outcome name	Times answered	Percentage of times answered
Correct	23	27.71%
Wrong	0	
All values in the matrix were kept positive before continuing with the calculations.	0	
Calculated the determinant of the difference of the degree matrix and the adjacency matrix.	7	8.43%
Calculated the absolute value of a cofactor of the adjacency matrix.	3	3.61%
Not Answered	50	60.24%

Table 7.19 Table of responses given by students for the question asking to determine the number of spanning trees in a graph given its adjacency matrix.

Similar to the visual question set, an overwhelming number of students chose not to answer this question at all. This question is important for students to attempt as it shows the importance of dealing with calculations within graph theory and so, more emphasis should be placed on this question. Similar to the visual question set, the distracter that is created by calculating the determinant of the difference of the degree matrix and the adjacency matrix was chosen by a number of students. However, it is less obvious to note this in the logical / mathematical question set as the graph does not readily appear. Therefore, in this case, choosing this result is not as obviously incorrect to the student, even though it should be expected that the corresponding graph, in most cases, will be connected.

As noted with the visual question set, the distracter created by calculating the absolute value of a cofactor of the adjacency matrix could only have been triggered during these attempts if a student takes the (1,1)-cofactor of the adjacency matrix. Similar to the question in the visual question set, the code behind this question has since allowed for other diagonal factors to be selected.

7.3.7.3 Finding a Proper Spanning Tree for a Graph, Given the Corresponding Adjacency Matrices

The third question asked to find a spanning tree that corresponded to a given graph; the graph and all MC options were given as adjacency matrices. The results generated are shown in Table 7.20. 76 attempts were performed for this question. The facility for this question was 0.272, implying that although the question was difficult, some students were able to answer the question correctly. The index of discrimination was 0.42, showing that this question helped to measure the same skills as the test overall.

Outcome name	Times answered	Percentage of times answered
Correct	18	23.08%
Cycle created.	7	8.97%
Not connected.	9	11.54%
It is a tree, but it does not correspond to the graph	8	10.26%
Disconnected subtrees.	4	5.13%
None Of These	6	7.69%
Did Not Know	24	30.77%
Not Answered	2	2.56%

Table 7.20 Table of responses given by students for the question asking to determine the spanning tree for a particular graph.

The results obtained for this question are very different to the cloned, “visual” question. In the visual question, nearly 27% of students selected the distracter that corresponds to a spanning tree for a different graph. However, in this question, only 10.26% of students triggered this distracter. Additionally, the number of those who selected the response, “Did Not Know”, spiked from 8.45% in the visual question to 30.77% in the logical / mathematical question. One reason for this is that it is possible students are not first converting adjacency matrices to graphs before solving these problems, thus causing all questions in this set to become more difficult. By first performing the conversion, students may be able to transform a logical / mathematical problem into a visual problem and then, using this, can answer the question more effectively.

7.4 Quantitative Analysis of Results for the Invigilated Test Session

Question descriptio	Type	Times shown	Max score	Mean score	Facility	Correlation
Bipartite graph / adjacency matrix search	MC	166	2	1.548	0.774	0.431
Find the correct Eulerian cycle in a graph or adjacency matrix (Indirect)	MC	82	3	1.915	0.638	0.596
Find the correct Eulerian cycle in a graph or adjacency matrix (Direct)	MC	79	3	2.203	0.734	0.449
Hamiltonian cycles for graphs and adjacency matrices (Indirect)	MC	71	3	1.775	0.592	0.487
Hamiltonian cycles for graphs and adjacency matrices (Direct)	MC	89	3	2.056	0.685	0.412
Find the planar graph or adjacency matrix (Maze)	MC	65	5	2.015	0.403	0.705
Find the planar graph or adjacency matrix (Student Teachers)	MC	43	5	2.05	0.41	0.743
Find the planar graph or adjacency matrix (Direct)	MC	52	5	2.288	0.458	0.755
Find the simple connected graph given the graphs or adjacency matrices (Direct)	MC	94	1	0.798	0.798	0.435
Find the simple connected graph given the graphs or adjacency matrices (Indirect)	MC	79	1	0.835	0.835	0.167
Number of spanning trees (with branches) using graphs or adjacency matrices	RNI	23	6	0.522	0.087	0.596
Spanning trees using adjacency matrices for a graph (Link Between Towns)	MC	10	3	0.5	0.167	0.04
Spanning trees using adjacency matrices for a graph (Business Departments)	MC	21	3	0.857	0.286	0.132
Spanning trees using adjacency matrices for a graph (Direct)	MC	20	3	1.3	0.433	0.574
Spanning trees using graphs for an adjacency matrix (Direct)	MC	20	3	1.5	0.5	0.598
Spanning trees using graphs for an adjacency matrix (Corporate Business)	MC	20	3	1.632	0.544	0.517
Spanning trees using graphs for an adjacency matrix (University Student Services)	MC	11	3	1.636	0.545	0.669
Spanning trees using graphs for an adjacency matrix (Link Between Towns)	MC	17	3	1.647	0.549	0.479
Spanning trees using adjacency matrices for a graph (University Student Services)	MC	13	3	2	0.667	0.622

Table 7.21 Table of quantitative results from practice questions looking only at adjacency matrices.

The quantitative results in this section look at the series of questions for an invigilated test session in graph theory. Each question combined graphs and adjacency matrices, compelling students to use visual and logical / mathematical intelligences in order to answer questions. A maximum of five attempts were given during this session and students had fifty-five minutes to complete the test.

7.4.1 Discrimination and Other Quantitative Results

There were 19 questions given in the invigilated test session, which composed of contextualised and decontextualised questions being asked separately. However, for each topic, one question is selected randomly and is given to a student to answer. Therefore, each test composed of six questions in six topics. Additionally, all questions included graphs and adjacency matrices throughout, thus forcing the student to use visual and logical / mathematical intelligences in order to solve all problems.

The following sections will cover these questions in clusters, depending on the relevant material being tested. Comparisons of the question types and scenarios are explained throughout, detailing possible similarities and differences with identical questions appearing in either the visual or logical / mathematical practice questions.

7.4.2 Simple and Connected Graphs

This topic looked at finding a simple and connected graph or adjacency matrix among a list of candidates. The results obtained from trialling the questions in this topic are found in Table 7.22.

141 attempts were made on these questions. Facilities for both scenarios are very high with 0.798 and 0.835 given to direct question and problem solving question categories respectively. However, discrimination values were 0.435 and 0.167 respectively; this could imply the problem solving question was less helpful in testing the same skills as the test overall.

For both questions, the distracter, “Loops around a pair of vertices” was never selected by students, although all other distracters were selected. This is somewhat surprising as although students are aware to avoid loops, they still managed to be caught by loops around a vertex; this was also the case for the practice attempts.

Outcome name	Times answered	Percentage of times answered	Times answered	Percentage of times answered
Correct	75	79.79%	66	83.54%
Loops around one vertex	4	4.26%	3	3.80%
Loops around a pair of vertices	0		0	
Unconnected	3	3.19%	5	6.33%
Loops (almost) everywhere.	3	3.19%	2	2.53%
None Of These	9	9.57%	3	3.80%
Did Not Know	0		0	
Not Answered	0		0	

Table 7.22 Table of responses given by students who answered questions looking at simple and connected graphs during the invigilated, online assessment. Columns two and three look at direct questions and columns four and five look at problem solving questions.

7.4.3 Bipartite Graphs

This topic looked at finding a bipartite graph or adjacency matrix among a list of candidates. The results from trialling this question are found in Table 7.23.

Outcome name	Times answered	Percentage of times answered
Correct	123	74.10%
Not complete	11	6.63%
Not bipartite.	11	6.63%
Complete graph	4	2.41%
Wheel graphs.	7	4.22%
None Of These	10	6.02%
Did Not Know	0	
Not Answered	0	

Table 7.23 Table of responses given by students who answered questions looking at bipartite graphs during the invigilated, online assessment.

A total of 166 attempts were made on the topic of bipartite graphs. The facility for this question was 0.774, which suggests students were easily able to determine the correct graph. The index of discrimination is 0.431, implying that the question helped somewhat to test the same skills as the assessment overall.

Attempts were made every time this question was viewed. This is especially good to see as students either responded with “Did Not Know” or they did not answer the question at all during the practice attempts. Therefore, it appears as though some confidence was given to students in attempting this question on the test.

7.4.4 Hamiltonian Graphs

This topic looked at finding a non-Hamiltonian graph or adjacency matrix among a list of candidates. The results obtained from trialling the questions in this topic are found in Table 7.24. A total of 160 attempts were made on the topic of bipartite graphs. The facilities for this question were 0.685 for the direct question and 0.592 for the problem solving question, which suggest students were able to determine how to solve this problem regardless of the question wording. The indices of discrimination are 0.412 for the direct question and 0.487 for the problem solving question, implying that the question helped well to test the same skills as the assessment overall.

Outcome name	Times answered	Percentage of times answered	Times answered	Percentage of times answered
Correct	61	68.54%	42	59.15%
Hamiltonian and Eulerian cycles	3	3.37%	4	5.63%
Loop-free graph with $\lvert V\rvert \geq 3$, where $\deg(x) \geq \frac{n}{2}$, $\forall x \in V$	6	6.74%	6	8.45%
Loop-free, undirected graph with $\lvert V\rvert \geq 3$ and $\lvert E\rvert \geq \binom{n-1}{2}+2$	3	3.37%	4	5.63%
Wheel Graph	3	3.37%	5	7.04%
None Of These	3	14.61%	9	12.68%
Did Not Know	0		1	1.41%
Not Answered	0		0	

Table 7.24 Table of responses given by students who answered questions looking at Hamiltonian graphs during the invigilated, online assessment. Columns two and three look at direct questions and columns four and five look at problem solving questions.

Attempts were made every time this question was viewed. This is especially good to see as students either responded with “Did Not Know” or they did not answer the question at all during the practice attempts. Therefore, it appears as though some confidence was given to students in attempting this question on the test.

7.4.5 Eulerian Graphs

This topic looked at finding an Eulerian graph or adjacency matrix among a list of candidates. The results obtained from trialling the questions in this topic are found in Table 7.25.

A total of 161 attempts were made on the topic of bipartite graphs. The facilities for this question were 0.734 for the direct question and 0.638 for the problem solving question, which suggest students were able to determine how to solve this problem regardless of the question wording. The indices of discrimination are 0.449 for the direct question and 0.596 for the problem solving question, implying that the question helped well to test the same skills as the assessment overall.

For the problem solving question style, three attempts resulted with students either admitting they did not know the answer or simply refusing to answer the question. Additionally, there is a significant percentage of students for either question style who selected the distracter that involves an edge added to a graph that was initially Eulerian. This provides further evidence to suggest this distracter is useful for catching students making errors in this topic.

Outcome name	Times answered	Percentage of times answered	Times answered	Percentage of times answered
Correct	54	68.35%	47	56.63%
Almost a cycle, but the two ends are not connected	2	2.53%	2	2.41%
Two cycles connected by one edge.	2	2.53%	7	8.43%
Edge added to an already Eulerian cycle.	10	12.66%	9	10.84%
Two unconnected subgraphs.	3	3.80%	3	3.61%
None Of These	8	10.13%	12	14.46%
Did Not Know	0		2	2.41%
Not Answered	0		1	1.20%

Table 7.25 Table of responses given by students who answered questions looking at Eulerian graphs during the invigilated, online assessment. Columns two and three look at direct questions and columns four and five look at problem solving questions.

7.4.6 Planar Graphs

This topic looked at finding a planar graph or adjacency matrix among a list of candidates. The results obtained from trialling the questions in this topic are found in Table 7.26.

A total of 157 attempts were made on the topic of planar graphs. The facilities for this question were 0.458 for the direct question, 0.403 for the maze scenario, and 0.41 for the student teachers scenario. This suggests that students were able to cope with the different question wordings, but also that they found the direct question to be slightly easier. The indices of discrimination are 0.755 for the direct question, 0.705 for the maze scenario, and 0.743 for the student teachers scenario, implying that the question helped well to test the same skills as the assessment overall, but again, also implies that the direct question style was more helpful in assessing students.

Outcome name	Times (direct) answered	Percentage of times answered	Times (maze) answered	Percentage of times answered	Times (student teachers) answered	Percentage of times answered
Correct	23	44.23%	26	40%	16	37.21%
K_5 subgraph	6	11.54%	4	6.15%	4	9.30%
$K_{3,3}$ subgraph	7	13.46%	18	27.69%	3	6.98%
Both subgraphs included	4	7.69%	4	6.15%	6	13.95%
Complete graph minus one edge	5	9.62%	1	1.54%	5	11.63%
None Of These	3	5.77%	11	16.92%	4	9.30%
Did Not Know	4	7.69%	1	1.54%	2	4.65%
Not Answered	0		0		3	6.98%

Table 7.26 Table of responses given by students who answered questions looking at planar graphs during the invigilated, online assessment. Columns two and three look at direct questions, columns four and five look at problem solving questions using a maze scenario, and columns six and seven look at problem solving questions using a student teachers scenario.

Ten attempts made at this question had students either not answering the question or admitting they did not know the answer to the question. Also, a significant percentage of students selected the distracters that had only one of the two key subgraphs included. A significant number of students answering the student teachers scenario questions used the distracter including both subgraphs. These distracters can therefore be seen as useful for this question. Surprisingly, though, a significant number of students also triggered the distracter that consisted of a complete graph minus one edge. Therefore, it can also be considered useful in catching students making errors.

7.4.7 Spanning Trees

7.4.7.1 The Number of Spanning Trees when Branches are Given

This topic looked at determining the number of spanning trees that exist in a graph, given its branches. The results obtained from trialling the question in this topic are found in Table 7.27.

This question was randomly selected amongst a selection of nine questions, of which this was the only RNI question. The facility for this question was 0.087, implying that students still found this question to be incredibly difficult, even though they had the chance to trial it beforehand. The index of discrimination, however, is 0.596, implying that the question helped very well to test the same skills as the assessment overall.

Outcome name	Times answered	Percentage of times answered
Correct	2	8.70%
Wrong	0	
Multiplied numbers of spanning trees in subgraphs by their corresponding numbers of copies.	0	
Initial matrix calculation was opposite of what it should have been.	0	
Calculated determinants of adjacency matrices for each subgraph.	0	
Calculated determinants of adjacency matrices for each subgraph, then multiplied numbers of spanning trees in subgraphs by their corresponding numbers of copies.	0	
Not Answered	21	91.30%

Table 7.27 Table of responses given by students who answered questions looking at determining the number of spanning trees in a graph, given the branches of the graph.

As noted in the practice tests, a significant number of students refused to answer this question. Most students were simply unwilling to answer the RNI questions in either the practice sets or the invigilated test session, but it is this question in particular that normally appears on the final examinations they perform at the end of the academic year. Therefore, more emphasis needs to be placed on RNI questions within graph theory so that students could be compelled to answer such questions in the future.

7.4.7.2 MC Questions on Spanning Trees

This topic looked at finding a spanning tree that corresponded to a particular graph or adjacency matrix. The results obtained from trialling the questions in this topic are found in Table 7.28.

Outcome name	Times (1A) given	Percentage of times answered	Times (1B) given	Percentage of times answered	Times (1C) given	Percentage of times answered	Times (1D) given	Percentage of times answered
Correct	4	19.05%	8	40%	1	10%	8	61.54%
Cycle created	1	4.76%	2	10%	2	20%	2	15.38%
Not connected	3	14.29%	3	15%	1	10%	0	
It is a tree, but it does not correspond to the graph	6	28.57%	2	10%	2	20%	2	15.38%
Disconnected subtrees	2	9.52%	4	20%	3	30%	0	
None Of These	5	23.81%	1	5%	0		1	7.69%
Did Not Know	0		0		1	10%	0	
Not Answered	0		0		0		0	
Outcome name	**Times (2A) given**	**Percentage of times answered**	**Times (2B) given**	**Percentage of times answered**	**Times (2C) given**	**Percentage of times answered**	**Times (2D) given**	**Percentage of times answered**
Correct	9	45%	10	50%	9	52.94%	5	45.45%
Cycle created	3	15%	4	20%	1	5.88%	2	18.18%
Not connected	0		0		2	11.76%	0	
It is a tree, but it does not correspond to the graph	4	20%	0		1	5.88%	3	27.27%
Disconnected subtrees	1	5%	1	5%	1	5.88%	0	
None Of These	2	10%	4	20%	2	11.76%	1	9.09%
Did Not Know	0		1	5%	1	5.88%	0	
Not Answered	1	5%	0		0		0	
LEGEND	1	Adjacency matrices for a graph			A	Business Departments		
	2	Graphs for an adjacency matrix			B	Direct Question		
					C	Links Between Towns		
					D	University Student Services		

Table 7.28 Table of responses given by students who answered MC questions looking at finding a spanning tree for a particular graph or adjacency matrix. All results are shown for each scenario and combination of graphs and adjacency matrices. A legend is provided.

The facilities and indices of discrimination for these questions were generally good, with the exceptions of two questions, namely:

- Finding a spanning tree using adjacency matrices and given a graph, using the business departments scenario
- Finding a spanning tree using adjacency matrices and given a graph, using the link between towns scenario

For these two scenarios, the facilities were 0.286 and 0.167 respectively, which imply that students have found these questions to be somewhat harder. It is possible that students may not have been able to comprehend the scenario well enough to understand what it was asking. However, it is also possible that although they did well in other questions where adjacency matrices were given as MC options, students may have simply found it more difficult to look at adjacency matrices as MC options rather than graphs; this is especially evident by the fact that facilities were generally higher overall between option types using the same scenarios. Additionally, the indices of discrimination for these scenarios are 0.132 and 0.04, suggesting that they did not help to assess the same skills as the test overall. It is worth noting that of the four questions involving graphs as MC options, it was with these two scenarios that the lowest indices of discrimination were recorded, although both were still relatively high.

A strongly significant number of students selected the distracter involving a spanning tree for another graph during the test, which again implies they may not have been reading the question properly. However, it is also worth noting that depending on the scenario and question style, either a significant number of students or no students at all triggered the distracter that involves a disconnected subgraph. The selection of this distracter occurred more often in questions where adjacency matrices were given as MC options.

7.5 Comparisons of Facilities and Indices of Discrimination

Another method for determining the effectiveness of the questions trialled is to compare the facilities with the indices of discrimination for all of the questions attempted. By comparing facilities with their corresponding indices of discrimination, it is possible to find outliers, which may provide further evidence to suggest particular levels of effectiveness of questions trialled.

Before comparing, it should be noted that since the practice question asking to determine for which combination of properties a planar graph is not formed generated a facility of 1 and an index of discrimination of -2, this would clearly appear as an outlier for any comparison performed. Also, it has already been noted that this question has not helped students to learn the course material better. Therefore, the facility and index of discrimination for this question will appear removed from any analysis performed in this section.

The first graph, shown in Figure 7.1, shows the comparisons of the graph theory questions presented based on their question style. First, recall that the test questions all contained visual and logical / mathematical components. Therefore, when analysing these results, it is worth comparing the visual with the logical / mathematical, but then to compare both together with the test questions.

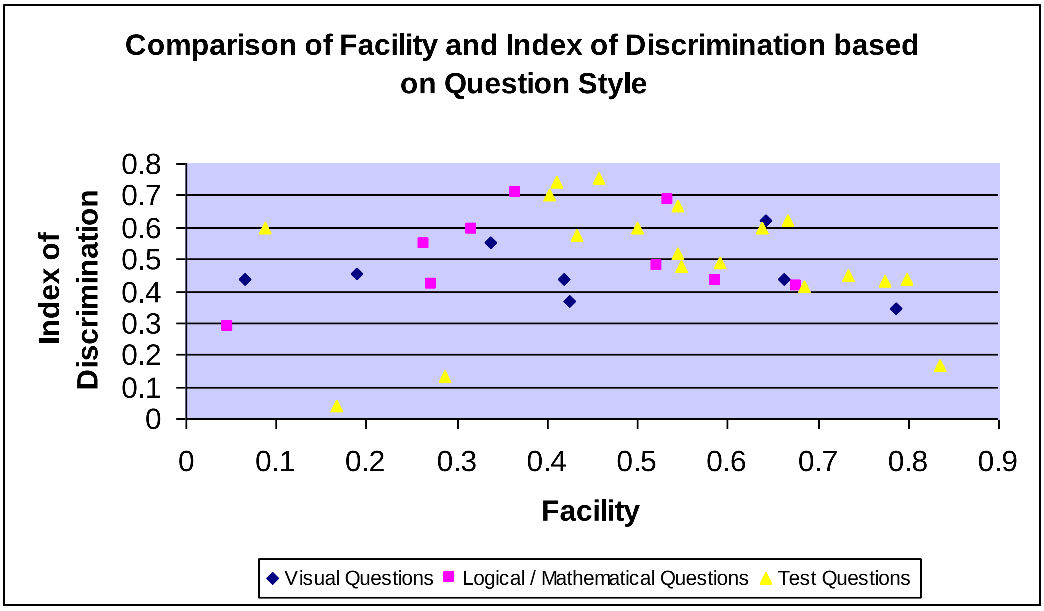


Figure 7.1 Scatter plot of facilities versus corresponding indices of discrimination for attempted Mathletics questions on graph theory, separated by their question styles.

When viewing the visual questions and also the logical / mathematical questions, it appears that there are multiple clusters in each case. For the logical / mathematical questions, these clusters appear somewhat closer to each other than the visual questions, but nonetheless, the distinction of two separate clusters is somewhat clear, along with one outlier. The outlier in the case of the logical / mathematical questions refers to the logical / mathematical RNI question asking to determine the number of spanning trees in a graph given the branches of the graph. Recall that the majority of students refused to answer this question and so, this outlier can be justified on this account. If more students were willing to attempt this problem, then a more accurate comparison could be made with respect to this particular question. For all logical / mathematical questions aside from the outlier, the indices of discrimination remain above 0.4, which is encouraging as this shows significance of the questions in relation to the material taught in the course / module.

The visual questions vary greatly in facility, but remain within a small range of indices of discrimination, ranging from just under 0.4 to just over 0.6. This is still encouraging to see, but the varied facility values helps to create three clusters. The leftmost cluster refers to the two RNI questions on spanning trees. Since there were only the two visual RNI questions on this topic, this shows that students may find it difficult to engage with a “visually written” question and then solve it using logical / mathematical skills. The middle cluster refers to MC questions on Hamiltonian graphs, planar graphs, and spanning trees. This is, perhaps the fairest cluster in the set as the facilities in each case are near 0.5, which is considered an optimal location for teachers to set their questions. The indices of discrimination for each of these questions also show that the questions are indeed significant as part of the overall test. The rightmost cluster refers to MC questions on simple and connected graphs, bipartite graphs, and Eulerian graphs. These results show that although the questions are significant as part of the overall assessment, students nonetheless find the questions easy to answer correctly.

The test questions all seem to be clustered well, with four exceptions. Three of the questions, all of whom have outliers on the left side of the graph in Figure 7.1, refer to various questions on spanning trees. Students must consider this topic to be difficult and so, either refuse to attempt the questions, as is normally the case with any RNI questions, or give a poor attempt in the case of any MC questions given. However, the fourth outlier, which appears on the far right side of the graph, refers to the problem solving question on simple and connected graphs. The facility suggests that students find the question easy to answer correctly, but the index of discrimination is low, which suggests that it is not helping much as part of the overall assessment. Therefore, it should be worth looking at both questions on simple and connected graphs, i.e. not just the question that generated this outlier result, to see how to make it more effective as part of the overall assessment.

So far, it has been noted that RNI questions, questions on spanning trees, and questions on simple and connected graphs may have some problems in terms of their effectiveness as part of an overall assessment. To explore this further, it helps to regroup the data points based on the question topics and types. The graph shown in Figure 7.2 shows a scatter plot of the facilities and indices of discrimination for the graph theory questions based upon the topics presented.

As noted earlier, the questions on simple and connected graphs all appear to have high facilities, which suggest that students found these questions easy to answer correctly. With one exception, which was noted earlier, all of the indices of discrimination for this topic appear around 0.4, which suggest there is some benefit to this topic as part of the overall assessment.

The questions on bipartite graphs all appear to be aligned well, each with indices of discrimination between 0.4 and 0.5. As expected, the facilities for these questions are high and so, students find this topic to be somewhat easy to understand. Nonetheless, with a reasonable range of indices of discrimination and a decent spread of data, it is likely that these questions are already significant and beneficial to the assessment.

The questions on Hamiltonian cycles all appear to have indices of discrimination hovering around 0.5, which is encouraging. However, the facilities for these questions differ greatly. This could suggest that the question styles had a significant effect on students as they may have found it difficult to understand what was being asked of them in some of the questions. However, upon further inspection, it is noted that the left two points refer to the practice test questions and the right two points refer to the invigilated test questions. Therefore, it is likely that the practice questions played a significant role in helping students to understand this topic better in order to perform better on the invigilated test.

The questions on Eulerian cycles have high facilities, but also have very high indices of discrimination. This suggests that the questions are easy to answer correctly, but that they are also very significant as part of the overall assessment. As such, it is worth keeping these questions as they are for future considerations in other assessments.

The questions on planar graphs all have facilities between 0.3 and 0.5, which is reasonably good. Three of the four questions appear in one cluster, with an outlier appearing with an index of discrimination of 0.555. This outlier refers to the visual practice question given to students, which is very interesting to note as it was expected students would find this question to be easier to solve than the logical / mathematical question. However, the other three questions all have indices of discrimination larger than 0.7. Nonetheless, it may be suggested that all four questions are assessing well the same skills as the test overall.

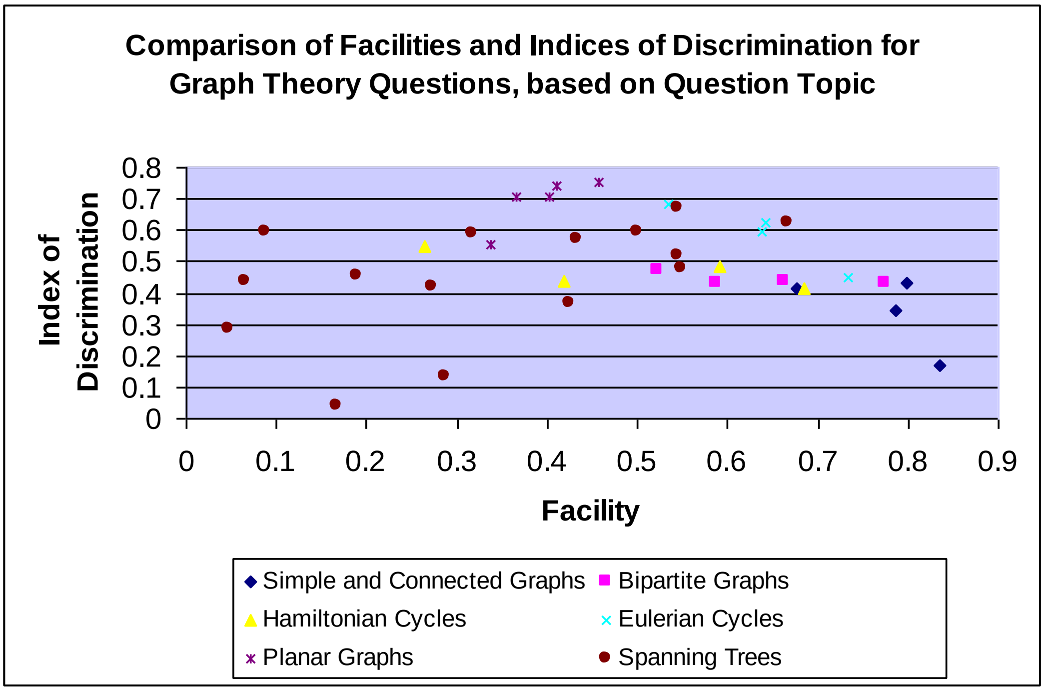


Figure 7.2 Scatter plot of facilities versus corresponding indices of discrimination for Mathletics questions on graph theory, separated by their question topics.

The questions on spanning trees, however, provide a much different story as students appeared to find them more challenging. It is quite surprising, though, that the facilities range from 0.046 to 0.667. Similarly, the indices of discrimination range from 0.04 to 0.669. Therefore, it is difficult to determine the effectiveness of this topic as part of the overall assessment. However, a further analysis can be performed by analysing only the facilities and indices of discrimination of these questions based on additional information about these questions.

The graph appearing in Figure 7.3 shows the facilities and indices of discrimination for the different scenarios and question types for the questions on spanning trees. For five of the scenarios or question types, the variations of the questions appear not to change the overall results much. However, for the business departments scenario and the links between towns scenario, there is a sharp difference between the results of one variant and the results of the other variant. In both cases, it was with the questions involving adjacency matrices as options that the lower facilities and indices of discrimination appeared. Since the results for the questions with graphs appearing as options produced reasonable results, it may be suggested that the variants with adjacency matrices as options created a significant difficulty for students when attempting these problems.

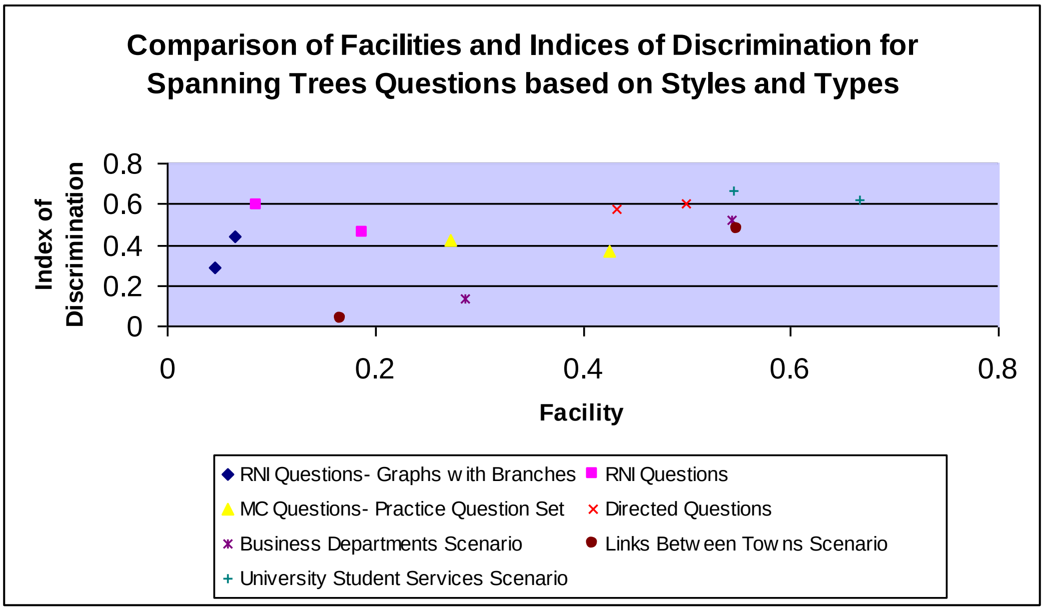


Figure 7.3 Scatter plot of facilities versus corresponding indices of discrimination for Mathletics questions on spanning trees, separated by their scenarios and types.

Additionally, by looking at the smallest of each pair in terms of facility, it can be noted that the logical / mathematical questions were generally seen as more difficult to answer. There is one exception with the university students services scenario, but the comparison of the two data points show close results for facility and index of discrimination in comparison to the other pairs. In addition to the observation of students' unwillingness to respond to the RNI questions, it may be concluded that students find it difficult to comprehend and apply course / module material in graph theory in relation to logical / mathematical intelligences.

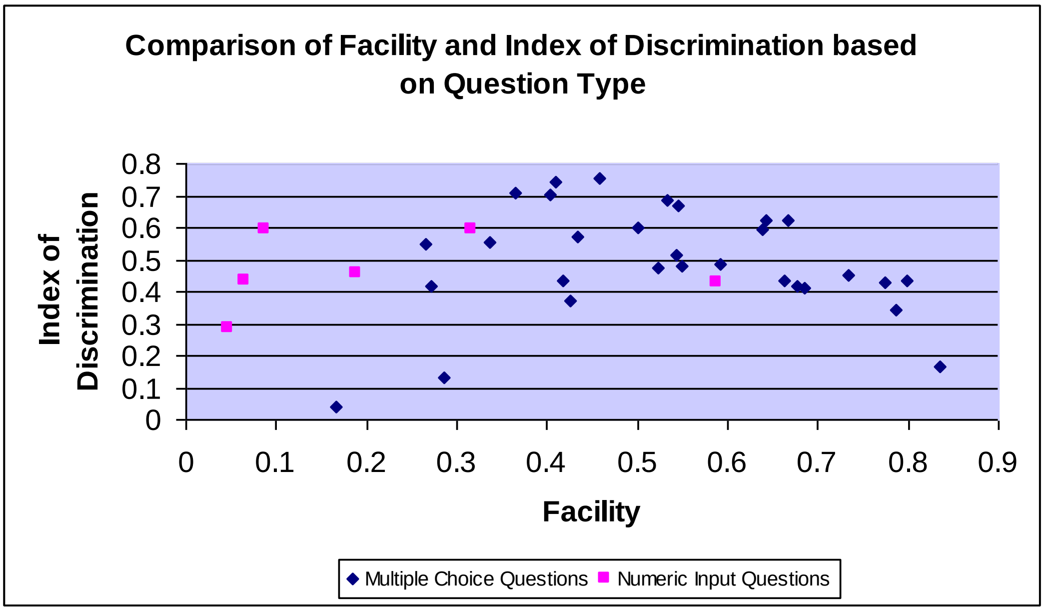


Figure 7.4 Scatter plot of facilities versus corresponding indices of discrimination for Mathletics questions on graph theory, separated by their question types.

The last graph, presented in Figure 7.4, shows the facilities and indices of discrimination for the graph theory questions based on their question types. It can be noted that the numeric input questions, with one exception, were more difficult for students to attempt than MC questions. Recall that most students refused to attempt the RNI questions on spanning trees and therefore, this is reflected within these results. The outlier for the numeric input questions refers to the NI question given for bipartite graphs in the logical / mathematical practice question set, whereas all other questions were RNI questions on spanning trees. Therefore, this outlier is not surprising to notice. The MC questions, however, appear to be well clustered and with only a few outliers. Two of the outliers, appearing towards the bottom of the graph, both refer to MC questions on spanning trees involving adjacency matrices as options. However, this was referred to earlier as these questions involve the business departments and links between towns scenarios. This therefore provides further evidence to suggest that students are finding it difficult to link their logical / mathematical intelligences to this learning material. The other outlier refers to the problem solving question on simple and connected graphs, which was discussed earlier.

7.6 Quantitative, Comparative Analysis of Mathletics Questions using ANOVA and Student t-Tests

7.6.1 Comparison of Questions Involving Simple and Connected Graphs

A two-factor, ANOVA experiment without replication was performed to determine if differences existed between different outcomes or if differences existed between different question styles for the questions performed on simple and connected graphs. The results appear in Table 7.29.

Using P-values, it can be easily determined that there are significant differences in the means for the different outcomes. However, this should be somewhat expected since each distracter is likely to have some motivation behind students choosing it. Additionally, removing the outcomes, "Not answered", "Did not know", and "None of These", the P-value for this source increases only to 0.000832. However, at the $\alpha = 5\%$ level of significance, there is not enough evidence to reject H_0 for the different question styles. Therefore, we can conclude that may be possible for there to be some commonality in answering the different question styles for the questions given on simple and connected graphs. This implies that students may have been successfully able to distinguish between the different question styles in order to master this topic.

Source	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Outcomes	75548.38	7	10792.63	10.10821	1.65×10^{-5}	2.487578
Question Styles	7224.625	3	2408.208	2.255493	0.111651	3.072467
Error	22421.88	21	1067.708			
Total	105194.9	31				

Table 7.29 ANOVA two-factor (without replication) Table for the practice and test questions involving simple and connected graphs and using Microsoft Excel.

7.6.2 Comparison of Questions Involving Bipartite Graphs

A two-factor, ANOVA experiment without replication was performed to determine if differences existed between different outcomes or if differences existed between different question styles for the questions performed on bipartite graphs. The results appear in Table 7.30.

Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Outcomes	36029.17	7	5147.024	15.2534	$1.46\text{x}10^{-5}$	2.764199
Question Styles	1652.583	2	826.2917	2.448747	0.12248	3.738892
Error	4724.083	14	337.4345			
Total	42405.83	23				

Table 7.30 ANOVA two-factor (without replication) Table for the practice and test questions involving bipartite graphs and using Microsoft Excel.

Using the P-values, it can be easily determined that there are significant differences in the means for the different outcomes. Additionally, removing the outcomes, “Not answered”, “Did not know”, and “None of These”, the P-value for this source increases only to 0.000437. However, at the $\alpha = 5\%$ level of significance, there is not enough evidence to reject H_0 for the different question styles. Therefore, we can conclude that it is *possible* for there to be some commonality in answering the different question styles for the questions given on bipartite graphs. This implies that students may have been successfully able to distinguish between the different question styles in order to master this topic.

7.6.3 Comparison of Questions Involving Hamiltonian Graphs

A two-factor, ANOVA experiment without replication was performed to determine if differences existed between different outcomes or if differences existed between different question styles for the questions performed on Hamiltonian graphs. The results appear in Table 7.31.

Using the P-values, it can be easily determined that there are significant differences in the means for the different outcomes. Additionally, removing the outcomes, “Not answered”, “Did not know”, and “None of These”, the P-value for this source increases only to 0.000437. However, at the $\alpha = 5\%$ level of significance, there is also enough evidence to reject H_0 for the different question styles. Therefore, there is also a significant effect on the question styles implemented.

Source	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Outcome	13950.38	7	1992.911	11.44059	6.36×10^{-6}	2.487578
Question Style	5377.375	3	1792.458	10.28987	0.000226	3.072467
Error	3658.125	21	174.1964			
Total	22985.88	31				

Table 7.31 ANOVA two-factor (without replication) Table for the practice and test questions involving Hamiltonian graphs and using Microsoft Excel.

Since there is a significant effect on the question styles, it is important to determine for which question styles these differences appeared. To do this, a series of t-tests are performed. The results for all t-tests appear in Table 7.32.

This experiment involves six comparisons. Therefore, $EER = 1 - 0.95^6 \approx 0.2649$, which implies there is nearly a 26.5% probability of a Type I error occurring with at least one of these experiments. Also, this set of experiments caused rejections of H_0 for the following comparisons:

- Visual practice set vs. Direct test set
- Visual practice set vs. Problem Solving test set
- Logical / mathematical practice set vs. Direct test set
- Logical / mathematical practice set vs. Problem Solving test set

It is interesting to note that the rejections of H_0 occur for comparisons of a practice question set with a test question set. This seems somewhat logical, though, as students were given ample time to complete the practice question sets and as often as they considered it necessary, whereas they only had one hour to attempt a maximum of five attempts of the direct and problem solving questions. Also, for all such cases, both the one-tailed and two-tailed tests failed and it may be concluded that $H_1 : \mu_1 > \mu_2$ for $\alpha = 5\%$. This implies that students did not appear to retain necessary information in the practice attempts for this material.

	Practice-Graphs	*vs. Practice-Matrices*	*vs. Test-Direct*	*vs. Test-Problem Solving*
Mean	35.625	36.125	11.125	8.875
Variance	1374.268	530.9821	423.2679	186.9821
Observations	8	8	8	8
t Statistic		**-0.06463**	**3.5167**	**3.071504**
One-tailed P-value		0.475139	0.004885	0.009015
One-tailed critical value for t		**1.894579**	**1.894579**	**1.894579**
Two-tailed P-value		0.950278	0.009771	0.018029
Two-tailed critical value for t		**2.364624**	**2.364624**	**2.364624**

	Practice-Matrices	*vs. Test-Direct*	*vs. Test-Problem Solving*
Mean	36.125	11.125	8.875
Variance	530.9821	423.2679	186.9821
Observations	8	8	8
t Statistic		**4.098125**	**4.693719**
One-tailed P-value		0.002292	0.001112
One-tailed critical value for t		**1.894579**	**1.894579**
Two-tailed P-value		0.004584	0.002225
Two-tailed critical value for t		**2.364624**	**2.364624**

	Test- Direct	*vs. Test-Problem Solving*
Mean	11.125	8.875
Variance	423.2679	186.9821
Observations	8	8
t Statistic		**0.908475**
One-tailed P-value		0.196914
One-tailed critical value for t		**1.894579**
Two-tailed P-value		0.393829
Two-tailed critical value for t		**2.364624**

Table 7.32 Table of T distribution results for all style pairings for questions on Hamiltonian graphs. Results highlighted in red indicate where the null hypothesis is rejected.

For the comparison of the two practice question sets and the comparison of the two test question sets, there is not enough evidence to conclude that there is a significant difference between these groups. This implies that it is possible students were able to distinguish between visual and logical / mathematical question styles, as well as between direct and problem solving question styles.

7.6.4 Comparison of Questions Involving Eulerian Graphs

A two-factor, ANOVA experiment without replication was performed to determine if differences existed between different outcomes or if differences existed between different question styles for the questions performed on Eulerian graphs. The results appear in Table 7.33.

Using the P-values, it can be easily determined that there are significant differences in the means for the different outcomes. Also, at the $\alpha = 5\%$ level of significance, there is also enough evidence to reject H_0 for the different question styles. Therefore, there is also a significant effect on the question styles used.

Source	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Outcomes	30574.72	7	4367.817	10.48108	1.26×10^{-5}	2.487578
Question Styles	5179.344	3	1726.448	4.142809	0.018724	3.072467
Error	8751.406	21	416.7336			
Total	44505.47	31				

Table 7.33 ANOVA two-factor (without replication) Table for the practice and test questions involving Eulerian graphs and using Microsoft Excel.

	Practice-Graphs	*vs. Practice-Matrices*	*vs. Test-Direct*	*vs. Test-Problem Solving*
Mean	33.625	37.25	9.875	10.375
Variance	3040.268	2012.5	330.9821	234.2679
Observations	8	8	8	8
t Statistic		**-0.55677**	**1.806658**	**1.637494**
One-tailed P-value		0.297516	0.056885	0.072769
One-tailed critical value for t		**1.894579**	**1.894579**	**1.894579**
Two-tailed P-value		0.595031	0.11377	0.145539
Two-tailed critical value for t		**2.364624**	**2.364624**	**2.364624**
		Practice-Matrices	*vs. Test-Direct*	*vs. Test-Problem Solving*
Mean		37.25	9.875	10.375
Variance		2012.5	330.9821	234.2679
Observations		8	8	8
t Statistic			**2.815537**	**2.538462**
One-tailed P-value			0.012969	0.019375
One-tailed critical value for t			**1.894579**	**1.894579**
Two-tailed P-value			0.025939	0.038751
Two-tailed critical value for t			**2.364624**	**2.364624**
			Test- Direct	*vs. Test-Problem Solving*
Mean			9.875	10.375
Variance			330.9821	234.2679
Observations			8	8
t Statistic				**-0.38592**
One-tailed P-value				0.355508
One-tailed critical value for t				**1.894579**
Two-tailed P-value				0.711015
Two-tailed critical value for t				**2.364624**

Table 7.34 Table of T distribution results for all style pairings for questions on Eulerian graphs. Results highlighted in red indicate where the null hypothesis is rejected.

Since there is a significant effect on the question styles, it is important to determine for which question styles these differences appeared. To do this, a series of t-tests are performed. The results for all t-tests appear in Table 7.34.

This experiment involves six comparisons. Therefore, $EER = 1 - 0.95^6 \approx 0.2649$, which implies there is, at most, a 26% probability of a Type I error occurring with at least one of these experiments. Also, this set of experiments caused rejections of H_0 for the following comparisons:

- Logical / mathematical practice set vs. Direct test set
- Logical / mathematical practice set vs. Problem Solving test set

It is interesting to note that the rejections of H_0 occur only for comparisons involving the logical / mathematical practice question set with either test question set. Also, for all such cases, both the one-tailed and two-tailed tests failed and it may be concluded that $H_1 : \mu_1 > \mu_2$ for $\alpha = 5\%$. This implies that students may have found it difficult to study this topic using adjacency matrices in every question.

For all other comparisons, there is not enough evidence at $\alpha = 5\%$ to suggest that there is a difference in the means of the compared question sets. This could imply that students generally were able to grasp this learning material, provided that adjacency matrices did not appear in the questions.

7.6.5 Comparison of Questions Involving Planar Graphs

A two-factor, ANOVA experiment without replication was performed to determine if differences existed between different outcomes or if differences existed between different question styles for the questions performed on planar graphs. The results appear in Table 7.35.

Using the P-values, it can be easily determined that there are significant differences in the means for the different outcomes. Also, at the $\alpha = 5\%$ level of significance, there is also enough evidence to reject H_0 for the different question styles. Therefore, there is also a significant effect on the question styles used.

Source	SS	df	MS	F	P-value	F crit
Outcomes	4152.4	7	593.2	7.221251	5.76×10^{-5}	2.35926
Question Styles	3907.1	4	976.775	11.89066	9.03×10^{-6}	2.714076
Error	2300.1	28	82.14643			
Total	10359.6	39				

Table 7.35 ANOVA two-factor (without replication) Table for the practice and test questions involving planar graphs and using Microsoft Excel.

Since there is a significant effect on the question styles, it is important to determine for which question styles these differences appeared. To do this, a series of t-tests are performed. The results for all t-tests appear in Table 7.36.

This experiment involves ten comparisons. Therefore, $EER = 1 - 0.95^{10} \approx 0.40$, which implies there is approximately a 40% probability of a Type I error occurring with at least one of these experiments. Also, this set of experiments caused rejections of H_0 for the following comparisons:

- Visual practice set vs. Logical / mathematical practice set
- Visual practice set vs. Direct test set
- Visual practice set vs. Maze scenario questions in test
- Visual practice set vs. Student teachers scenario questions in test
- Logical / mathematical practice set vs. Direct test set
- Logical / mathematical practice set vs. Maze scenario questions in test (only for one-tailed test)
- Logical / mathematical practice set vs. Student teachers scenario questions in test

It is interesting to note that the rejections of H_0 occur only for comparisons involving the practice question sets with any of the test question sets. Also, for almost all such cases, both the one-tailed and two-tailed tests failed and it may be concluded that $H_1 : \mu_1 > \mu_2$ for $\alpha = 5\%$. In the case of the visual practice set, we can conclude that students found it difficult to study the material in this topic when graphs appeared in the questions. However, in the case of the logical / mathematical practice set, it is necessary to consider its comparison with the maze scenario questions in the test. For the one-tailed t-test, $t_0 > t_{\alpha,n-1}$, therefore, we reject H_0 and conclude $H_1 : \mu_1 > \mu_2$ for $\alpha = 5\%$. However, using the two-tailed t-test, $|t_0| \not> t_{\alpha/2, n-1}$. Therefore, it is possible for the test to "succeed" (although not to absolute certainty) for a value of α smaller than 5%; in fact, for $\alpha = 1\%$, $t_0 = 2.272502 < t_{\alpha/2, n-1} = 2.998$.

	Practice-Graphs	*vs. Practice-Matrices*	*vs. Test-Direct*	*vs. Test-Maze*	*vs. Test-Student Teachers*
Mean	31.875	16.125	6.5	8.125	5.375
Variance	598.125	165.26786	48.857143	89.553571	19.982143
Observations	8	8	8	8	8
t Statistic		**2.8146324**	**4.0284711**	**4.008862**	**3.6388291**
One-tailed P-value		0.0129863	0.0025027	0.0025659	0.00415
One-tailed critical value for t		**1.8945786**	**1.8945786**	**1.8945786**	**1.8945786**
Two-tailed P-value		0.0259726	0.0050055	0.0051317	0.0083
Two-tailed critical value for t		**2.3646243**	**2.3646243**	**2.3646243**	**2.3646243**
		Practice-Matrices	*vs. Test-Direct*	*vs. Test-Maze*	*vs. Test-Student Teachers*
Mean		16.125	6.5	8.125	5.375
Variance		165.26786	48.857143	89.553571	19.982143
Observations		8	8	8	8
t Statistic			**3.2087313**	**2.272502**	**2.9000011**
One-tailed P-value			0.0074421	0.0286329	0.0114929
One-tailed critical value for t			**1.8945786**	**1.8945786**	**1.8945786**
Two-tailed P-value			0.0148843	0.0572658	0.0229859
Two-tailed critical value for t			**2.3646243**	2.3646243	**2.3646243**
			Test-Direct	*vs. Test-Maze*	*vs. Test-Student Teachers*
Mean			6.5	8.125	5.375
Variance			48.857143	89.553571	19.982143
Observations			8	8	8
t Statistic				**-0.855867**	**0.9601829**
One-tailed P-value				0.2101998	0.1844735
One-tailed critical value for t				**1.8945786**	**1.8945786**
Two-tailed P-value				0.4203996	0.368947
Two-tailed critical value for t				**2.3646243**	**2.3646243**
				Test-Maze	*vs. Test-Student Teachers*
Mean				8.125	5.375
Variance				89.553571	19.982143
Observations				8	8
t Statistic					**1.1103588**
One-tailed P-value					0.1517623
One-tailed critical value for t					**1.8945786**
Two-tailed P-value					0.3035245
Two-tailed critical value for t					**2.3646243**

Table 7.36 Table of T distribution results for all style pairings for questions on planar graphs. Results highlighted in red indicate where the null hypothesis is rejected and results highlighted in yellow indicate the rejection of the null hypothesis only for the corresponding one-tailed test.

For all other comparisons, there is not enough evidence at $\alpha = 5\%$ to suggest that there is a difference in the means of the compared question sets.

This could imply that students generally were able to grasp the differences in the direct and problem solving question styles.

7.6.6 Comparison of RNI Questions Involving Spanning Trees

A two-factor, ANOVA experiment without replication was performed to determine if differences existed between different outcomes or if differences existed between different RNI question styles for the questions performed on spanning trees. The results appear in Table 7.37.

Using the P-values, it can be easily determined that there are significant differences in the means for the different outcomes. However, there is not enough evidence to reject H_0 for the different question styles, either; in fact, the P-value corresponding to the question styles is significantly large, which then makes one wonder if H_0 could be "accepted almost to complete certainty".

However, it should be noted, as was done previously, that students were very unwilling to attempt RNI questions in graph theory, even during the invigilated test session, when marks were being allocated to such work. In order to better determine the effectiveness of this question and the implementation of the various question styles within it, more students will need to be tested and to do so requires these questions to be forced upon them in future assessments so that they can no longer avoid having to perform calculations in graph theory.

Source	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Outcomes	3931.75	5	786.35	69.69129	0.000127	5.050329
Question Styles	14.08333	1	14.08333	1.248154	0.314694	6.607891
Error	56.41667	5	11.28333			
Total	4002.25	11				

Table 7.37 ANOVA two-factor (without replication) Table for the RNI practice and test questions involving spanning trees and using Microsoft Excel.

7.6.7 Comparison of MC Questions Involving Spanning Trees

A two-factor, ANOVA experiment without replication was performed to determine if differences existed between different outcomes or if differences existed between different question styles for the questions performed on planar graphs. The results appear in Table 7.38.

Using the P-values, it can be easily determined that there are significant differences in the means for the different outcomes. Also, at the $\alpha = 5\%$ level of significance, there is also enough evidence to reject H_0 for the different question styles; in fact, the evidence is stronger for the question styles than it is for the outcomes. Therefore, there is also a significant effect on the question styles used.

Since there is a significant effect on the question styles, it is important to determine for which question styles these differences appeared. To do this, a series of t-tests are performed. The results for all t-tests, along with descriptions of all question types and scenarios used, appear in Appendix B.

Source	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Outcomes	513.0875	7	73.29821	6.492153	9.02×10^{-6}	2.158829
Question Styles	693.6125	9	77.06806	6.826055	8.86×10^{-7}	2.032242
Error	711.2875	63	11.29028			
Total	1917.988	79				

Table 7.38 ANOVA two-factor (without replication) Table for the MC practice and test questions involving spanning trees and using Microsoft Excel.

This experiment involves forty-five comparisons. Therefore, $EER = 1 - 0.95^{45} \approx 0.90$, which implies there is approximately a 90% probability of committing a Type I error in this set of experiments. Also, this set of experiments caused rejections of H_0 for the following comparisons:

- Visual practice set vs. all test question sets and scenarios
- Logical / mathematical practice set vs. all test question sets and scenarios
- Test question given adjacency matrices as options and a graph as part of the question with the university student services scenario vs. test question given matrices as options and an adjacency matrix as part of the question with the business departments scenario
- Test question given graphs as options and an adjacency matrix as part of the question with the business departments scenario vs. test

question given graphs as options and an adjacency matrix as part of the question with the university student services scenario

It is interesting to note that the rejections of H_0 occur only for comparisons involving the practice question sets with any of the test question sets. Also, for all such cases, both the one-tailed and two-tailed tests failed and it may be concluded that $H_1 : \mu_1 > \mu_2$ for $\alpha = 5\%$. However, it is worth noting that the two-tailed test only barely failed for the practice question set vs. the test question given adjacency matrices as options and a graph as part of the question with the business departments scenario. With an EER of over 90%, it may be possible to suggest that any error occurring in this experiment may likely come from this particular comparison. For either practice question set, we can conclude that students found it difficult to study the material in this topic when graphs or adjacency matrices appeared in the questions.

The test question that had graphs as options and an adjacency matrix as part of the question with the business departments scenario produced some interesting results of its own as the null hypothesis is rejected for both test questions involving the university student services scenario. In both cases, the experiment has shown that the business departments scenario, using graphs as options and an adjacency matrix as part of the question, produced greater results than either of the test questions using the university student services scenario. It is difficult to understand why this is happening and so, it is also likely that if a Type I error is occurring in this experiment, then it could be suggested that it is coming from these comparisons, especially as these are the only instances of rejected null hypotheses between test questions for the entire experiment. However, if this is not a result of a Type I error occurring, then more research will be needed to determine why this is happening.

Chapter 8 Analysis of Test Scores for Introductory Graph Theory

8.1 Hypotheses

The assessments from 2008 – 2014 were conducted in practice environments with foundation year students at Brunel University. The setup for all questions was that questions would either involve questions that could contain any combination of graphs and adjacency matrices; some questions included both elements, whereas other questions included only one of the elements. All questions were either assessed with scores of 1 (correct) or 0 (incorrect). Although this assessment strategy will have an impact on some hypothesis testing, comparisons of facilities and discriminations should not vary significantly. For hypothesis testing, determining differences in proportions of results has been used since the scoring system is binary. Also, because there was a significant change in topics assessed from 2011 – 2014, discrimination values cannot be carried through in an overall comparison from 2008 – 2014 and so, two separate analyses have been conducted.

Looking at the facility and discrimination values, any comparisons between academic years are expected to remain consistent, assuming that in-class teaching has been consistent and the syllabus has not changed. Therefore, it is being hypothesized that facility and discrimination values will not be statistically different between academic years. Also, it is not expected that there will be a significant difference in question types between questions. However, it is expected that some question topics will be easier for students to answer than others. Therefore, it is being hypothesized that there will be a significant difference in values between topics. Furthermore, it is expected that for some topics, it is more advantageous to use either graphs or adjacency matrices, so it is being hypothesized that there will be some significant differences between questions involving graphs and questions involving adjacency matrices.

8.2 2008 – 2011 Analysis

8.2.1 Summary Results

Spearman rank correlation is useful to determine the relation between values when there may be some degree of inter-dependence between two sets. This is useful for the analysis of these questions as overall assessment scores between academic year groups with different students may vary.

The Spearman rank correlations for the facility values from the 2008 – 2011 data are as shown in Table 8.1:

Spearman Rank Correlations	**2009 - 2010**	**2010 - 2011**
2008 - 2009	**0.8858**	**0.7438**
2009 - 2010		**0.7446**

Table 8.1 Spearman rank correlations for 2008 – 2011.

These strong, positive correlations suggest there is significant improvement in the results for each question from one academic year to the next academic year; the tables in Appendix C further show improvement through the difficulties of respective questions through each academic year and overall for these years.

Overall difficulties for the question set ranged from 46.21% to 54.75% for each academic year, with an overall facility of 50.35% for the three academic years. In addition, none of the questions presented had a negative discrimination value; discrimination values ranged from 0.195 (2009 – 2010) to 0.747 (2010 – 2011). Overall difficulties for each question ranged from 23.45% to 71.58% between the three academic years.

The comparison of facility and discrimination values from 2008 to 2011, as shown in Appendix C, show that, generally speaking, as the facility value increases, the discrimination values appear to begin converging between 0.5 and 0.6; this especially appears to be the case for questions asked after the 2008 – 2009 academic year. This is encouraging as it is showing that questions which are easier for students to answer correctly are not resulting in lower discrimination values.

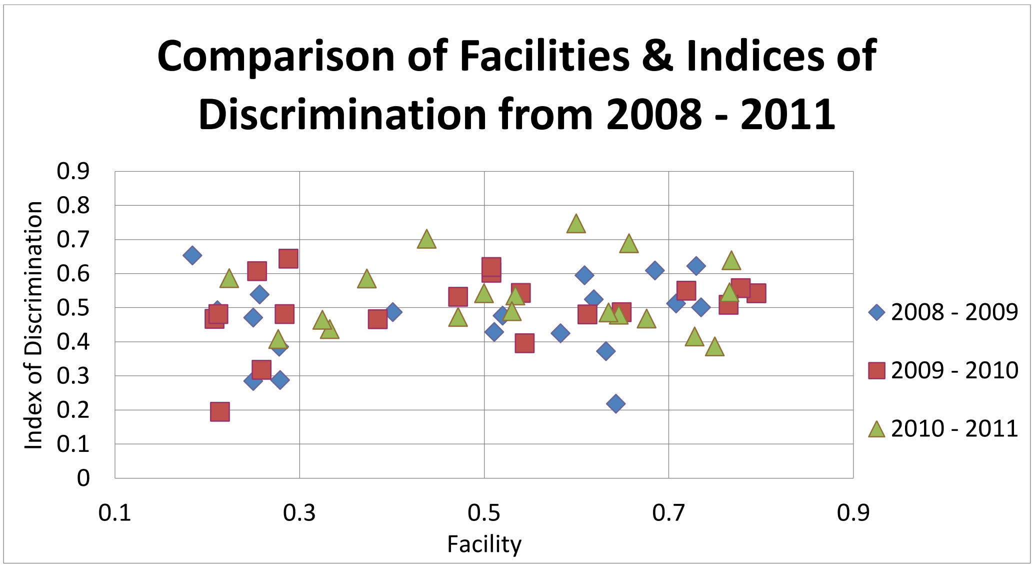


Figure 8.1 Scatter diagram of index of discrimination versus facility for questions given to students from 2008 to 2011 in MA0422.

The comparison of the question topics, as shown in Figure 8.2, shows that all of the bipartite graph questions were more difficult for students to answer correctly than almost all other questions, which the exceptions of some questions on degree and edge sets. Bipartite graphs do involve some prerequisite knowledge about graphs, which can include learning material based on all other topics noted in this question set (except for shortest path problems); it is worth noting that although the shortest path problems appear to have been answered more correctly by students, there were only three specific problems in the set and did not necessarily refer to specific algorithms; Kruskal’s and Prim’s algorithms, which typically refer to shortest path problems, will appear in the 2011 – 2014 question set.

The comparison of facility and discrimination values, based on question type, is provided in Figure 8.3. Numerical input questions, which varied largely in facility, were generally consistent in terms of discrimination values. Multiple-choice questions appear to vary greatly throughout the question set. Word input questions had reasonably high discrimination values, but those that had displayed pop-up windows, asking students to double-check the formatting of their answers, appear to have significantly larger facility values, suggesting the pop-up windows were helpful in increasing the number of correct answers made by students.

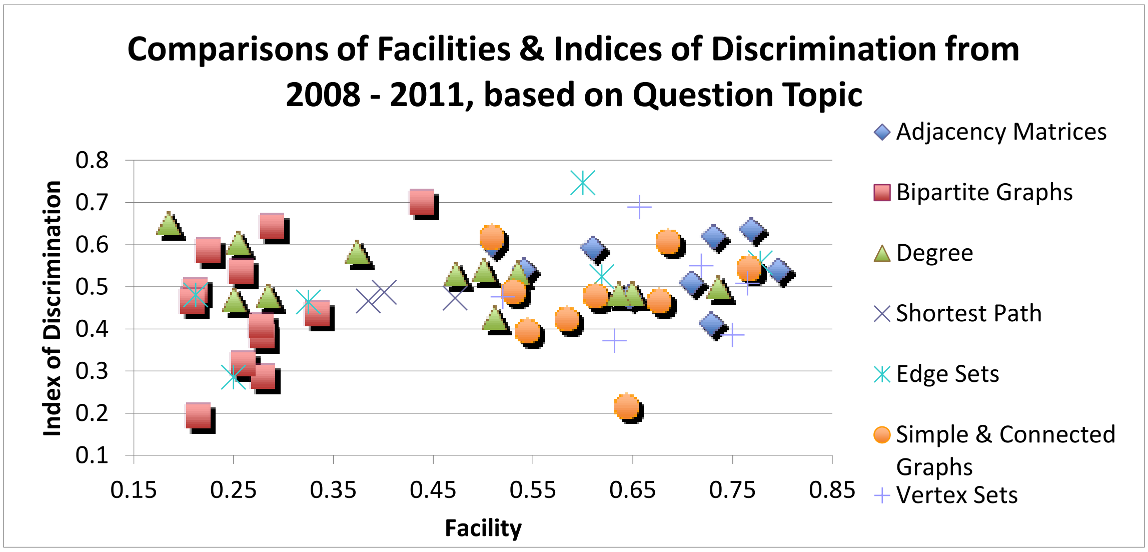


Figure 8.2 Scatter diagram of index of discrimination versus facility for questions given to students from 2008 to 2011, based on question topic.

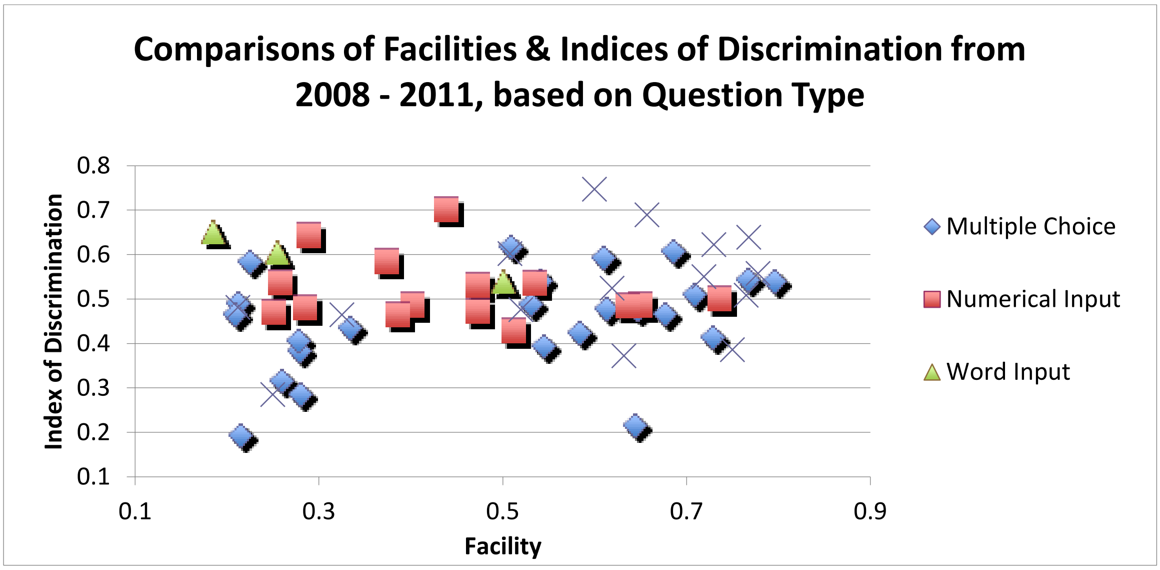


Figure 8.3 Scatter diagram of index of discrimination versus facility for questions given to students from 2008 to 2011, based on question type. Xs in the chart refer to Word Input questions that included pop-up checks as part of the question design.

Wilcoxon signed rank tests were conducted at $\alpha = 0.05$ with the null hypothesis stating the medians of values (either facility or discrimination values) between academic years are equal. There were 19 questions in each test set, the critical value for a two-tailed test is 46. The test statistics calculated are shown in Table 8.2.

It is important to note that the null hypothesis was rejected at the 5% level of significance for the two-tailed test for both tests involving the 2010 – 2011 question set. The overall mean for facility of the 2010 – 2011 questions is higher than that of the other academic years, implying that students performed better than expected with the same questions.

Facility	**2009 - 2010**	**2010 - 2011**	**Discrimination**	**2009 - 2010**	**2010 - 2011**
2008 - 2009	86	41	**2008 - 2009**	72	55
2009 - 2010		39	**2009 - 2010**		65

Table 8.2 Test statistics for facility and discrimination values using the Wilcoxon signed rank test for the 2008 – 2011 question set. Results highlighted in red (with white text) refer to tests where the null hypothesis, i.e. the medians of the corresponding values (i.e. either facility values or discrimination values) are equal, was rejected at $\boldsymbol{\alpha = 0.05}$.

8.2.2 Question Analysis

8.2.2.1 Bipartite Graphs

Three of the four most difficult questions for students to answer were bipartite graph questions, all of which were multiple-choice (MC) questions. The numerical input (NI) question was answered better by students, but still appears to have a considerably low facility.

The first three questions are posing the same problem, i.e. which of the following is bipartite, but what changes is the use of graphs and adjacency matrices. Many characteristics of graphs are easier to see from a graph than they are to determine from an adjacency matrix. However, for the topic of bipartite graphs, it can be argued that it is equally likely to find two partite sets as the two sets can easily appear in a symmetric matrix, as shown in Example 8.1.

Although a difference in facility values exists, it is relatively small and furthermore, the ranking of facility generally changes from one academic year to the next, as shown in Table 3. Therefore, it cannot be concluded that showing graphs and adjacency matrices has any effect on the facility of determining if a graph is bipartite.

$$\begin{bmatrix} & A & B & C & D & E \\ A & 0 & 0 & 1 & 1 & 0 \\ B & 0 & 0 & 0 & 1 & 1 \\ C & 1 & 0 & 0 & 0 & 0 \\ D & 1 & 1 & 0 & 0 & 0 \\ E & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

Example 8.1 A symmetric matrix representing a bipartite graph with sets, $\{\boldsymbol{A}, \boldsymbol{B}\}$ and $\{\boldsymbol{C}, \boldsymbol{D}, \boldsymbol{E}\}$.

In comparison, the NI question already tells the student that the graph in question is bipartite. Therefore, less effort is required as some additional information about the graph being viewed is provided. Upon determining the number of vertices in each set, the student simply has to choose the required number of vertices (i.e. largest or smallest, depending on the question) and type this value in to answer the question correctly.

Although the NI question was not answered well in the 2008 – 2009 academic year, it performed better than the MC questions for the other two academic years and with increasing significance. Knowing that the graph in question is bipartite does reduce the effort involved in answering the question, and especially as there is only one graph involved instead of four graphs or adjacency matrices, this question could reasonably be answered quicker than the MC questions.

The discrimination values for these questions are all positive. However, they vary significantly from 0.195 (2009 – 2010) to 0.702 (2010 – 2011). Although they each provided some positive effect to students' overall assessment scores, they are not doing so with reasonably consistent values of discrimination; for instance, the discrimination value for the bipartite *graph* search MC question in 2009 – 2010 was 0.195, whereas the discrimination value for the bipartite *adjacency matrix* search MC question in 2010 – 2011 was 0.586.

8.2.2.2 Edge Set

The question asking to input the edges of a given graph was challenging, but the similar question, which asked to input the edges of a given digraph, was considerably easier. The overall facility for the question involving a typical graph was 0.2688, whereas the overall facility for the question involving a digraph was 0.6667. The correlations for the question involving a typical graph were also considerably lower than those for the digraph, suggesting that the question involving the digraph resulted in an improvement in students' overall assessment scores.

With regards to these two questions, there were very specific guidelines for how to input answers. For any graph, $\overline{AB}$ and $\overline{BA}$ represent the same edge, so students were reminded to input their answers in alphabetical order. However, for digraphs, this was not necessary as $\overrightarrow{AB}$ and $\overrightarrow{BA}$ are two different edges. Therefore, although digraphs normally require some prerequisite understanding of how graphs function, answering the question using standard graphs requires the additional step of alphabetically listing all edges. For the entire list of edges, this, too, had to be in alphabetical order, but this was required for both questions.

8.2.2.3 Indegree and Outdegree

The question on indegree and outdegree did not perform as well as the question asking for the degree of a typical graph. The question about indegree and outdegree looks specifically at digraphs as the directionality of edges determines unique values for these two degree values. For the question looking at indegree and outdegree, the overall facility was 0.3117, but it is also worth noting that this is based on continually increasing difficulties from one academic year to the next. Also, for the 2008 – 2009 academic year, this question was attempted only 28 times, as opposed to 67 and 59 times in 2009 – 2010 and 2010 – 2011 respectively. The similar question, asking to find the degree of a vertex of a typical graph, had an overall facility of 0.5063, but the difficulties for each academic year were somewhat comparable and did not continue to increase from one academic year to the next. For both questions, discrimination values were similar and were contained within a small range, with a minimum of 0.429 and a maximum of 0.585.

8.2.2.4 Degree Sequences

The question asking to generate the degree sequence resulted in better facility values than the question on indegree or outdegree, but not as well as the question asking for the degree of a vertex of a typical graph. This was a Responsive Word Input (RWI) question, where students would normally receive a pop-up window upon answering the question, asking them to double check their responses before clicking to submit a second time. This question only looked at typical graphs, so there was less prerequisite knowledge required than indegree and outdegree, but there are more steps required in answering this particular question as repeated calculations of degree are required and the answer has to be given in a particular format, matching all degrees in the numerical order of their corresponding vertices.

As noted earlier, this question was not answered well in 2008 – 2009, resulting in the lowest facility value for any question. However, facility values increase significantly in later years, resulting in an increase of facility of 0.316 from the 2008 – 2009 academic year to the 2010 – 2011 academic year. It is also important to note that the number of attempts of this question also increased significantly from 2008 – 2011 as attempts more than doubled from the 2008 – 2009 academic year (38) to the 2010 – 2011 academic year (78).

8.2.2.5 Shortest Distance Problems

The RNI question about the shortest distance between two towns was answered numerous times in each academic year with more than double the number of attempts than any other question for respective academic years. Facility values were generally consistent, remaining between 0.385 (2009 – 2010) and 0.472 (2010 – 2011). Discrimination values were generally consistent, remaining between 0.466 (2009 – 2010) and 0.487 (2008 – 2009), but there are also significantly more attempts made from one academic year to the next, reaching from 142 (2008 – 2009) to 192 (2009 – 2010) and then 214 (2010 – 2011). This question is a stand-alone question as there were no other questions in the assessment relating to the distance between two vertices using weighted graphs.

8.2.2.6 Simple and Connected Graphs

There were three questions asking to find a simple and connected graph amongst a list of possibilities. In a similar format to the bipartite graph MC questions, these three MC questions differ in terms of what is displayed, i.e. adjacency matrices, graphs, or a combination of both. A comparison of the results is provided in Table 8.3.

QUESTION	2008 – 2009		2009 – 2010		2010 – 2011		OVERALL
Fac. = Facility Dis. = Discrimination	Fac.	Dis.	Fac.	Dis.	Fac.	Dis.	Facility
Find the simple connected graph given the adjacency matrices; RandMC	0.685	0.609	0.544	0.396	0.53	0.49	0.5727
Find the simple connected graph given the graphs or adjacency matrices; RandMC	0.583	0.425	0.508	0.619	0.676	0.468	0.5926
Find the simple connected graph given the graphs; RandMC	0.643	0.218	0.612	0.48	0.766	0.545	0.6897

Table 8.3 Comparison of MC questions asking to find a simple and connected graph.

Unlike the question on bipartite graphs, visualisation in this question is very important as it should be easier to see a graph not containing loops or being disconnected rather than determining it through an adjacency matrix. From these results, this appears to generally be the case, although for the 2008 – 2009 academic year, students performed slightly better with the adjacency matrices questions. Discrimination values vary, but for 2008 – 2009 and 2009 – 2010, the discrimination values for the graphs questions is lower than at least one of the other respective discrimination values for each academic year, whereas in 2010 – 2011, the discrimination value for the graphs questions is higher than the other two discrimination values; however, for 2010 – 2011, discrimination values do not vary as much as they do in the other academic years.

8.2.2.7 Adjacency Matrices

The next two questions ask to match a graph to a corresponding adjacency matrix or vice versa. The question asking to find a matching adjacency matrix proved to be only somewhat more difficult, with an overall facility of 0.6311 compared to 0.7150 for the question asking to find a matching graph.

QUESTION	2008 – 2009			2009 - 2010			2010 - 2011		
# = Number of attempts Fac. = Facility Dis. = Discrimination	#	Fac.	Dis.	#	Fac.	Dis.	#	Fac.	Dis.
Given graph, find matching adjacency matrix; MC	46	0.609	0.595	87	0.54	0.543	47	0.728	0.415
Given adjacency matrix, find matching graph; MC	48	0.708	0.512	73	0.795	0.542	58	0.646	0.479

Table 8.4 Comparison of MC questions asking to match a graph to an adjacency matrix or vice versa.

The lowest and highest facility values both appear in the 2009 – 2010 academic year, with 0.54 for finding the matching adjacency matrix and 0.795 for finding the matching graph. In 2008 – 2009, although the question asking to find the matching graph has a higher mean facility value, the difference between it and the corresponding facility value for finding the matching adjacency matrix is significantly less than that from 2009 – 2010; furthermore, facility values swap rank in 2010 – 2011, showing a higher facility value for finding the matching adjacency matrix. There are also significantly more attempts made in 2009 – 2010 in comparison to any of the other academic years, which may have some representative effect on the presented facility values. Also, although not much less than some other facility values, the facility value of 0.54 for finding the matching adjacency matrix (2009 – 2010) does appear to be an anomaly in some way in that this question had the most attempts made, yet has a discrimination value consistent with many other discrimination values present. Discrimination values throughout are generally consistent, ranging from 0.512 (2008 – 2009) to 0.728 (2010 – 2011).

The last question looks at an adjacency matrix for a given graph, asking students to input the location of an error in the matrix. This question is similar to the MC questions, but is an RWI question with a check so that students verify that the formatting of their answers matches the required format for assessing their answers objectively. Students are aware in the question that there is a fault with the corresponding adjacency matrix, so the question asks them to find it.

This question performed generally well, although there is an anomaly in the facility value for 2009 – 2010 (0.508), compared to 2008 – 2008 (0.73) and

2010 – 2011 (0.768). Discrimination values are consistent with a range of 0.037 between all three academic year groups. The MC question asking to find the matching graph, given the adjacency matrix, had a higher overall facility, with a difference of 0.0336. However, the facility values for 2008 – 2009 and 2010 – 2011 are higher for the RWI question; the reason for the MC question having a higher facility value is due to the anomaly in 2009 – 2010, which caused a significant change of rank between the difficulties of the two questions.

8.2.2.8 Vertex Sets

The next two questions ask to look at a given graph and to input the vertices of the graph. The difference between the questions is that there are disconnected vertices in one of the graphs with no edges connecting it to other vertices. The objective of these questions was to determine if students would not input the disconnected vertices in their answers, even though they belong to the graphs. The results are shown in Table 8.5.

The overall facility for the question that had disconnected vertices was 0.6596 and the overall facility for the question without disconnected vertices was 0.7158. Generally speaking, the question without disconnected vertices achieved better facility values, but similar to the question on simple and connected graphs, there is an issue with the 2009 – 2010 academic year; in this case, the question with disconnected vertices had a larger facility. The difference in facility values is not necessarily significant as it is only in the 2008 – 2009 academic year that the facility values differ by more than 0.100. The question without disconnected vertices had the highest overall facility of any question overall for 2008 – 2011.

QUESTION	2008 – 2009			2009 – 2010			2010 – 2011		
# = Number of attempts Fac. = Facility Dis. = Discrimination	#	Fac.	Dis.	#	Fac.	Dis.	#	Fac.	Dis.
Given graph, input vertices (with disconnected vertices); WI+check	25	0.52	0.476	13	0.765	0.508	26	0.657	0.689
Given graph, input vertices; WI+check	19	0.632	0.372	12	0.719	0.55	23	0.75	0.386

Table 8.5 Comparison of WI + Check questions asking to input the vertices of a given graph.

The discrimination values for these two questions generally show that the question with disconnected vertices had a stronger effect on students' overall assessment scores than the question without disconnected vertices. With the exception of 2009 – 2010, the discrimination values for the question with disconnected vertices is either 0.104 (2008 – 2009) or 0.303 (2010 – 2011) higher than the discrimination values for the question without disconnected vertices in respective academic years. The discrimination values for 2009 – 2010 are similar, with a difference of 0.042.

The question asking to calculate the sum of entries of an adjacency matrix had good results. Its facility values have a range of 0.100, suggesting some consistency, and discrimination values are more significantly consistent, having a range of only 0.015. A continuously increasing number of students attempted this question from one academic year to the next, going from 49 (2008 – 2009) to 74 (2010 – 2011). Exactly $\frac{2}{3}$ of the students answered this question correctly from 2008 – 2011 and it is the question related to degree that has the best overall facility; the question related to degree with the second best overall facility is the NI question asking to determine the degree of a vertex of a graph (0.5063).

8.2.3 Hypothesis Testing

8.2.3.1 Test for Difference in Proportions Within Topics

Hypothesis testing was carried out on comparable questions within each topic. The results are shown in Table 8.6.

The analysis shows that students found it easier to answer adjacency matrix questions when the adjacency matrix was given and they had to determine the matching graph. Students found that it was easier to answer the question on edge sets that involved the digraph rather than the graph.

The simple and connected graphs questions showed that the use of graphs made it more difficult for students to answer correctly than when adjacency matrices were used, but the use of graphs made questions easier to answer than those questions that had both graphs and adjacency matrices included. However, the null hypothesis could not be rejected for the difference in proportions between adjacency matrices and the combination of graphs and adjacency matrices, which appears to contradict the logic of the other two hypothesis tests. Therefore, it is possible that a Type I error may be involved, even with $EER = 0.0975$.

Topic	Issue	Z
Adjacency Matrices	Given graph vs. Given adjacency matrix	-1.8366
Bipartite Graphs	Adjacency Matrix vs. Graph	-0.4825
Bipartite Graphs	Adjacency Matrix vs. Combination	-0.2179
Bipartite Graphs	Adjacency Matrix vs. Combination	-0.7261
Degree	No available questions	
Shortest Distance	No available questions	
Edge Sets	Graph vs. Digraph	-5.2083
Simple & Connected Graphs	Graphs vs. Adjacency Matrices	-2.5051
Simple & Connected Graphs	Adjacency Matrices vs. Combination	-0.4098
Simple & Connected Graphs	Graphs vs. Combination	2.0041
Vertex Sets		0.8339

Table 8.6 List of $\boldsymbol{Z_{test}}$ values for hypothesis testing of questions within topics for questions tested from 2008 - 2011. Values highlighted in red show a rejection of the one-tailed test in favour of $\boldsymbol{H_1}: \boldsymbol{\mu_1^\alpha} < \boldsymbol{\mu_2^\alpha}$. Values highlighted in blue show a rejection of the one-tailed test in favour of $\boldsymbol{H_1}: \boldsymbol{\mu_1^\alpha} > \boldsymbol{\mu_2^\alpha}$.

8.2.3.2 Test for Difference in Proportions Between Topics

Test values for the comparisons between topics (using comparable questions) is shown in Table 8.7. In this table, it is evident that the adjacency matrix and vertex sets questions were significantly easier to answer than all questions, except questions on vertex sets, for which no definite conclusion can be made. Questions on bipartite graphs were significantly more difficult for students to answer than any other topic; this was an expected result because there is some prerequisite knowledge about graphs required in order to answer questions on this topic. Questions on shortest distance were the next most difficult for students to answer correctly, with bipartite graphs questions being more difficult and questions on degree not being significantly different enough to draw a conclusion, but questions on degree were comparatively difficult to all other topics. Questions on edge sets were easier for students to answer correctly, although no conclusion could be drawn when compared to questions on simple & connected graphs.

Topic 2 → / Topic 1 ↓	Bipartite Graphs	Degree	Shortest Distance	Edge Sets	Simple & Connected Graphs	Vertex Sets
Adjacency Matrices	13.8088	7.6523	8.6385	1.8427	2.1037	-0.3544
Bipartite		-6.7465	-5.1251	-6.8248	-11.8225	-10.1000
Degree			1.3928	-2.6513	-5.5290	-5.4521
Shortest Distance				-3.4372	-6.6161	-6.2714
Edge Sets					-0.5759	-1.7978
Simple & Connected Graphs						-1.7659

Table 8.7 List of Z_{test} values for hypothesis testing of questions between topics for questions tested between 2008 - 2011. Values highlighted in red show a rejection of the one-tailed test in favour of $H_1: \mu_1^\alpha < \mu_2^\alpha$. Values highlighted in blue show a rejection of the one-tailed test in favour of $H_1: \mu_1^\alpha > \mu_2^\alpha$.

8.2.3.3 Test for Difference in Proportions Between Question Types

Test values for the comparisons between question types for the entire data set are given in Table 8.8.

Word input questions with checks included (WI + Check) were easier for students to answer correctly than all other questions; these questions, however, the topics for these questions were adjacency matrices, edge sets, and vertex sets, which were the three question topics that had significantly higher facility values, as was shown in Table 8.7. Nonetheless, these questions had significantly better facilities than the (Responsive) Numeric/Word Input (NI/RNI; WI/RWI) questions, which could partially result from asking students to ensure their answers were written in the correct format.

Type 2 → / Type 1 ↓	NI/RNI	WI/RWI	WI+Check
MC	4.1679	4.4585	-3.5205
NI/RNI		2.4231	-6.5692
WI/RWI			-6.1546

Table 8.8 List of Z_{test} values for hypothesis testing of questions between question types for questions tested between 2008 - 2011. Values highlighted in red show a rejection of the one-tailed test in favour of $H_1: \mu_1^\alpha < \mu_2^\alpha$. Values highlighted in blue show a rejection of the one-tailed test in favour of $H_1: \mu_1^\alpha > \mu_2^\alpha$.

Multiple-choice (MC) questions were significantly easier to answer correctly; the topics for these questions were simple & connected graphs, adjacency matrices, and bipartite graphs. The topic facilities varied when compared to each other, so it is possible that question type has played a significant role.

8.3 2011 – 2014 Analysis

8.3.1 Summary Results

The Spearman rank correlations for the 2011 – 2014 data are as shown in Table 8.9.

These strong, positive correlations suggest there is significant consistency in the results for each question from one academic year to the next academic year; the tables in Appendix C further show question consistency through the difficulties of respective questions through each academic year and overall for the three academic years.

Overall difficulties for the question set ranged from 47.35% to 57.33% for each academic year, with an overall facility of 53.72% for the three academic years. In addition, none of the questions presented had a negative discrimination value; discrimination values ranged from 0.029 (2012 - 2013) to 0.824 (2011 – 2012). Overall difficulties for each question ranged from 18.92% to 83.87% between the three academic years.

Spearman Rank Correlations	**2012 – 2013**	**2013 – 2014**
2011 – 2012	**0.8060**	**0.7044**
2012 – 2013		**0.8596**

Table 8.9 Spearman rank correlations for 2011 – 2014.

The overall difficulties for the question set in each academic year from 2011 - 2014 are consistent with the overall difficulties for the question set in each academic year from 2008 – 2011. However, there is a larger range of overall difficulties for each question, increasing from 0.4813 to 0.6495. Furthermore, there are some worrying discrimination values, with some values reaching as low as 0.029.

The numbers of attempted questions between the two sets is comparable, with 3,412 total attempts made between 2008 – 2011 and 3,358 total attempts made between 2011 – 2014.

One major change in this question set is the replacement of questions relating to bipartite graphs for questions relating to minimum spanning tree algorithms, namely Prim's algorithm and Kruskal's algorithm. There were four questions relating to bipartite graphs, but there are twelve questions relating to minimum spanning trees, with six questions dedicated to each of the two named algorithms.

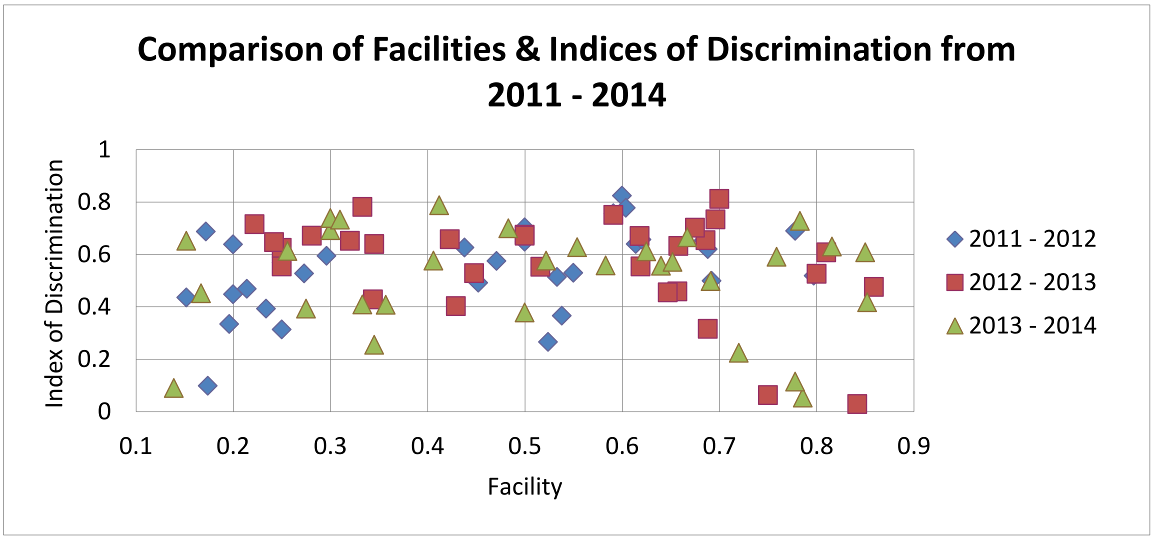


Figure 8.4 Scatter diagram of index of discrimination versus facility for questions given to students from 2008 to 2011 in MA0422.

Also interesting to note is that the correlation between overall facility values for each set correlate positively with the number of questions attempted in each academic year; this differs from the 2008 – 2011 assessments, where an increase in the number of attempted questions did not necessarily provide a higher facility. This could suggest students were generally able to learn from their mistakes in previous attempts in order to perform better, at least in comparison to the 2008 – 2011 data.

Figure 8.4 shows the facility and discrimination values for the questions presented from 2011 to 2014. Questions that have facility values ranging from approximately 0.3 to 0.7 have more consistent discrimination values than those which have other discrimination values. There are some questions with lower facility and discrimination values and some questions with very high facility values, but significantly low discrimination values; these lower discrimination values are concerning as the questions are less useful in the learning and assessment taking place within the module.

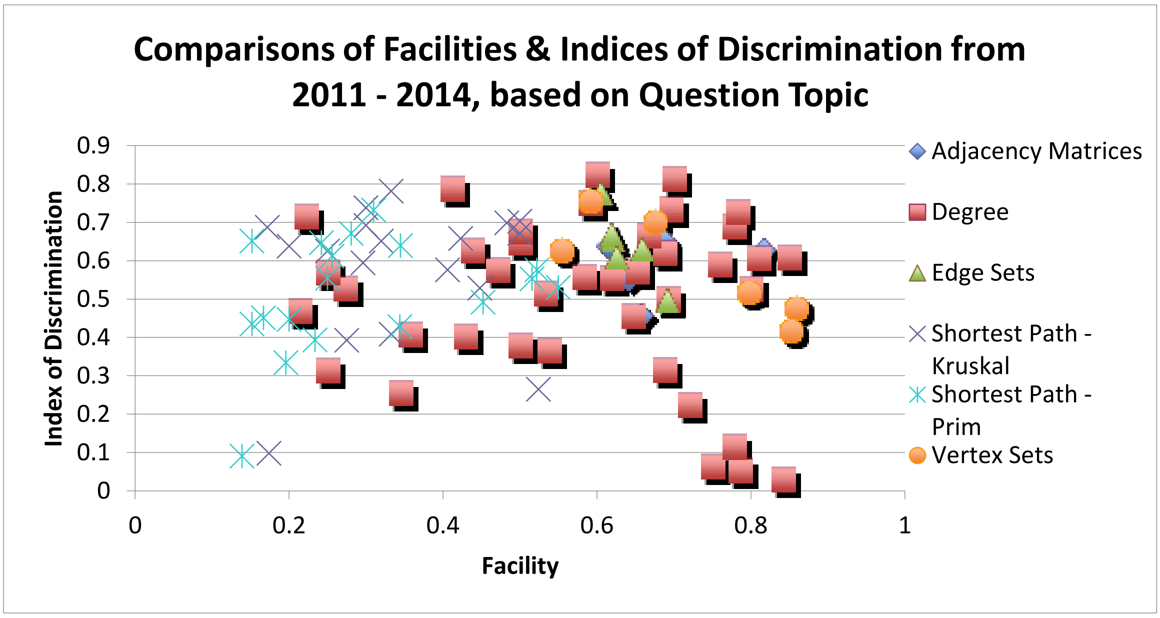


Figure 8.5 Scatter diagram of index of discrimination versus facility for questions given to students from 2008 to 2011, based on question topic.

Figure 8.5 shows the facility and discrimination values based on question topics. The questions on adjacency matrices were answered well, with consistently high facility and discrimination values. Questions on degree varied significantly and the topics within the category of degree which had larger facility values and lower discrimination values varied also between outdegree and degree sequence questions. Questions on edge and vertex sets were consistent in facility and discrimination values. The shortest path problems vary significantly in discrimination values, but were generally more difficult for students to answer than the other topics; similar to the topic of bipartite graphs, this is most likely due to the fact that prerequisite knowledge is necessary in understanding how the algorithms work, but in the case of algorithms and due to the marking scheme set up for these questions, one mistake in the algorithmic process would have likely resulted in an incorrect solution and thus, a mark of zero awarded for the entire question. In comparison to the shortest path problems, the most significant difference is in the discrimination values as they are less consistent than the 2008 – 2011 questions on shortest path problems. It is worth noting that all shortest path problems created from 2011 – 2014 were Word Input (WI) questions without checks.

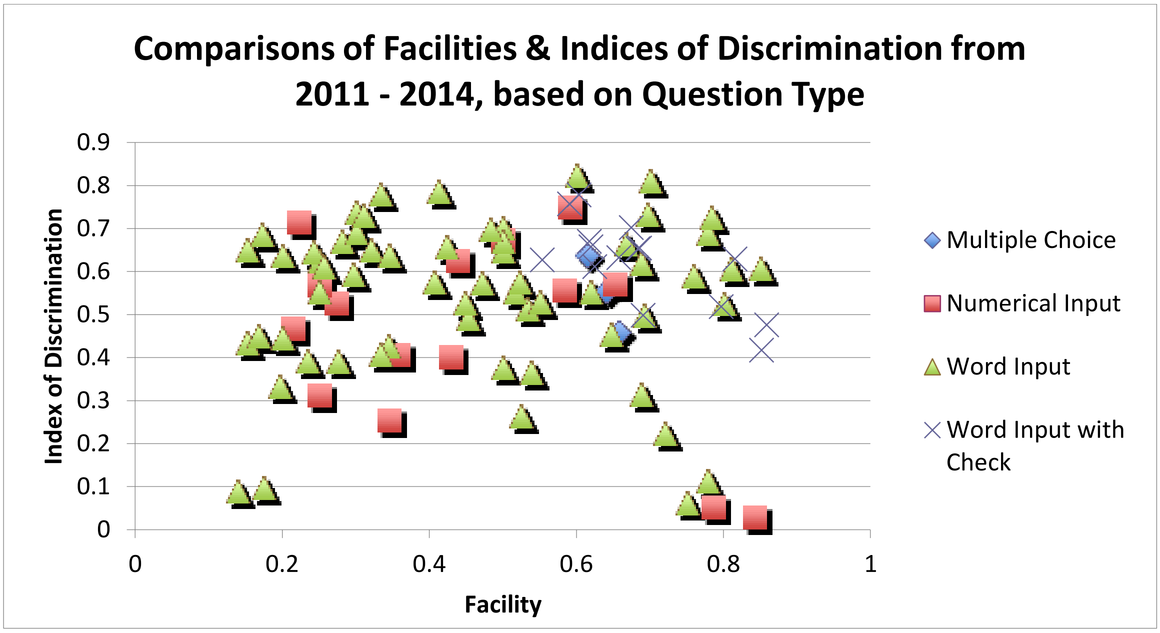


Figure 8.6 Scatter diagram of index of discrimination versus facility for questions given to students from 2008 to 2011, based on question type.

The comparison of facility and discrimination values based on question type, is shown in Figure 8.6. There was just one multiple-choice question about adjacency matrices. Numerical input questions generally held facility values from 0.2 to 0.7 and discrimination values between 0.25 and 0.75, with the exception of the question on outdegree, which had higher facility values, but significantly lower discrimination values. All of the Prim's and Kruskal's algorithms questions were WI questions, along with various other questions on degree sequences. In comparison, just like the 2008 – 2011 questions showed, those questions with pop-up windows asking students to double check the formatting of their answers resulted in higher facility values; these questions also had less variation of discrimination values.

8.3.2 Question Analysis

8.3.2.1 Minimum Spanning Trees

The first set of questions explores two specific algorithms for determining the minimum spanning tree of a graph, i.e. Prim's Algorithm and Kruskal's Algorithm. These questions were designed by Zaczek[44] and used as part of the new online practice assessment, reflecting upon an addition to the syllabus that took place in 2011. There are 12 questions in total on these two algorithms.

Table 8.10 shows the analysis of these specific questions.

These questions generally appear with the lowest overall difficulties for all questions asked between 2011 – 2014; the only exceptions are questions relating to indegree, outdegree, and degree of a symmetric matrix. However, the overall facility values is significantly varied with a range of 0.3407. The ranking of facility values is provided in Table 12.

There are some varied numbers of attempts with these questions, although not necessarily significant as there is a large selection of questions from the same topics. It does not appear that one algorithm was found to be more challenging than another as the facility values are higher for Kruskal's algorithm at various times and are higher for Prim's algorithm at other times for comparable questions within the set. Also, it does not appear that the number of vertices had any particular effect on student results as graphs using 7 vertices appear to have varied rankings of facility values throughout the set, compared to graphs using 5 or 6 vertices.

There is clearly a variation in the correlations for the 2013 – 2014 academic year in relation to the other academic year groups; however, it does correlate well with the overall facility rankings for 2011 – 2014, as did the other academic year groups. Although the correlations are all positive, it is clear from the original data the questions do not generally rank in the same position during each academic year.

The questions asking for the minimum spanning tree of a graph generally had the lowest facility, although this was mixed with the questions asking about a particular edge within the algorithmic process. Facility values for these questions are not affected either by the type of algorithm used or the number of vertices involved. Facility values significantly improve for all four questions from the 2011 – 2012 academic year to the 2012 – 2013 academic year and are somewhat consistent between the 2012 – 2013 and 2013 – 2014 academic years, although some variation does exist for the questions whose graphs have 7 vertices. Discrimination values are significantly positive throughout, ranging from 0.408 (2013 – 2014) to 0.781 (2012 – 2013).

Question description	2011 - 2012		2012 - 2013		2013 - 2014		OVERALL
Fac. = Facility Dis. = Discrimination	Fac.	Dis.	Fac.	Dis.	Fac.	Dis.	Facility
what is the minimum spanning tree_5-6 vertices_Kruskal; WI	0.172	0.687	0.333	0.781	0.333	0.408	0.2885
what is the minimum spanning tree_7 vertices_Kruskal; WI	0.2	0.638	0.25	0.625	0.406	0.576	0.2963
was AB edge added/rejected/not considered and at what step_5-6 vertices_Kruskal; WI	0.296	0.594	0.423	0.658	0.275	0.392	0.3226
was AB edge added/rejected/not considered and at what step_7 vertices_Kruskal; WI	0.174	0.098	0.32	0.651	0.483	0.699	0.3377
which is the n'th edge of the minimum spanning tree_7 vertices_Kruskal; WI	0.5	0.703	0.448	0.528	0.3	0.692	0.4286
which is the n'th edge of the minimum spanning tree_5-6 vertices_Kruskal; WI	0.524	0.265	0.5	0.671	0.3	0.738	0.4478
what is the minimum spanning tree_7 vertices_Prim; WI	0.152	0.435	0.281	0.671	0.152	0.651	0.1892
was AB edge added/rejected/not considered and at what step_7 vertices_Prim's algorithm; WI	0.196	0.334	0.242	0.647	0.139	0.09	0.1913
was AB edge added/rejected/not considered and at what step_5-6 vertices_Prim's algorithm; WI	0.234	0.393	0.344	0.428	0.167	0.451	0.2477
what is the minimum spanning tree_5-6 vertices_Prim's algorithm; WI	0.2	0.447	0.345	0.639	0.31	0.732	0.2903
which is the n'th edge of the minimum spanning tree_7 vertices_Prim's algorithm; WI	0.452	0.492	0.25	0.554	0.256	0.612	0.3248
which is the n'th edge of the minimum spanning tree_5-6 vertices_Prim's algorithm; WI	0.55	0.53	0.516	0.553	0.522	0.576	0.5299

Table 8.10 Analysis of Prim's and Kruskal's algorithms questions.

The four questions asking if an edge was added, rejected, or not considered had significantly different facility and discrimination values throughout the analysis. Two questions have discrimination values less than 0.100, namely the question for a 7-vertex graph using Prim's algorithm (0.09; 2013 – 2014) and the question for a 7-vertex graph using Kruskal's algorithm (0.098; 2011 – 2012). The discrimination values for these two questions in other academic years vary greatly, reaching values as high as 0.647 for Prim's algorithm (2012 – 2013) and

0.699 for Kruskal's algorithm (2013 – 2014). These two questions also vary greatly in overall facility, with values of 0.1913 using Prim's algorithm and 0.3377 using Kruskal's algorithm. For all four questions, the range of facility is 0.344 and the range of discrimination values is 0.609, with the minimum and maximum values for both statistics coming from the same two questions for 7-vertex graphs and both occurring in the 2013 – 2014 academic year.

The four questions asking if an edge was added, rejected, or not considered require students to implement the algorithms fully, similar to the questions asking for the minimum spanning tree. However, with these four questions in particular, students then have to backtrack through their work to answer a more specific question. The number of steps involved in the method for answering these questions is therefore the same as the number of steps involved in the method for answering the questions that ask for the minimum spanning tree.

The last set of four questions asks students to determine the n^{th} edge to be added in either algorithmic process. These questions have generally higher overall facility values, ranging from 0.3248 to 0.5299. The facility values for three of these questions appear to have reduced from 2011 – 2012 values; the exception is the question using a graph of 5 – 6 vertices and Prim's algorithm, which has consistently good facility values with a range of only 0.034 from 2011 – 2014. The discrimination values are significantly positive, but with one anomaly, namely the question using a graph of 5 – 6 vertices and Kruskal's algorithm (2011 – 2012), which has a discrimination value of 0.265.

8.3.2.2 Degree

8.3.2.2.1 Indegree and Outdegree

The next five questions look at degree, indegree and outdegree. Unlike the 2008 – 2011 question on both topics, five separate questions were designed for 2011 – 2014, exploring each type of vertex degree in different ways. There are two questions on each of indegree and outdegree, but they are essentially identical, i.e. the elements of each question are the same throughout. However, there is one catch in the terminology, namely that a network matrix for a digraph has the weights attached, whereas the adjacency matrix would have the number of edges counted.

QUESTION	2011 - 2012		2012 - 2013		2013 - 2014		OVERALL FACILITY
Fac. = Facility Dis. = Discrimination	Fac.	Dis.	Fac.	Dis.	Fac.	Dis.	
indegree of the vertex of the network matrix of a digraph; NI	0.214	0.468	0.222	0.715	0.345	0.255	0.2787
outdegree of the vertex of the network matrix of a digraph; NI	0.273	0.527	0.429	0.402	0.357	0.407	0.3585
degree of the vertex of the network matrix (symmetric graph); NI	0.25	0.313	0.5	0.675	0.583	0.557	0.4630
indegree of the vertex of the adjacency matrix; NI	0.25	0.569	0.591	0.75	0.652	0.571	0.5439
outdegree of the vertex of the adjacency matrix; NI	0.438	0.626	0.842	0.029	0.786	0.053	0.6939

Table 8.11 Comparison of NI question on indegree and outdegree from 2011 – 2014.

The questions involving a network matrix involve an extra step in that students are not to count the numbers they see, but rather calculate the sum of non-zero entries they see. It is clear that this has had some effect on students' attempts, especially as the appearance of both questions has caused many students to not consider the extra step in their efforts. However, there may be an issue in the wording itself: the "network matrix of a digraph" and "adjacency matrix" could be similar in meaning for any question on indegree and outdegree. A network matrix typically does not include the weights of the edges, but network matrices do exist where the weights are included. Therefore, the wording has been shown to be the issue in this question and hence, no valid conclusions can be made about why students performed better on some questions on degree, indegree and outdegree than on other similar questions.

8.3.2.2.2 Degree Sequences

The next set of questions looks at degree sequences. Similar to the topic of indegree and outdegree, there is a significant range of facility values for the eight questions presented in this topic. The detailed analysis is provided in Table 8.12.

QUESTION	2011 - 2012		2012 - 2013		2013 - 2014		Overall Fac.
Fac. = Facility Dis. = Discrimination	Fac.	Dis.	Fac.	Dis.	Fac.	Dis.	
degree sequence of the simple, disconnected adjacency matrix; WI	0.688	0.62	0.61 9	0.554	0.412	0.787	0.5741
degree sequence of the graph (with multi edges and loops); WI	0.538	0.366	0.68 8	0.315	0.5	0.378	0.5778
degree sequence of the adjacency matrix (with multi edges); WI	0.471	0.575	0.64 7	0.455	0.85	0.608	0.6667
degree sequence of the adjacency m. w/ multi edges and loops; WI	0.692	0.499	0.75	0.063	0.667	0.665	0.7073
degree sequence of simple disconnected graph WI	0.533	0.514	0.81	0.607	0.778	0.114	0.7111
degree sequence of the adjacency matrix (simple, connected graph); WI	0.6	0.824	0.7	0.812	0.759	0.59	0.7119
degree sequence of the graph (with multi edges); WI	0.5	0.648	0.8	0.526	0.783	0.727	0.7193
degree sequence of the graph (simple, connected graph); WI	0.778	0.689	0.69 6	0.734	0.72	0.224	0.7193

Table 8.12 Comparison of degree sequence questions from 2011 – 2014.

Facility values are generally good throughout, ranging from 0.412 (2013 – 2014) to 0.85 (2013 – 2014). The discrimination values, however, are concerning as there are some significantly low discrimination values present, namely 0.063 (2012 – 2013) and 0.114 (2013 – 2014); these low discrimination values appeared one time in each of two academic years and for two different questions, but otherwise, the discrimination values were very good for these two questions and for each academic year.

Two of these questions, namely those using a simple, disconnected adjacency matrix and a graph with multi-edges and loops, appear to have been more difficult than the other six questions, all of which appear as a group (with one question on outdegree placed in the middle of the group) with high overall facility values for the 2011 – 2014 assessments. However, the two more difficult questions differ in terms of the information provided in answering the questions; there is either a graph or an adjacency matrix and the properties change from simple and disconnected to multi-edges and loops. Two questions on simple and connected graphs and adjacency matrices exist in the set and performed better than the other questions, although one question involving multi-edges did perform better than that using the simple and connected adjacency matrix. The analysis of this data shows that there is clearly no significant difference in facility between the various characteristics that define these questions uniquely.

8.3.2.3 Vertex Sets

There are two questions involving vertex sets. One question uses a disconnected graph and the other uses a connected graph. These two questions are identical to questions asked in the 2008 – 2011 data set. However, unlike the 2008 – 2011 data set, the data for 2011 – 2014 provides more conclusive evidence that the question involving the connected graph was answered more correctly than the question involving the disconnected graph. Results are shown in Table 8.13.

The facility values for these two questions range from 0.554 (2013 – 2014) to 0.859 (2012 – 2013). The discrimination values range from 0.417 (2013 – 2014) to 0.757 (2011 – 2012). Therefore, these questions performed generally well. However, the three lowest discrimination values all appear for the question involving the connected graph and in decreasing order from 2011 – 2014; in a similar fashion, the three discrimination values for the question involving the disconnected graph continued decreasing from 2011 – 2014.

QUESTION	**2011 - 2012**		**2012 - 2013**		**2013 - 2014**		Overall Fac.
Fac. = Facility Dis. = Discrimination	**Fac.**	**Dis.**	**Fac.**	**Dis.**	**Fac.**	**Dis.**	
Given disconnected graph_input vertex set; WI+check	0.591	0.757	0.675	0.702	0.554	0.627	0.6158
Given connected graph_input vertex set; WI+check	0.797	0.518	0.859	0.476	0.852	0.417	0.8387

Table 8.13 Comparison of vertex set questions from 2011 - 2014

Unlike the 2008 – 2011 data set, the facility values for each academic year were lower for the disconnected graph question than those for the connected graph question; also, the discrimination values for each academic year were higher for the disconnected graph question than those for the connected graph question. This shows that in the 2011 – 2014 data set, there was clearly an impact on the additional step required in answering the questions, i.e. determining if disconnected vertices belong to the vertex set of a graph.

8.3.2.4 Edge Sets

QUESTION	2011 - 2012		2012 - 2013		2013 - 2014		Overall Fac.
Fac. = Facility Dis. = Discrimination	Fac.	Dis.	Fac.	Dis.	Fac.	Dis.	
Given simple, connected graph_input edge set; WI+check	0.62	0.657	0.618	0.67	0.625	0.611	0.6205
Given graph with loops_input edge set; WI+check	0.604	0.778	0.658	0.632	0.691	0.498	0.6548

Table 8.14 Comparison of edge set questions from 2011 – 2014.

The next two questions involve edge sets. One question uses simple, connected graphs, whereas the other question includes loops in the graph. Interestingly, the question involving graphs with loops had a higher facility value overall than the question involving a simple and connected graph. However, these two questions appear close in the overall rankings for the questions in the 2011 – 2014 question set; the question involving simple and connected graphs had an overall facility of 0.6205, whereas the question involving the graph with loops had an overall facility of 0.6548. It is also worth noting, though, that 31 more attempts were made for the question involving the graph with loops, which is significant as only 166 attempts were made on the question involving the simple and connected graph between 2011 and 2014. Results are shown in Table 8.14.

8.3.2.5 Adjacency Matrices

The last two questions from the 2011 – 2014 question set investigate students' understanding of adjacency matrices. These two questions are very different in that there is a multiple-choice (MC) question and a responsive word input (with a check; RWI) question. The MC question provides a graph and asks students to find the matching adjacency matrix, whereas the RWI question provides a graph and the corresponding adjacency matrix before asking students what mistake was made within the adjacency matrix.

QUESTION	2011 - 2012		2012 - 2013		2013 - 2014		Overall Fac.
Fac. = Facility Dis. = Discrimination	Fac.	Dis.	Fac.	Dis.	Fac.	Dis.	
Given graph, find matching adjacency matrix; MC	0.614	0.639	0.657	0.459	0.64	0.556	0.6400
What is wrong with the adjacency matrix; RWI+check	0.684	0.653	0.686	0.654	0.816	0.629	0.7346

Table 8.15 Comparison of adjacency matrix questions from 2011 – 2014.

The results are very interesting as it is showing that the MC question did not perform as well as the RWI question. However, recall that the setup of MC questions is such that the correct answer is sometimes “None of these”; this means that if students go through all four possibilities and cannot find a mistake in any of them, then they should select “None of these”. To search for an error in four adjacency matrices is time consuming, so to not find an error in any of them may cause a sense of doubt in the minds of some students. However, with the RWI question, there is less time consumption as it is already known that an error exists within the single adjacency matrix provided and students are simply being asked to find it within that one adjacency matrix. Students clearly performed better with the RWI question and were more engaged with the question, attempting it more than the MC question.

However, these results are comparable to the 2008 – 2011 questions on the same topic; the MC question had an overall facility of 0.6311 from 2008 – 2011, compared to 0.6400 from 2011 – 2014, and the RWI question had an overall facility of 0.6814 from 2008 – 2011, compared to 0.7346 from 2011 – 2014, with 169 more attempts made. From 2008 – 2011, 21 more students attempted the MC question, but 23 more students attempted the RWI question from 2011 – 2014. Also note that there was an additional question in the 2008 – 2011 question set, asking to match a graph to a given adjacency graph, which had a higher overall facility than the other two questions from the 2008 – 2011 question set; however, also note that its facility value of 0.7150 is still lower than the 0.7346 facility value obtained for the RWI question in the 2011 – 2014 question set.

8.3.3 Hypothesis Testing

8.3.3.1 Test for Difference in Proportions Within Topics

Hypothesis testing was carried out on comparable questions within each topic. The results are shown in Table 8.16.

Topic	Issue	Z
Adjacency Matrices	No available questions	
Degree: Indegree vs. Outdegree of Graphs	Indegree vs. Outdegree	-0.9143
Degree: Indegree vs. Outdegree of Matrices	Indegree vs. Outdegree	-1.5812
Degree: Indegree	Graph vs. Adjacency Matrix	-2.9302
Degree: Outdegree	Graph vs. Adjacency Matrix	-3.3872
Degree Sequence: Simple & Disconnected	Graph vs. Adjacency Matrix	1.4114
Degree Sequence: Multi edges and Loops	Graph vs. Adjacency Matrix	-1.2496
Degree Sequence: Multi edges	Graph vs. Adjacency Matrix	0.6013
Degree Sequence: Simple & Connected	Graph vs. Adjacency Matrix	-0.0910
Degree Sequences	Simple & Connected vs. Multi-Edges and Loops	-0.0448
Degree Sequences	Simple & Disconnected vs. Multi-Edges	-0.8797
Degree Sequences	Simple & Disconnected vs. Simple & Connected	-1.2392
Degree Sequences	Multi-Edges & Loops vs. Multi-Edges	13.6489
Degree Sequences	Multi-Edges & Loops vs. Simple & Connected	-1.1477
Degree Sequences	Multi-Edges vs. Simple & Connected	-0.3604
Degree Sequences	Graph vs. Adjacency Matrix	0.4942
Edge Sets	Simple & Connected vs. Graph with Loops	-0.6787
Spanning Trees	Kruskal vs. Prim	1.8081
Spanning Trees	5-6 vertices vs. 7 vertices	2.4613
Spanning Trees	Minimum Spanning Tree vs. AB Edge	-0.0714
Spanning Trees	Minimum Spanning Tree vs. nth Edge	-4.9587
Spanning Trees	nth Edge vs. AB Edge	4.8058
Vertex Sets	Connected vs. Disconnected	5.0083

Table 8.16 List of Z_{test} values for hypothesis testing of questions within topics for questions tested from 2011 - 2014. Values highlighted in red show a rejection of the one-tailed test in favour of $H_1: \mu_1^{\alpha} < \mu_2^{\alpha}$. Values highlighted in blue show a rejection of the one-tailed test in favour of $H_1: \mu_1^{\alpha} > \mu_2^{\alpha}$.

The analysis shows that questions on indegree and outdegree were easier to answer if the questions used adjacency matrices instead of graphs. However, there were no significant differences in the proportions of correct answers between the topics of indegree and outdegree.

The only significant difference in proportions for questions on the topic of degree sequences was for the questions between multi-edges & loops and multi-edges. Questions that included multi-edges and loops were significantly easier for students to answer than those that did not also have loops. However, issues of graphs vs. adjacency matrices were not significantly different, nor were any other graphical comparison; this could suggest that the use of graphs with multi-edges and loops is a significant factor in understanding this topic.

Most comparisons for the questions on spanning trees resulted in significant differences in the proportions of correct answers given. Students appeared to have found Kruskal's algorithm to be easier than Prim's algorithm. Also, fewer vertices in the graph resulted in higher facility values. The questions on finding the n^{th} edge have larger proportions of correct answers than either of the other two questions, suggesting it may be an easier question for students to answer.

The questions on vertex sets show that those questions involving connected graphs resulted in significantly higher facility values than the same questions using disconnected graphs. It is possible that students neglected disconnected vertices in answering this question.

8.3.3.2 Test for Difference in Proportions Between Topics

Topic 2 → / Topic 1 ↓	Degree	Edge Sets	Kruskal	Prim	Vertex Sets	Degree Sequences
Adjacency Matrices	4.9882	1.6452	11.8134	14.7304	-1.7456	0.4282
Degree		-3.5289	2.3515	3.5924	-5.7964	-4.0984
Edge Sets			8.4935	10.7095	-2.9583	-0.9404
Kruskal				1.8081	-11.6385	-8.7897
Prim					-14.0404	-10.8306
Vertex Sets						1.7997

Table 8.17 List of Z_{test} values for hypothesis testing of questions between topics for questions tested between 2011 - 2014. Values highlighted in red show a rejection of the one-tailed test in favour of $H_1: \mu_1^\alpha < \mu_2^\alpha$. Values highlighted in blue show a rejection of the one-tailed test in favour of $H_1: \mu_1^\alpha > \mu_2^\alpha$.

Test values for the comparisons between question topics for comparable questions are presented in Table 8.17.

Questions on vertex sets had significantly larger facility values than any other topic. Questions on Prim's algorithm had significantly lower facility values than any other topic, followed by Kruskal's algorithm. The shortest path algorithms questions involve multiple steps and some prerequisite understanding about graphs, so if a student makes a mistake somewhere in the algorithmic process, then it is likely (s)he will not submit a correct answer. Also, the shortest path algorithm questions did not have the pop-up check appearing, which may have factored into the lower facility values.

The only comparisons which did not result in a rejection of the null hypothesis are the comparisons of questions involving degree sequences with either adjacency matrices or edge sets. In the 2008 – 2011 comparisons, the topic of degree included questions on degree and degree sequences, but there were also fewer questions to be compared. However, questions on degree were replicated for the 2011 – 2014 assessments to ensure questions on both indegree and outdegree appeared in the students' tests; similarly, multiple replications of the questions on degree sequences were created to assess students further on similar questions involving different types of graphs. The replications of the degree sequence questions had the pop-up boxes removed, so they no longer had the double-checking capability. Questions on degree, which were NI questions, had significant higher facility values than questions on degree sequences, which were WI questions.

8.3.3.3 Test for Difference in Proportions Between Question Types

Test values for the comparisons between question types for the entire data set are given in Table 8.18.

Again, WI + Check questions proved to have significantly higher facilities than other questions. However, as was the case with the 2008 – 2011 question set, the topics for these questions do not vary; the topics that included WI + Check were adjacency matrices, edge sets, and vertex sets. Therefore, it is more likely that WI + Check questions were easier because of the topics they covered. Five WI + Check questions were in this question set.

Type 2 → / Type 1 ↓	NI/RNI	WI/RWI	WI+Check
MC	4.5000	7.8486	-2.3081
NI/RNI		1.5592	-7.6806
WI/RWI			-15.2587

Table 8.18 List of Z_{test} values for hypothesis testing of questions between question types for questions tested between 2011 - 2014. Values highlighted in red show a rejection of the one-tailed test in favour of $H_1: \mu_1^\alpha < \mu_2^\alpha$. Values highlighted in blue show a rejection of the one-tailed test in favour of $H_1: \mu_1^\alpha > \mu_2^\alpha$.

Again, MC questions were easier than the NI/RNI and WI/RWI questions. In this question set, however, there was only one MC question on adjacency matrices, so it is likely that MC questions also scored better because of the topic.

There is no significant difference in the proportions of correct answers given between the NI/RNI questions and the WI/RWI questions. These question types represent $\frac{25}{31}$ of the questions provided to students, even though 20 of these questions were WI/RWI questions. NI/RNI questions were only on the topics of indegree and outdegree, whereas the WI/RWI questions included the topics of shortest path algorithms (12) and degree sequences (8).

Chapter 9 Statistical Analysis and Review

9.1 Introduction

This chapter discusses any possible conclusions that can be raised from the statistics presented in chapters 6 and 7, following from the 2007 – 2014 assessments conducted at Brunel University. Following discussions of the results, further discussion will investigate limitations to the research conducted and recommendations for future analyses.

9.2 2007 – 2008 Assessment Conclusions

The 2007 – 2008 assessments were designed for second-year undergraduate students studying mathematics at Brunel University. The course module was designed to focus on two subjects over the academic year, with graph theory being studied in the second semester. These assessments were designed to provide students with ample opportunities to practise using the online software and answering questions online in graph theory prior to sitting an invigilated, online assessment later. Two sets of assessments were designed, one involving graphs only and the other involving adjacency matrices only. Since this format was not repeated in later academic years, these results cannot be used reliably in forecasting models.

The facility values of the visual question set attempts show significantly lower facility values for RNI questions than for MC questions, suggesting that RNI questions may be challenging for graph theory. However, both RNI questions in the question set relate to numbers of spanning trees involving an intermediate or advanced level concept, implying that these questions may have reasonably been more challenging because of the difficulty level of the learning material. Questions on Hamiltonian and Eulerian cycles, as well as planar graphs, may be challenging as well, but also may have been easier to answer as MC questions; nonetheless, there is not enough evidence to make a sufficient conclusion regarding these specific topics.

In comparison, for the logical / mathematical question set, spanning trees questions again received lower facilities, meaning these questions were challenging. However, the NI question on the number of vertices in a partition of a bipartite graph was answered reasonably well with a facility value of 0.587, implying that the question type is not significant in the design of questions, but rather the difficulty level of the learning material. Additionally, one question was answered correctly by every student, proven by a zero value for the standard deviation of results and a facility value of 1; the correlation value of -1 shows this question was not helpful in the overall assessment and therefore, has not been used in later assessments.

In the invigilated test session, there were nineteen questions used, eighteen of which were MC questions and only one RNI question on the number of spanning trees in a graph. The RNI question was again challenging with a significantly lower facility value, and the MC questions had significantly higher facility values (with only one exception). However, in this case, some of the MC questions were questions on spanning trees; these questions, though, focussed on finding a spanning tree rather than calculating the total number of possible spanning trees available, which may have been an easier task to complete. Additionally, the MC question on spanning trees that had a lower facility value also had a low, yet still positive, correlation; this implies that this particular question did not fare well in this particular assessment. This MC question may be better suited in another assessment, particularly one which focusses well on spanning trees, but it does not appear to have been well suited to this assessment, which looks at a range of topics.

There were no negative correlations in any of the three assessments, with the exception of the question noted earlier from the logical / mathematical question set in which every student answered the question correctly when attempted. Although some of the correlations are closer to having zero correlation, the fact almost every question has a positive correlation suggests the questions were well structured for the assessments given. The questions were therefore effective for assessment purposes.

9.3 2008 – 2014 Assessment Conclusions

The 2008 – 2014 assessments were similar to each other in that all questions were assessed using the same scheme, namely that a correct answer is awarded one mark and an incorrect answer is awarded zero marks. From 2008 – 2011, the scheme of work for the assessments was consistent, but the scheme of work changed in 2012, with the 2011 – 2014 assessments being consistent in their own right. This group of six assessments was therefore reviewed in two halves as the change in the scheme of work may have an effect on forecasting models.

9.3.1 2008 – 2011 Assessment Results

It was encouraging that the spearman rank correlations between the academic years' results for 2008 – 2011 was significantly positive. However, there are a couple of possible dangers to having such significantly positive correlations:

- Without further evidence, it may be wrongfully assumed that later results were improved on earlier results; the statistics only shows that, generally speaking, an increase from one academic year to the next was consistent throughout all questions, but it does not specify in which direction this occurred.
- It may be possible that past students, still enrolled at Brunel University at this point, may have spoken to "current" students enrolled in the module about the online assessments. If this is the case, then the "current" students would have gone into the assessments with a better understanding of the assessments than previous students and thus, possibly were better prepared.

Overall facilities for these academic years were consistent with expectations as it was hoped an overall facility of 0.5 was obtained. A majority of topics also had facility values between 0.4 and 0.6, implying questions were individually well structured as well. There were no negative discrimination values in any of the assessments, implying that all questions served well in the overall assessments. All of these things show that the questions were well designed for these assessments overall.

It was then noted through hypothesis testing that the median scores of the 2008 – 2009 assessments were significantly different to the other assessments (at $\alpha = 0.05$). As there were positive correlations in facility values from 2008 – 2011, it is likely that there was a consistent increase or decrease in overall results from one academic year to the next academic year. However, the overall facility values for each academic year do not follow a consistent pattern, going from 0.4918 to 0.4621 to 0.5475 in order, but the numbers of attempts at questions does increase consistently, from 854 to 1199 to 1359 in order; this is not to suggest that the numbers of attempts is significant in itself, but rather that this may have had some effect in the analysis and comparisons. It is also worth noting that the comparisons of medians of discrimination values were not rejected at $\alpha = 0.05$ for any of the assessments from 2008 – 2011; since discrimination values were all positive throughout these assessments, it shows these questions were consistently well structured for these assessments.

For four of the topics presented in these assessments, there were significant differences in mean facility values obtained. Questions on adjacency matrices & simple and connected graphs had significantly improved results when adjacency matrices were used; this is noteworthy for future assessments as teachers / lecturers could decide to make use of this to either provide easier assessments or challenge students further to better understand these topics. However, simple & connected graphs questions had better results when graphs were used instead of a combination of graphs and adjacency matrices. Having a combination of answer types may involve more effort on the part of the student, but this would need to be analysed further in future research. Additionally, edge sets questions were significantly better when digraphs were used. This may be understandable for these assessments as there were specific instructions given on the formatting of answers in these questions, which may have had some impact on the results.

The comparisons of topics for 2008 – 2011 showed that adjacency matrices questions had significantly better results than most other questions; the only exception was for vertex sets, where the null hypothesis could not be rejected. It was also shown that bipartite graphs questions had significantly lower results than all other topics in the question set. It was the case that for most comparisons, there were significant differences in facility values between topics. However, this ought to be expected as different topics will present different issues and problems for students. The structuring of questions between topics will be different and so, very few comparisons were expected not to have been rejected at $\alpha = 0.05$. Nonetheless, it is noteworthy that questions on adjacency matrices had significantly better results than most other topics and that questions on bipartite graphs had significantly lower results than all other topics because this shows an apparent variation in difficulty levels between the topics, which is helpful for future consideration by researchers and teachers / lecturers.

In the comparisons of question types, it was encouraging to see that WI questions that had pop-up windows appearing proved effective. The WI + Check questions all had significantly better results than all other question types in all assessments. This shows a positive impact of the pop-up window appearing, getting students to double check their answers prior to officially submitting their responses. Any questions where the formatting of answers is important should have this pop-up check provided as it would be rather unfortunate for students to have obtained the correct answers, but then receive no marks for their efforts because the formatiing of their answers does not conform to the question standards.

9.3.2 2011 – 2014 Assessment Results

Again, it was encouraging to see all positive correlations between the 2011 – 2014 assessments, but as noted earlier, this does not necessarily correspond to increased, continuous success from one academic year to the next academic year. For these assessments, the numbers of attempted questions goes from 985 to 1235 to 1143. Overall facility values go from 0.4735 to 0.5733 to 0.5529, which does not correspond necessarily to the spearman rank correlations, but are still good results.

A majority of questions has discrimination values ranging from 0.4 to 0.6, which is encouraging, but the majority of facility values are either to the left or to the right of the 0.4 to 0.6 range for facility, implying that questions were either easier or more difficult to answer. These questions did still serve a purpose towards the assessments as all discriminations are positive and to some extent, having larger and smaller facility values does balance out, but it would be preferable for future considerations to have facility values focussed in the 0.4 to 0.6 range.

Questions on shortest paths were understandably more challenging as facility values were lower for these questions than other questions. The topic of shortest paths is more challenging than all other topics presented and the facility values of all other questions were significantly higher, thus possibly explaining the fluctuations in facility values whilst maintaining acceptable overall facility values.

Discrimination values varied more significantly for these assessments than the 2008 – 2011 assessments. With questions seemingly being either more challenging or less challenging, it is not surprising that discrimination values varied so greatly for these assessments. However, all discrimination values were positive, meaning that they all were effective in some measure in the assessments provided.

Hypothesis testing within topics showed that there were some significant effects on the styles of questions presented. Questions on degree had higher facility values when adjacency matrices were involved; this is understandable as it ought to be easier to use the numbers in adjacency matrices to calculate the degree of a vertex. Questions on degree sequences had higher facility values when graphs presented had both multi-edges and loops instead of just multi-edges; there could be a formatting issue involved as edges had to be provided in alphabetical order, but there is not enough information on this alone to make a reasonable conclusion. Questions on vertex sets had higher facility values when graphs were connected rather than disconnected; this is understandable as some students may have omitted the disconnected vertices from their answers (noting that this was somewhat expected to occur). However, questions on minimal spanning trees were significantly different in almost every comparison. It may be possible that students could have confused Prim's and Kruskal's algorithms and it may be possible that the increased number of vertices allowed more opportunities for incorrect answers to appear. However, this does highlight significantly the importance of effective question design as results can be greatly altered by the presentation and style of a question. It is also worth noting that in the case of finding the "AB edge", any edges that had identical weights to other edges may have impacted results due to the alphabetical formatting requirement within these particular questions.

Hypothesis testing between topics showed that questions on vertex sets had better facilities than all other topics, followed by questions on adjacency matrices. Kruskal and Prim's algorithms had significantly lower facility values than all other topics; these algorithms require more prerequisite knowledge than other topics presented and it is quite likely that this factors into the facility values. However, it must also be noted that there were programming issues found with the design of questions on Kruskal and Prim's algorithms as formatting of answers was not double-checked through pop-up windows appearing, asking students to double-check their answers prior to submitting a second time. Statistical analyses indicated pop-up windows were significantly helpful in increasing facility values in word input questions, so these should have been included in these questions to avoid any potential issues from occurring.

9.4 Further Considerations

This section will look at additional statistical analyses that have been considered, but not thoroughly investigated for the purpose of this thesis. Recall that the objective of this thesis in terms of statistical analysis was to investigate the performance of the questions themselves for the purposes of online assessment and not to investigate the impact these questions may have on students. These statistical analyses explore briefly some of the statistical analyses that could be considered and what results can be obtained from the data already collected.

From the 2007 – 2008 assessments, an exploration of test-retest coefficients and numbers of attempts will explore the impact of attempting similar assessments as a means to explore assessment and learning. Additionally, other statistical methods will be considered and reviewed.

From the 2008 – 2014 assessments, a time series analysis will investigate any trends in final examination scripts and what to expect from future cohorts of students in MA0422 at Brunel University.

9.4.1 2007 – 2008 Assessment Considerations

9.4.1.1 Final Examination Analyses

The tables below show all of the quantitative results for the final examination scripts performed by students in MA2920: Algebra and Discrete Mathematics, for the 2005 to 2008 examination periods. Recall that these examination scripts were read to gather some additional insight into errors made by students in their examinations; details of noted errors appear in Appendix D. Quantitative data provided includes descriptive, statistical results, along with t-testing results of various comparisons between sets of examination scripts.

	2004-2005	2005-2006	2006-2007	2007-2008
Mean	4.416667	3.4	3.564516	4.245902
Standard Error	0.241584	0.270175	0.282226	0.202317
Median	5	3	4	5
Mode	6	6	6	5
Standard Deviation	1.673744	2.092764	2.222249	1.580145
Skewness	-0.81468	-0.18732	-0.29687	-0.91248
Range	5	6	6	6
Minimum	1	0	0	0
Maximum	6	6	6	6
Count	48	60	62	61

Table 9.1 Descriptive statistics for examination questions asking to determine the number of spanning trees of a graph with a large number of vertices. The maximum obtainable score for these questions was 6 marks each time.

In the 2004 – 2008 final examination scripts for MA2920, two questions repeatedly appeared for graph theory, focusing on the calculation of a number of spanning trees for a graph and vertex colouring of a graph. Table 9.1, Table 9.3, and Table 9.4 investigate the number of spanning trees for a graph, whereas Table 9.2, Table 9.5, and Table 9.6 investigate the vertex colouring of a graph.

	2004-2005	2005-2006	2006-2007	2007-2008
Mean	0.642276	0.472222	0.609091	0.348387
Standard Error	0.041386	0.057697	0.059581	0.042216
Median	0.666667	0.583333	0.7	0.3
Mode	0.666667	0.833333	1	0.2
Standard Deviation	0.264997	0.346181	0.395215	0.235047
Skewness	-0.33658	-0.25094	-0.38804	0.270676
Range	0.833333	0.833333	1	0.8
Minimum	0.166667	0	0	0
Maximum	1	0.833333	1	0.8
Count	41	36	44	31

Table 9.2 Descriptive statistics for examination questions asking to determine the number of colours needed to colour a particular graph. Quantitative data presented reflects equivalent percentage scores for questions given.

	2004-2005	vs. 2005-2006	vs. 2006-2007	vs. 2007-2008
Mean	4.416667	3.4	3.564516	4.245902
Variance	2.801418	4.379661	4.938392	2.496858
Observations	48	60	62	61
Degrees of freedom		106	108	98
t Statistic		**2.805121**	**2.293793**	**0.54192**
One-tailed critical value for t		**1.659356**	**1.659085**	**1.660551**
Two-tailed critical value for t		**1.982597**	**1.982173**	**1.984467**
		2005-2006	vs. 2006-2007	vs. 2007-2008
Mean		3.4	3.564516	4.245902
Variance		4.379661	4.938392	2.496858
Observations		60	62	61
Degrees of freedom			120	110
t Statistic			**-0.42108**	**-2.50615**
One-tailed critical value for t			**1.657651**	**1.658824**
Two-tailed critical value for t			**1.97993**	**1.981765**
			2006-2007	vs. 2007-2008
Mean			3.564516	4.245902
Variance			4.938392	2.496858
Observations			62	61
Degrees of freedom				110
t Statistic				**-1.96222**
One-tailed critical value for t				**1.658824**
Two-tailed critical value for t				**1.981765**

Table 9.3 Table of T distribution results for all final examination pairings for the questions looking at the number of spanning trees of a graph of a large number of vertices. Results highlighted in red indicate where the null hypothesis is rejected in favour of $H_1: \mu_1 > \mu_2$. Results highlighted in green indicate where the null hypothesis is rejected in favour of $H_1: \mu_1 < \mu_2$.

It can be seen from the descriptive statistics that after a sharp decline in 2005, the mean scores for the questions on the number of spanning trees in a graph continued to increase, whereas the mean scores for the questions on vertex colouring continued to fluctuate. In the 2007 – 2008 academic year, when Mathletics was presented to MA2920 for the first time, the mean score for the question on vertex colouring decreased significantly, implying that it may be possible that any Mathletics questions on vertex colouring were not designed in line with the MA2920 syllabus. However, this will be investigated in later analyses.

The academic year comparisons of final examination results for the questions on the number of spanning trees in a graph show that there was a significant change in results between 2004 – 2005 and both 2005 – 2006 and 2006 – 2007, specifically showing that the 2004 – 2005 results were significantly better than the other results. However, it is also shown that the 2005 – 2006 and 2006 – 2007 results were significantly lower than the 2007 – 2008 results, when Mathletics was introduced to the cohort; this shows it may be possible that the implementation of Mathletics was significant in improving overall results for questions on the number of spanning trees to appear in a graph.

	2004-2007	2007-2008
Mean	3.747059	4.245902
Variance	4.27292	2.496858
Observations	170	61
Degrees of freedom		138
t Statistic		**-1.94076**
One-tailed critical value for t		**1.65597**
Two-tailed critical value for t		**1.977304**

Table 9.4 Table of T distribution results for the comparison of all final examinations in MA2920 looking at the question regarding the number of spanning trees of a graph of a large number of vertices. Results highlighted in green indicate where the null hypothesis is rejected in favour of $\boldsymbol{H_1: \mu_1 < \mu_2}$.

The academic year comparisons of final examination results for questions on vertex colouring show a significant decrease in results in 2007 – 2008, when Mathletics was introduced. However, recall that there were no Mathletics questions on vertex colouring presented in the 2007 – 2008 assessments due to time constraints with the presentation of the assessments; if such questions had been allowed in the assessments, then these results may have changed significantly.

	2004-2005	vs. 2005-2006	vs. 2006-2007	vs. 2007-2008
Mean	0.642276	0.472222	0.609091	0.348387
Variance	0.070224	0.119841	0.156195	0.055247
Observations	41	36	44	31
Degrees of freedom		65	76	68
t Statistic		**2.394965**	**0.457453**	**4.97123**
One-tailed critical value for t		**1.668636**	**1.665151**	**1.667572**
Two-tailed critical value for t		**1.997138**	**1.991673**	**1.995469**
		2005-2006	vs. 2006-2007	vs. 2007-2008
Mean		0.472222	0.609091	0.348387
Variance		0.119841	0.156195	0.055247
Observations		36	44	31
Degrees of freedom			78	62
t Statistic			**-1.65024**	**1.732155**
One-tailed critical value for t			**1.664625**	**1.669804**
Two-tailed critical value for t			**1.990847**	**1.998971**
			2006-2007	vs. 2007-2008
Mean			0.609091	0.348387
Variance			0.156195	0.055247
Observations			44	31
Degrees of freedom				71
t Statistic				**3.570265**
One-tailed critical value for t				**1.6666**
Two-tailed critical value for t				**1.993943**

Table 9.5 Table of T distribution results for all final examination pairings for the questions looking at the number of colours needed to colour a graph. Results highlighted in red indicate where the null hypothesis is rejected in favour of $H_1: \mu_1 > \mu_2$.

	2004-2007	2007-2008
Mean	0.579614	0.348387
Variance	0.119451	0.055247
Observations	121	31
Degrees of freedom		67
t Statistic		**4.393881**
One-tailed critical value for t		**1.667916**
Two-tailed critical value for t		**1.996008**

Table 9.6 Table of T distribution results for the comparison of all final examinations in MA2920 prior to the introduction of CAA material in graph theory with the 2007-2008 examination, after CAA was introduced, and looking at the question regarding the number of colours needed to colour a particular graph. Results highlighted in red indicate where the null hypothesis is rejected in favour of $H_1: \mu_1 > \mu_2$.

Unfortunately, as these assessments were never replicated and as MA2920 was eventually discontinued, it will not be known from this research what impact vertex colouring questions in Mathletics could have in other assessments; fortunately, though, these questions have not been deleted and future research could investigate this in depth to determine any benefits that may appear from presenting such problems to students using Mathletics.

9.4.1.2 Implementing Other Statistical Methods

To further investigate the validity of the results, it helps to understand what past, statistical research in objective, online learning and assessment has provided to educators and education researchers, especially in the United Kingdom as the original focus of the subject material came from a U.K. education curriculum.

Farrell and Leung, in their work on IT education using confidence-based measurement[62], utilise the Kolmogorov-Smirnov test for normality to check for an (approximately) normal distribution. However, the test fails and so, the Wilcoxon Signed Rank Test for non-parametric data with repeated measures is used. Using SPSS to generate the required data, the Wilcoxon Signed Rank Test[63] gives an associated significance level, p, which defines to what level of significance a null hypothesis cannot be rejected. They also employ various measures of correlation used in this thesis to further interpret the results.

The Wilcoxon Signed Rank Test looks at the differences in data, but it investigates two observations made on the same subject each time. In comparison, this thesis wanted to investigate the scores of different individuals in their final examinations with or without the help of Mathletics. Since different individuals are being investigated in the experiments held, the Wilcoxon Signed Rank Test was not used.

Davies[63] discusses the differences in assessment between the student himself / herself, the student's peers, and the tutor's original and final marks. Much of this discussion relates to mark consistency and correlation only between peer marks. In the experiment conducted in this thesis, it would be possible to replace the number of peers with the number of attempts made by students. However, this would only show any consistency in their respective attempts and instead, we are looking for significant improvement in their attempts. Therefore, this strategy was not employed.

MacGillivray[64] uses scatterplots to find significant, positive correlation between quizzes set by the lecturer and the final exam scores and the smaller, positive correlation between own-choice group project work and final exam scores. In a second example, she uses p-values to justify significant predictors of the assignments score for a statistical modelling course / module. However, such techniques have been used in this thesis and no other methods appear to be used.

In comparison, Means, Toyama, Murphy, Bakia, and Jones[65] investigated 51 empirical studies of online learning. The empirical studies explored had a large variance of student numbers, ranging from 16 to 1,857, and the range of learner ages was 31 years. Analysts used a $\alpha < .05$ level of significance for testing differences. In their meta-analysis, they use retention rates for online and face-to-face learning, as well as a **weighted mean effect size*, Hedges'*** *g*+, and the **Q-statistic**, which determines the extent to which the variation in effect sizes cannot be explained by the sampling error alone. However, Huedo-Medina, Sánchez-Meca, Marín-Martínez, and Botella[66] argue that the Q-statistic only reports on the existence of heterogeneity in a meta-analysis, whereas the **I^2 Index** has been used to measure the extent to which heterogeneity exists and requires the Q-statistic value in its calculation.

The discussion on the Q-statistic and the I^2 index is interesting, but it involves the analysis of outcomes within each element to be tested. This could be useful if each question type was assessed in this thesis as elements, but with different marking schemes in each question type used as outcomes. Also, the outcomes do not have to be the same for each element; for example, numerical input questions (as an element) could explore NI, xNI, NAI, and xRNAI questions, whereas word input questions could explore its use in algorithms versus graph theory. However, since the Q-statistic uses a weighting factor and assumes a fixed effects model is being used, both the Q-statistic and the I^2 index are not preferred methods to use for this research.

Looking back at *g*+, we have the following equation for the weighted mean effect size:

$$(g+) = \frac{\overline{x_1} - \overline{x_2}}{\sqrt{\frac{\sum_{i=1}^{n_1}(x_i - \overline{x_1})^2 + \sum_{j=1}^{n_2}(x_j - \overline{x_2})^2}{n_1 + n_2 - 2}}}$$

Equation 9.1 Equation for the weighted mean effect size.

	g+	**Result for two-tailed test**	**Result for one-tailed test**
2004-2005 vs. 2005-2006	-0.529983	$H_1 : \mu_1 \neq \mu_2$ accepted.	$H_1 : \mu_1 > \mu_2$ accepted.
2004-2005 vs. 2006-2007	-0.425628	$H_1 : \mu_1 \neq \mu_2$ accepted.	$H_1 : \mu_1 > \mu_2$ accepted.
2004-2005 vs. 2007-2008	-0.105285	Fail to reject $H_0 : \mu_1 = \mu_2$	Fail to reject $H_0 : \mu_1 = \mu_2$
2005-2006 vs. 2006-2007	0.076181	Fail to reject $H_0 : \mu_1 = \mu_2$	Fail to reject $H_0 : \mu_1 = \mu_2$
2005-2006 vs. 2007-2008	0.456721	$H_1 : \mu_1 \neq \mu_2$ accepted.	$H_1 : \mu_1 < \mu_2$ accepted.
2006-2007 vs. 2007-2008	0.352916	$H_1 : \mu_1 \neq \mu_2$ accepted.	$H_1 : \mu_1 < \mu_2$ accepted.
2004-2007 vs. 2007-2008	0.260343	$H_1 : \mu_1 \neq \mu_2$ accepted.	$H_1 : \mu_1 < \mu_2$ accepted.

Table 9.7 Table of weighted mean effect sizes compared to t-test results for data presented in Section 9.4.1.1.

The value of *g*+ is a quotient of a difference of means and the pooled standard deviation, s_{pooled}, of the two samples being compared; but then, since $s_{pooled} > 0$ and since the effect is based primarily on the sign of the calculated value, the difference between the two sample means determines the end effect for *g*+. If one student were to do exceptionally well on a test, the mean score could increase significantly. Thus, *g*+ does not take the spread of the data into account.

Out of interest, the value of *g*+ was calculated for each possible pairing between MA2920 final examination scripts (Recall the t-statistic used $\alpha = 0.05$.). Calculated *g*+ values are shown in Table 9.7.

A strong, negative value can represent an earlier examination's scores being much better than that of a later examination; similarly, a strong, positive value can represent a later examination's scores being much better than that of an earlier examination. In the cases where *g*+ was close to 0, the hypothesis tests failed to reject the null hypothesis, as expected. Therefore, calculating *g*+

appears to have the same effect in understanding the differences in test scores as the t-test results and hence, is not to be considered further in this thesis.

9.4.1.3 Test-Retest Coefficients and Numbers of Attempts

9.4.1.3.1 Visual Question Set Assessment Results

The correlation matrix for the practice results of the visual question set are given in Figure 9.1.

	1^{st}	2^{nd}	3^{rd}	4^{th}	5^{th}	6^{th}	7^{th}	8^{th}	$\cdots$	44^{th}
1^{st}	1	**0.169**	0.248	0.353	0.491	0.104	0.396	0.467	–	–
2^{nd}	0.169	1	**0.375**	0.337	0.260	0.584	0.538	0.231	–	–
3^{rd}	0.248	0.375	1	**0.630**	0.364	0.735	0.777	0.526	–	–
4^{th}	0.353	0.337	0.630	1	**0.386**	0.444	0.428	0.237	–	–
5^{th}	0.491	0.260	0.364	0.386	1	**0.633**	0.429	0.582	–	–
6^{th}	0.104	0.584	0.735	0.444	0.633	1	**0.741**	0.819	–	–
7^{th}	0.396	0.538	0.777	0.428	0.429	0.741	1	**0.677**	–	–
8^{th}	0.467	0.231	0.526	0.237	0.582	0.819	0.677	1	–	–
$\vdots$	–	–	–	–	–	–	–	–	$\cdots$	–
n	16	6	9	6	4	2	4	3	$\cdots$	1

Figure 9.1 Correlation matrix for attempts made by students on practice questions involving only graphs, along with the number of attempts made by a particular number of students in each case.

Since the correlation matrix is symmetric, we only need to consider the upper triangular set of data, i.e. all data points above the diagonal of 1s. Also, because the test-retest coefficient looks at the correlation between two tests, we only need to consider the correlations of adjacent tests in an $n \times n$ correlation matrix, i.e. all data points found at positions, $a_{i,(i+1)}$, where $1 \leq i \leq n - 1$. The values highlighted in bold print in refer to these values for which we should consider. In each case, the test-retest coefficients being considered are all positive, which suggest that each trial helped students to progress further in their understanding of the course/module material. Also, these values generally increased from the 1st attempt through to the last attempt, with only one distinct exception, which all suggests that students are able to maintain abilities learned in previous attempts to perform even better on later attempts.

Note that there is no correlation between a student not attempting any of these tests and actually performing the first attempt; this is a reasonable assumption as there would normally be no statistical data to compare prior to the first attempt. Regardless, the slope/gradient for the least squares regression line is positive, which helps to show that students are able to hold onto the material they learned in previous attempts in order to do better on later attempts. However, the coefficient of determination, $R^2 = 0.6998$, also tells us that approximately $\frac{7}{10}$ of the variability in the data can be explained in the regression line. Since students were allowed to trial these questions at their own leisure and as often as they liked and also since many students live off-campus, implying they have no immediate access to trial these questions, this value for the coefficient of determination seems reasonable under these circumstances.

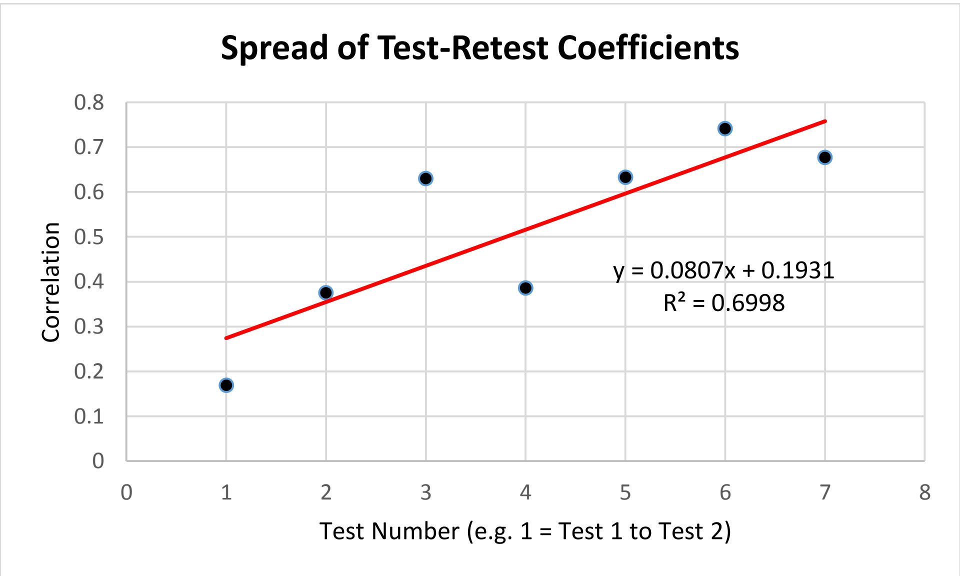


Figure 9.2 Line graph of test-retest coefficients and least squares regression line (with assumption that the y-intercept is equal to zero) for students' attempts at the practice questions involving only graphs.

It should first be noted that almost all students performed eight attempts or less. However, one student was very keen with practicing these questions and so, (s)he trialled the visual questions 44 times. Although this creates an outlier, it is preferable to include all students in the study due to their different abilities and study habits and so, the outlier is being considered.

9.4.1.3.2 Logical / Mathematical Question Set Assessment Results

The correlation matrix for the practice results of the logcal / mathematical question set are given in Figure 9.3.

At most eleven attempts were performed on the logical / mathematical practice question set for graph theory. The coefficient of determination is 0.1863. This data, although containing all positive values, appears to have an outlying value of 0.023 for the correlation between the 6^{th} and 7^{th} attempts; this significantly affects the coefficient of determination, as shown in Figure 9.4. If this outlying value were removed, then the coefficient of determination would more than double to 0.4073.

	1^{st}	2^{nd}	3^{rd}	4^{th}	5^{th}	6^{th}	7^{th}	8^{th}	9^{th}	10^{th}	11^{th}
1^{st}	1	**0.321**	0.302	0.152	–0.094	0.214	0.439	0.260	0.034	0.650	0.776
2^{nd}	0.321	1	**0.550**	0.339	0.322	0.246	0.771	–0.118	0.223	0.835	0.971
3^{rd}	0.302	0.550	1	**0.369**	0.300	–0.027	0.782	–0.310	0.293	0.806	0.984
4^{th}	0.152	0.339	0.369	1	**0.411**	0.124	0.302	–0.676	0.004	0.397	0.946
5^{th}	–0.094	0.322	0.300	0.411	1	**0.298**	0.398	0.713	0.707	0.579	0.018
6^{th}	0.214	0.246	–0.027	0.124	0.298	1	**0.023**	0.389	0.366	0.889	0.887
7^{th}	0.439	0.771	0.782	0.302	0.398	0.023	1	**0.308**	0.822	0.919	0.652
8^{th}	0.260	–0.118	–0.310	–0.676	0.713	0.389	0.308	1	**0.614**	0.186	–0.650
9^{th}	0.034	0.223	0.293	0.004	0.707	0.366	0.822	0.614	1	**0.619**	0.129
10^{th}	0.650	0.835	0.806	0.397	0.579	0.889	0.919	0.186	0.619	1	**0.745**
11^{th}	0.776	0.971	0.984	0.946	0.018	0.887	0.652	–0.650	0.129	0.745	1
n	10	12	4	6	5	7	3	0	1	1	4

Figure 9.3 Correlation matrix for attempts made by students on practice questions mainly involving adjacency matrices, along with the number of attempts made by a particular number of students in each case.

The correlations appear to be generally decreasing until around the 10^{th} attempt, when they begin to increase considerably. As the test-retest coefficient was considerably larger at around this point, it suggests that the test-retest coefficients may correspond to an improvement in student learning rather than a decrease in the number of students reattempting the problems.

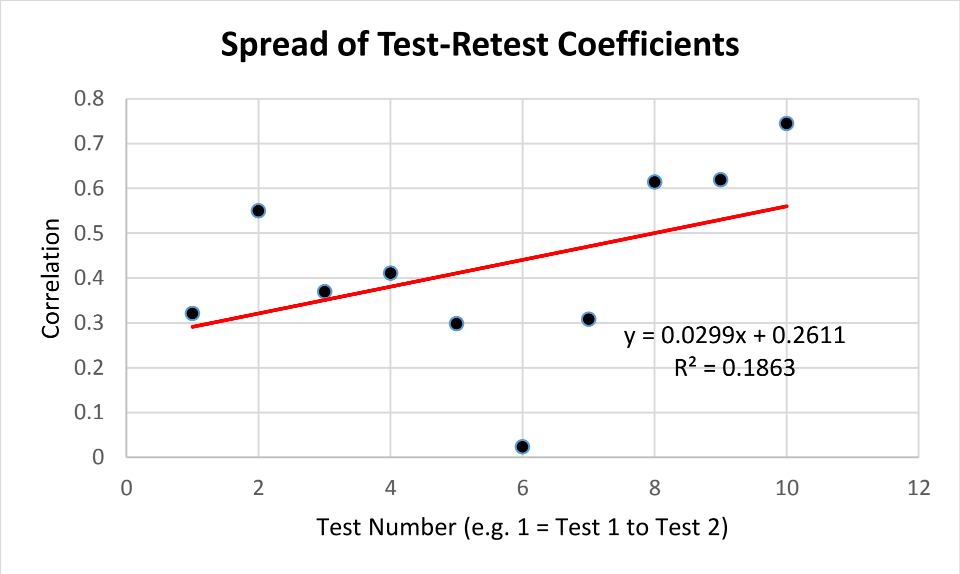


Figure 9.4 Line graph of test-retest coefficients and least squares regression line for students' attempts at the practice questions mainly involving adjacency matrices.

Also notice that unlike the correlation matrix in Figure 9.1, this correlation matrix has negative values within it. This is generally indicative of students doing worse between such attempts, but since these do not occur at the key diagonals, these will not be considered as significant. However, it is worth noting that $\frac{2}{3}$ of these negative values appear at the eighth attempts, while two other values, both notably closer to zero, appear at the fifth and sixth attempts. No students completed only eight attempts and $\frac{2}{3}$ of the negative correlations come from this attempt, which could imply that there is a great effect by not having students not performing a particular number of attempts at a test; but then, this surely takes away some responsibility from students since any studying they should be doing before the test may be performed during the test instead.

9.4.1.3.3 Invigilated Test Session Results

The correlation matrix for the results of the visual question set are given in Figure 9.5. The test-retest coefficients actually appear to decrease after increased attempts are made. The test-retest coefficient for the 4^{th} – 5^{th} tests is significantly improved, but this is most likely because only three students completed all five attempts. However, the graphical representation of these coefficients, shown in Figure 9.6, shows a more positive outlook.

	1^{st}	2^{nd}	3^{rd}	4^{th}	5^{th}
1^{st}	1	**0.290**	0.038	0.149	0.892
2^{nd}	0.290	1	**0.066**	0.121	–0.157
3^{rd}	0.038	0.066	1	**0.067**	0.538
4^{th}	0.149	0.121	0.067	1	**0.695**
5^{th}	0.892	–0.157	0.538	0.695	1
n	18	11	21	9	3

Figure 9.5 Correlation matrix for attempts made by students on test questions comprising of graphs and adjacency matrices in each question, along with the number of attempts made by a particular number of students in each case.

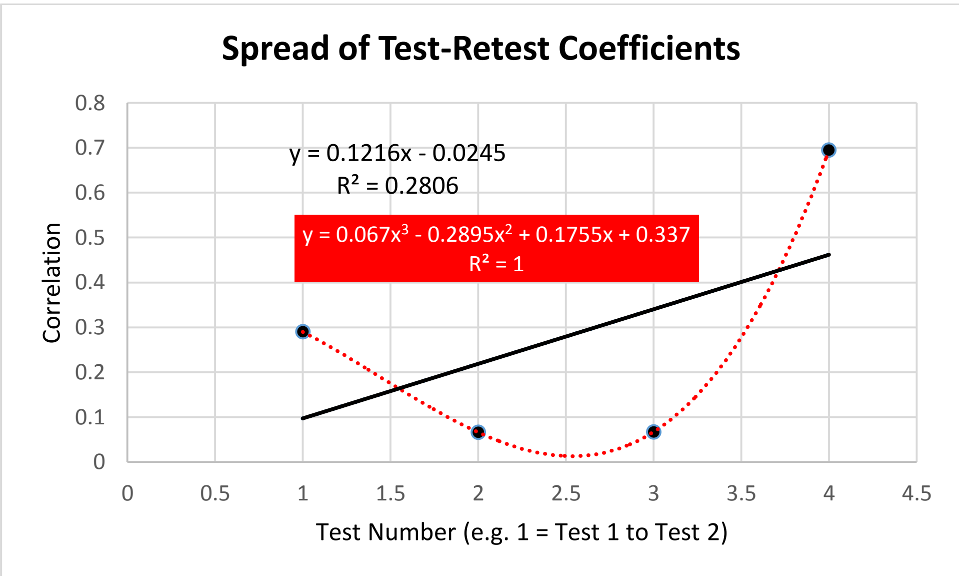


Figure 9.6 Line graph of test-retest coefficients and cubic curve of best fit (with assumption that the y-intercept is equal to zero) for students' attempts at the practice questions mainly involving adjacency matrices.

The regression line for this data is positive with all data points themselves being positive, which suggests that students were able to improve from one test attempt to the next. However, the coefficient of determination, listed at 0.2806, suggests that only 28.06% of the variability in the data can be explained by the regression line. Since there are only four points involved, a cubic curve is drawn in Figure 9.6 using all four points with absolute accuracy. Although this curve gets very close to 0 at around 2.5, it remains above 0; the minimum value is reached at approximately $x = 2.536$ with a correlation of 0.013. In fact, from the point, $x = 0$, which is a practically relevant starting point for this data, the correlation value is positive (namely 0.337) and remains positive for all positive x values; this further shows evidence of a progression of learning resulting in improved test scores from one test attempt to the next test attempt.

Since the invigilated test session was only worth 5% of the course/module mark, feedback remained attached to the answer screens during the test. Therefore, it is possible that this helped them to continue to learn and improve during the test session, which then helped them to receive a better grade overall.

9.4.2 2008 – 2014 Final Examination Analyses Considerations

The 2007 – 2008 assessment structure was never replicated, but the 2008 – 2014 assessment structure for online assessment use was replicated. Since all students enrolled in MA0422 at Brunel University had been exposed to Mathletics in the same way, it is worthwhile to consider a time series analysis of the 2008 – 2014 final examination results to investigate patterns in final examination results. Unfortunately, due to a change in syllabus in 2012, the results cannot be completely reliable. However, using an expectation of a three-year continuation period prior to another change in syllabus, a three-period time series analysis can be conducted nonetheless to explore these results further.

The chart presented in Figure 9.7 shows the three-period forecasting model for the 2008 – 2014 final examination results in MA0422.

The moving means for this period only consists of four points; therefore, it is not too difficult for a trendline to be somewhat representative of the moving means values, as shown by the R^2 value of 98.626%. However, since this research was initially conducted, data from 2015 and 2016 have updated the results to form the table in Figure 9.8. The updated data shows the trendline is still reasonably representative of the moving means values with a R^2 value of 75.538%. The positive slopes of the trendlines indicate final examination scores are increasing from one academic year to the next and it is possible that this is partially due to the continued implementation of Mathletics in MA0422.

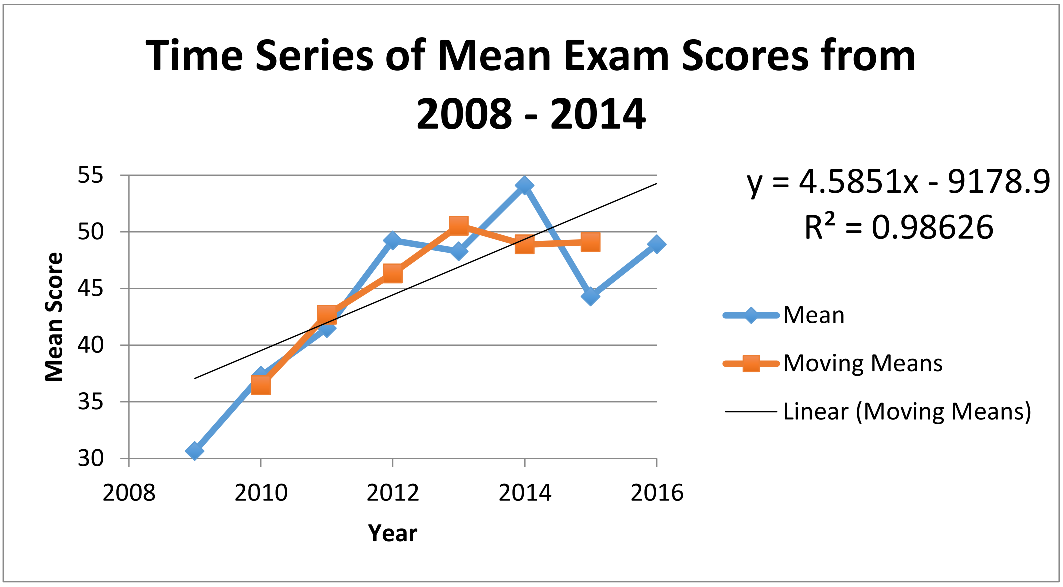


Figure 9.7 Time Series of Mean Exam Scores from 2008 – 2014.

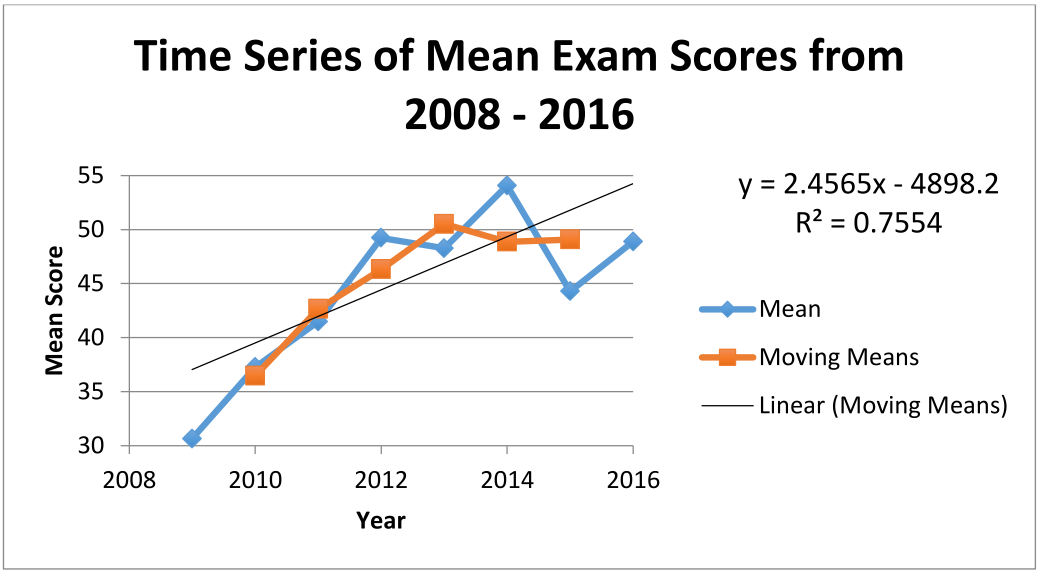


Figure 9.8 Time Series of Mean Exam Scores from 2008 – 2016.

The numbers of students sitting MA0422 final examinations at Brunel University has consistently been between 80 and 120 students during these academic years, so the numbers of students does not have a significant impact on the forecasting model. However, it ought to be expected that over time, the slope of the trendline will continue to decrease towards zero as it cannot be expected that students will achieve a mean score of 100% between all members in the cohort. Additionally, recall that there was a significant syllabus change in 2012; any future syllabus changes will create possible impacts on future results and each syllabus change needs to be considered in more detail for future consideration. Other issues, such as differences in student cohorts (e.g. ages, genders, learning backgrounds, admission criteria), may impact future results; some issues may be difficult to obtain due to ethics issues, so, although the time series analysis seems promising, it needs to be taken with some degree of caution.

9.5 Research Question: Difficulty Factors in Graph Theory Questions

This section answers the research question:

Which factors, if any, can cause an objective question in graph theory to be more difficult than other questions in the same topic?

The statistical analyses shown in Chapter 7 and Chapter 8 show that although there were some differences in assessment scores between questions involving graphs and questions involving adjacency matrices, these results were not consistent throughout the library of graph theory questions and therefore, it cannot be concluded that there are any significant differences in using graphs or adjacency matrices within graph theory questions. It is good that this is indeed the case because it will help to emphasize the importance of understanding graphs and adjacency matrices within the context of graph theory for questions in any topic within the subject.

There was not much variety in question types in the 2007 – 2008 analyses, but there was more in the 2008 – 2014 analyses. However, most differences in assessment scores appeared due to a lack of variation in the topics themselves. Nonetheless, it was shown that there was a significant increase in assessment scores between WI questions and WI + Check questions, implying that the use of the pop-up window was helpful in reminding students to verify their own answers before submitting their responses.

In the 2007 – 2008 statistical analyses, it was evident that each question within a topic was significantly different. It is good that the questions within a topic are significantly different as this provides a sense of variety between questions, which can be more appealing to a teacher or lecturer when choosing questions for an assessment. It was also evident that the use of word problems caused a significant difference in assessment scores compared to similar questions which were not given in any context. Although it would generally be preferable to remove the notion of context from questions due to the added difficulty of interpreting a real-world problem into a mathematical problem, this is not a reasonable response as it is important students understand the practicality of using mathematics outside the classroom. Test-retest coefficients for attempts made in these assessments were all positive, showing that these questions can be helpful in assisting students' progress in understanding the learning material.

The 2008 – 2014 statistical analyses both showed that all questions had positive discrimination values, implying that all questions provided some benefit to the overall assessments produced. However, for the 2011 – 2014 assessments, more questions were closer to having a discrimination value of zero, implying that these questions showed no benefit to the overall assessments. WI+Check questions performed better than WI questions, implying that students benefitted from the inclusion of the pop-up windows.

Questions from the 2008 – 2014 statistical analyses which required some prerequisite knowledge of graph theory topics (i.e. bipartite graphs and Kruskal's & Prim's algorithms) resulted with generally lower facility values than other questions. However, all questions assessed from 2008 – 2014 were scored with results of either 0 or 1, whereas the 2007 – 2008 questions used partial marking where possible. In the 2007 – 2008 statistical analyses, it was shown in Section 7.4.1 that this distinction is not as clearly evident as discrimination values vary differently. This shows promise for the inclusion of partial marking within the assessment framework, especially for responsive questions that have carefully designed distracters, which can be sought and assessed to provide a means of partial credit where available. This, however, does not discredit the use of all-or-none marking, as there may be valid reasons for imposing this strategy within an assessment (e.g. business-related assessments, where making an error could cause a significant loss of profit).

9.6 Additional Remarks about Statistical Conclusions

Statistical information was used to determine whether or not a question is suitable within assessments. These statistical analyses are generic for any assessment with QMP as facility and discrimination values can be provided directly by QMP's Assessment Manager. Various issues have been noted, including an issue with one MC question regarding the properties of a planar graph, which, as noted in Table 7.10, had a facility of 1, standard deviation of 0, and a discrimination of -1, implying that all students answered the question correctly and that it failed to be an effective question for the given assessment. Another issue that occurred was that the work produced by Zaczek, highlighted in Section 3.10, which included Word Input (WI) questions that did not have such dynamic input (i.e. fewer random parameters and no pop-up windows asking students to double check their work); it is suggested that low facility values in these questions may correspond to a lack of dynamic input and the absence of pop-up windows, resulting in formatting errors and unnecessary errors in answering questions. However, similar to hypothesis testing in statistics, where you cannot accept a hypothesis, but rather either reject it or "fail to reject" it, it is easier to determine what makes a question bad, rather than to prove whether or not a question is "good".

It is not being suggested here that further work be conducted to determine what makes a question "bad" or "good"; instead, what is being suggested is a careful consideration of the wording and structure of a question. When examination boards write their examination questions for secondary mathematics, much careful consideration is given into the wording and presentation of questions so that students are provided with ample opportunities to answer questions correctly. With the inclusion of social media, it is easy to hear of stories of seemingly "bad" questions being discussed online, like the 2015 GCSE maths question about Hannah's sweets[67] or even the entire 2016 Core Mathematics 1 (C1) maths paper from EdExcel[68]. In the case of the C1 paper, many students complained about the lack of whole numbers appearing on the non-calculator examination, but it is an expectation for students to be able to perform such calculations at this level without absolutely needing to use a calculator, but the students' argument was that past papers were not as difficult as this examination paper. Whilst conducting research at Brunel University, it was observed and noted, even directly by students, that students are much more likely to "study" past examination papers rather than actually studying the learning material taught in lectures; in other words, they would rather surface learn than deep learn[69]. Because of such study habits and the ever-changing curricula and syllabi in secondary and post-secondary education, it is necessary for questions to continually be modified and adapted to suit the new course structures. Therefore, for future consideration, it may be worth exploring, for instance, the adaptability of questions using random parameters in CAA or even the structure of question design itself within CAA.

Changes to a syllabus are expected, but will have an effect on statistical analyses, especially when forecasting models are used. Time series analysis is not therefore recommended when analysing results from one academic year to the next. However, when students are allowed to attempt questions as often as possible in practice environments, then it may be more useful to use time series analysis to determine if it is possible to notice a mastery of assessment and learning through repeated attempts at questions. Future research and experimentation could explore this more in depth, ensuring that detailed data is collected from QuestionMark Perception during attempts.

Additionally, the statistical analyses conducted were completed using Microsoft Excel. Although Excel has many features, more advanced statistical analysis applications, such as SPSS, may provide additional results and thus, provide more information leading to more detailed conclusions. At the time of conducting this research, such advanced software was not made available by the university and so, was not used. Since such software is presently more readily available, it may be possible for future consideration to use only more advanced statistical analysis software and applications to conduct similar statistical analyses.

Chapter 10 Conclusions

10.1 Answering the Research Questions

First, recall the research questions set to be answered in this thesis; the results have been paraphrased for this section, but the sections in which the questions have been fully answered are referenced:

1. *What question features exist that could change how students interact with questions?*
   From Section 3.13, it was noted that relevant features can be implemented within questions in Mathletics dealing with graph theory. Existing adjacency matrix functions and new graphs functions, with the help of SVG graphics, provided a great range of questioning. Different question types helped with question design and word problems were used and discussed as part of question design.

2. *How can the potential of computer-aided assessment be exploited to set versatile and robust questions in graph theory?*
   From Section 5.4, it was shown that Mathletics is helpful in creating an organised library of questions for graph theory. Random parameters included in all elements of a question, including the design of a graph, help to create individualised assessments. The structure of a graph was set up so that all vertices are equally spaced around a circle and embedded within an image box (or frame).

3. *Which factors, if any, can cause an objective question in graph theory to be more difficult than other questions in the same topic?*
   From Section 9.5, it was shown that although some differences in assessment scores existed, they were not consistent throughout

and therefore, it could not be concluded that there were any significant differences in using graphs or adjacency matrices within graph theory questions. More variety appeared in the 2008 – 2014 analyses than the 2007 – 2008 analyses, but most differences were due to a lack of variation in topics. The inclusion of pop-up windows in WI questions was shown to be useful as assessment scores were higher when they were included. Only one question has a negative facility value throughout and this question was immediately discarded after learning of the negative facility value. More difficult questions from the 2008 – 2014 assessments may have received lower facilities due to the all-or-none marking scheme, whereas the 2007 – 2008 assessments included partial marking, thus resulting in more favourable facility values. However, all-or-none marking schemes could be used depending on the environment in which assessments are being taken (e.g. high-risk environments, such as business administration).

10.2 Issues Arising from Research and Future Considerations Stemming from These Issues

Throughout this research, various challenges have been presented, all of which needed to be overcome in order to progress further with the research. This section will review key issues that occurred and will be detailed by reflecting back on ideas discussed earlier in this thesis.

10.2.1 Designing Random Graphs and Challenges with Mathletics

The biggest issue in this research was designing a random graph using SVG in Mathletics. It was important that dynamic graphs could be created with variables embedded so that it provided additional flexibility to the question designer when designing graphs with specific properties later; again, the issue is not how much more difficult graphs with more vertices would be for students to answer, but rather if the questions could be reasonably designed and used appropriately in online assessments. Creating a template function for a graph of n vertices with multiple variables allows the question designer to provide suitable flexibility in designing graphs for questions that were reasonable; however, this does mean that the issue of fairness lies with the question designer and so, the question designer needed to have some background knowledge about any modules for which the assessments would eventually be created. In this research, collaboration with the module lecturer helped to structure the design of questions for MA2920 and in the case of MA0422, the lecturer designed the assessments directly for his students.

Regarding the structure of the graphs, points had to be plotted in various places, but it was ideal for the points to be equally spaced apart. Section 5.2 details how this was resolved so that a random graph of n vertices would appear on the screen. Additionally, there was an issue with using too many vertices as programming errors would appear. It was eventually determined that the main issue that caused this was the dimensions of the image box in which the graphics would appear. To resolve this, various formulae were used as the values of the dimensions in order to allow graphs of larger numbers of vertices to be generated if needed. Since the functions noted in Chapter 2 have been created into a graph theory template, they can easily be called for other purposes, including the design of graphs for use in questions in other subjects, such as business administration, economics, or engineering. However, as has been highlighted in Chapter 2, if the vertices of a graph need to be altered based on their location in the image box, then new functions would need to be created; this is not a major issue in that an alteration from a copy of the graph function can suffice, but the alteration needs to follow a formulated pattern for the placement of n vertices, which may require some thought, depending on the required positioning.

This research follows on from that conducted by Ellis[29], Gill[36], and Baruah[42], who have investigated the implementation of Mathletics within other mathematics disciplines and modules, either within the Department of Mathematics or within other departments at Brunel University. Later research from Zaczek[44] does investigate a broadening of Mathletics for use in decision mathematics, but questions are not necessarily designed with as much random parameterisation as presented in graph theory and this should be explored further to better understand what challenges may occur in other topics. The series of templates created for Mathletics from all of this research provides numerous functions in multiple disciplines within mathematics, including calculus, statistics, linear algebra, and now graphs and decision mathematics. Templates provide a more general framework from which functions can then be called to create individual questions. However, the design of questions in Mathletics requires calling functions from templates and the more templates that are called, the longer it will take for questions to load. Some topics, such as graph theory, require the use of linear algebra and graphs functions, so combination templates have been created in such cases, but creating one general template could create time delays in getting questions to load effectively on the screen.

10.2.2 Random Parameterisation

A great deal of care and consideration has been taken in creating random parameters within questions to avoid unwelcoming situations from occurring, such as programming errors (e.g. dividing by zero accidentally within a lengthy calculation). Repeated attempts by numerous students throughout this research, by means of practice attempts initially, helped to uncover any errors not detected earlier in designing the questions and where any unfortunate occurrences appeared in invigilated assessments, accommodations were made so that students would not be penalised for these errors occurring; allowing students repeated attempts at invigilated assessments helped somewhat to deal with this situation. The strategies used to avoid errors from occurring follow from the previous research of Ellis[29], Gill[36], and Baruah[42], but it still has its limitations and further research into avoiding these issues was not considered for this thesis. Future research into this can explore possibilities for avoiding errors, but with millions of realisations of a single question being possible, it is unrealistic at this stage to test every single realisation. However, looking carefully at the step-by-step approaches used to solve problems (i.e. looking carefully at the approaches used in the programming) can help to look at any issues that may occur. Reverse engineering of problems may also help to avoid errors, setting questions in a specific way to bypass problematic situations from occurring. Future considerations into investigating this could provide some excellent insight for CAA and CAL from a technological perspective.

10.2.3 Efficacy of Assessment Versus Effects on Students

It was very important that this interdisciplinary research focused on the questions designed for use in online assessment and did not focus on any element referencing students' abilities in answering questions or how it could impact their learning. Questions about whether or not CAA helps students and who it benefits cannot be answered within the scope of this thesis as this would involve additional research that could be better investigated by educational experts. Of course, this is not to say that such questions have no value to this research, but it is to say that this could be better investigated through future research conducted either by educational experts or by more joint efforts between mathematics and education researchers. Some ideas have been presented in Section 9.4, using statistics to highlight how it could be shown that students may benefit from the implementation of Mathletics within their learning, but such experimentation needs to be conducted more thoroughly and with more structured assessments, providing more consistent results that can be better analysed together.

10.2.4 Creating Suitable Distracters and Feedback

Creating distracters, as shown in Chapter 4, was not necessarily difficult to manage. In Section 4.7, it was shown how it is possible to use past student errors in order to design distracters; however, it is more difficult for a third party to do this than a direct source. As highlighted in Chapter 1, examination boards are not willing to disclose examination data; this could be because this data refers to children under the age of 18, but similar cautions are being conducted in postsecondary institutions, where students are usually adults. To view such data often involves finding out who made the errors and this information is not easy to remove as it appears on the examination scripts so that it may be referenced later in case a student wanted to challenge his/her final grades. For future consideration, anyone wanting to review previous student errors to see which errors are appearing more frequently should preferably be involved directly in the assessing of any coursework for modules that cover the topics to be assessed, but in doing so, it is still expected that this would need to be cleared with an ethics committee to ensure all safeguards are considered to avoid confidential information from being released.

Relating to the distracters, the provision of feedback had to be carefully worded. In some cases, it could be possible that students would provide an incorrect answer that could be obtained as a distracter, yet used a completely different approach to come to the same answer. With the additional randomisation provided in Mathletics, this removes some of the likelihood of this occurring, yet it is not enough on its own. Some of the questions presented involved detailed calculations with answers to be given to a level of accuracy. For such questions, it is much less likely students will give an answer that could be generated by a distracter without using the assumed strategy for obtaining the distracter. However, as was shown in Chapter 4, distracters can be carefully considered so that appropriate feedback may then be provided to discuss why these distracters provide incorrect answers and in some cases, these distracters can warrant partial credit as the distracters themselves were created by slight alterations of the correct methods for answering questions. To accomplish this with random parameters, though, does require formulating distracters within question codes, almost as additional answers to be triggered so that they can then provide their own feedback if triggered by a student's response. Similar strategies regarding the creation of distracters and additional feedback can be used in other subjects, but some additional care may need to be taken since mathematics can easily take advantage of generic formulations of distracters and feedback, whereas other subjects may sometimes have less formulated distracters to use. However, it is worth noting that since questions designed for graph theory do make use of current linear algebra templates, it may be easier to test for expansion of the strategies and methodologies used from this thesis in elementary algebra and calculus and then into statistics, especially where some advanced statistics makes use of calculus (e.g. continuous random variables).

10.3 Limitations and Recommendations

10.3.1 Technical and Programming Improvements

10.3.1.1 Including Computer-Aided Learning and Issues with Current Versions of Questionmark Perception

In Chapter 1, it was noted that this research would focus specifically on the creation of CAA questions in graph theory using Questionmark Perception (QMP) and Mathletics to provide a versatile and robust library of questions for assessment purposes. This research did not focus on computer-aided learning (CAL) as the software was primarily used as an assessment tool. However, it may be possible that software exists which uses CAL to help students learn more about graph theory. It would make for very interesting research to see how to use the tools shown in this thesis to develop a new CAL tool for graph theory, especially if randomisation and SVG graphics (or a similar graphics tool) could be used to individualise the learning in some way.

However, this also leads to the current dilemma with QMP in that newer versions make it impossible to program randomised parameters into the coding of questions to suit specific needs. The older version of QMP is currently being used at Brunel University, but it was noted by Gwynllyw and Henderson[24] that these changes made it so difficult for them to use QMP that they instead created DEWIS. To continue this kind of research in the future, it may be necessary to use a system like DEWIS or *maths e.g.* to assess and analyse questions.

Following from previous suggestions, one good place to start would be A-level Decision Mathematics in the U.K., which teaches graphs and networks, critical path analysis, and matchings as part of the module syllabus. Current tools, like MyMaths[70], do provide some good CAL on decision mathematics, but questions do not include random parameterisation, so repeated attempts will often involve the same questions.

However, this then raises an issue in whether or not repeated attempts with the same questions reinforces learning better than repeated attempts with similar, yet technically different questions. It may be argued that there is more confidence in revisiting the same question until it is mastered before approaching different questions, but it may also be argued that key learning skills involved in answering questions are not being maintained from revisiting the same question. A comparable study into CAL could investigate this more thoroughly, investigating the process of students looking at the same question and an identical number, perhaps the same students, looking at similar, yet technically different questions to determine what impacts occur within their learning.

10.3.1.2 Dealing with Distracters

From Section 4.8, although distracters can be categorised, it is preferable to deal with distracters individually within questions as similar distracters would still warrant individual attention and additional feedback. However, a categorisation of errors may help to establish a “framework of errors” from which it can be better understood which types of errors are occurring more regularly. Errors occurring in graph theory are not necessarily different from that of other mathematical topics, such as statistics, calculus, linear algebra, and mechanics, all of which have previously been researched for Mathletics. Furthermore, where learning material does overlap into other fields, such as economics, business, and engineering, additional consideration can be provided to review the categorisation and frequency of errors between different groups of students.

To investigate the appropriateness of having a framework of errors, previous research may be considered, but an in-depth look at created errors needs to be considered, which would involve looking at previous attempts on questions; this requires the approval of an ethics committee to begin investigating this. Furthermore, to look at framework design would require looking at numerous attempts to establish categories with some confidence. Additionally, how errors are to be categorised may come into dispute; for instance, a calculation error to one person may be seen as a methodology error to another person and so, some careful planning into defining each error type is also required.

Distracters considered for designing questions on spanning trees reviewed the algorithm for calculating the number of spanning trees and some properties of spanning trees in search for them in a graph. An analysis of past examination papers at Brunel University, as provided in Appendix D, provide a more detailed list of errors students made during examinations, but these were not observed until after the CAA questions were designed. The design of distracters can use theory to help determine what errors students may make, but it is preferable to have a direct look at previous student attempts to see which errors they actually are making in order to program these into the question coding. Although student errors were considered elsewhere in the design of questions and the MA2920 examinations only had two questions on the subject of graph theory, these final examination papers have been helpful and so, for future consideration, a more thorough investigation into students' attempts and errors should provide for more efficient distracters with which to provide better feedback and possibly also award some partial credit for the question.

10.3.2 Development of Further Topics

10.3.2.1 Covering More Topics in Graph Theory

The library of questions developed does not cover all topics in graph theory, but rather mostly those that are taught at the post-secondary level. Other post-secondary institutions may teach different topics, so more questions should be developed within Mathletics. Additionally, networks are used in other disciplines, including engineering, business studies, and computer science. Revisiting the algorithm for producing graphs and improving it for use in topics related to other subjects will provide numerous additional opportunities, especially in the secondary sector, where the Decision Mathematics syllabi between the examination boards will often discuss business-related topics, such as critical path analysis[86]. Some of the created questions were not assessed and analysed in this thesis, especially questions on the topics of isomorphisms and graph colouring. The current syllabus for MA2726[7] and the previous MA2920[43] only briefly covered the topic of isomorphisms without formally assessing students' understanding of this topic. The topic of graph colouring had not been taught at the time the 2007 – 2008 assessments took place, so these questions were not included in the assessments and analyses. Unfortunately, the Department of Mathematical Sciences at Brunel University has since chose not to assess students using Mathletics for this upper-level module and therefore, no testing has yet to take place. However, it would be worthwhile to find opportunities to make use of these questions for students to practise their understanding and application of this advanced topic within the subject.

10.3.2.2 Secondary Mathematics

For secondary mathematics, this research may prove useful for future consideration in decision mathematics. A-level decision mathematics investigates graphs and networks throughout D1 and D2. Making use of CAA may provide a means of interaction in class through online assessments and practice so that students may work more intently with different graphical structures for practical purposes relating to topics, such as critical path analysis and matchings. Examination boards investigate students' answers to review which common errors were made in final examinations and discuss these in their examiners' reports, but having this extra resource could provide them with opportunities to highlight these issues to provide students with more opportunities to avoid common errors and provide distracter answers; this could, in turn, be implemented in other mathematics modules so that students may be given better opportunities to perform better in their studies.

10.3.2.3 Methodology for Question Design

To design versatile and robust graph theory questions in CAA with randomised parameterisation does require some in-depth knowledge and understanding about the subject itself and where graph theory is typically taught as a stand-alone module in post-secondary institutions, question designers must be knowledgeable about the content and theory of the learning material along with the design of questions and programming skills. The theoretical approaches used in the creation of questions and distracters, although helpful, was clearly not complete. For planar graphs, it is sufficient to simply look for K_5 and $K_{3,3}$ subgraphs, but for other topics, the theories used are likely just a sample of what is available. Other theories may provide more opportunities for more responsive questions and detailed feedback within questions.

The methodology for designing questions can have some good insight for designing questions in other subjects. However, it is important to remember that all questions must be objective, so any subjective questioning must be avoided. Also, the use of distracters can be helpful in other subjects, too, but formulating distracters in a way similar to mathematics, where formulae are used to generalize distracters, may prove to be rather difficult in some cases.

To design versatile and robust graph theory questions for other disciplines, it is helpful for question designers to be working closely with the other disciplines, discussing in depth how learning material is presented, how topics are discussed, and what learning objectives students are expected to accomplish. Minor differences in learning and teaching approaches can impact how questions are designed, so it is very important for question designers to investigate this more thoroughly with other disciplines. Some universities structure their programmes so that each subject is taught by those best able to teach the learning material effectively; for instance, engineering students needing to take a module on differential equations would be taught by a mathematics lecturer rather than by an engineering lecturer. However, this is not the case in every university and this is unfortunate in this case as such strategies may provide more opportunities for interdisciplinary collaborations to promote better learning and teaching strategies in post-secondary environments.

10.3.3 Improvements in Assessment Structure and Pedagogy

This section looks at some educational theories which could be considered for future research, relating them to CAA.

10.3.3.1 Bloom's Taxonomy

Some basic consideration into Bloom's Taxonomy[71] was given in the design of questions for this thesis. Originally created by B. S. Bloom in 1956[72], this taxonomy was modified in 2002[73] with two sub-models, namely a *Knowledge dimension* and a *Cognitive Process dimension*; the Cognitive Process dimension of the updated design is shown in Figure 10.1.

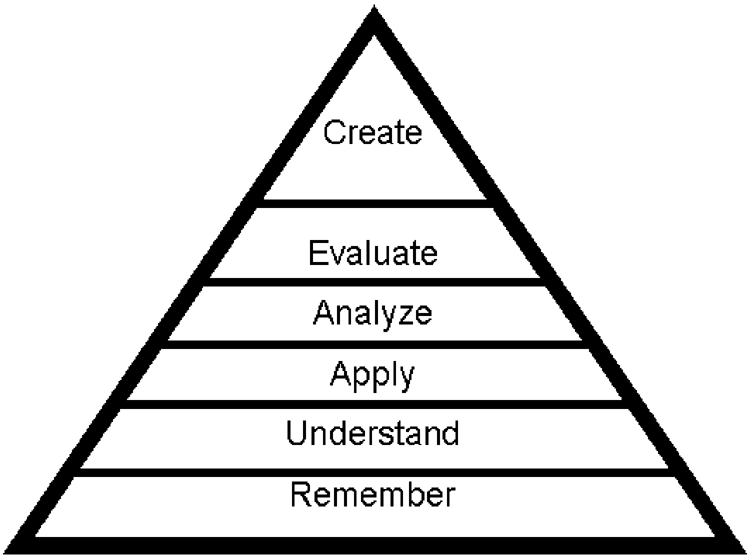


Figure 10.1 Revised version of Bloom's Taxonomy (updated 2002).

Recall that Smith, Wood, Coupland, and Stephenson[74] suggest an alteration to the typical model, called the Mathematical Assessment Task Hierarchy (MATH); this is shown in Figure 1.2. In this figure, assessments are structured so that assessments may be better structured at the postsecondary level with questions moving from one group of questioning to another group. They also state that the point of their research is to investigate the *nature* of the activity within the questions, not the *difficulty*, as difficulty may be subjectively measured depending on each student's view of understanding *what is difficult*.

This model is helpful to show the importance of question design within postsecondary mathematics, to which the research in this thesis has been aimed. However, as the authors also point out themselves, their hierarchy is *loose* in that the same questions can appear in different locations of their hierarchy depending on the students' learning skills; the example they provided was in proving a theorem, where one student who has never seen the theorem before and proves it, could be demonstrating *application in new situations*, but being asked to do so again later may change the skill to *factual knowledge* by recalling the proof (s)he created earlier. Also, these terms differ from the original structure of Bloom's Taxonomy, which may create some initial confusion for any teacher or lecturer who is already familiar with the original taxonomy.

Research from Baruah and Hatt[75] investigated the structuring of Bloom's Taxonomy to determine a more effective approach for understanding question design, coming up with a reconstructed pyramidal model, as shown in Figure 10.2.

This model gives some good insight into how questions can be structured for different target audiences and especially for students at the postsecondary level; for example, designing effective questions in analysis and evaluation would be helpful in challenging students further in their understanding and appreciation of the topics tested. Most questions that have been designed in Mathletics for use in graph theory focus on the Student Section of Bloom's Taxonomy, focusing on the understanding and application of graph theory; some questions will also show potential for analysis or evaluation within the question. Tags can be constructed within Mathletics, which will help a teacher or lecturer to design questions with a suitable mix of characteristics, depending on preferences.

In CAA, the marking scheme is important in question design due to the possibility of including specific feedback if a particular incorrect answer is given. It is simple enough to allow a question to have an *all-or-none* marking scheme, i.e. a correct answer is worth one mark and an incorrect answer is worth zero marks. However, as a tool to also be used in CAA and especially for use in higher levels of mathematics, Mathletics can provide marking schemes that are better structured to fit in with a typical assessment scheme, analysing student answers in more detail. Looking at the assessment of distracters can be helpful in this case, but numerous distracters would require much coding and unfortunately, with CAA, it is currently impossible to get students to submit workings out online in a fashion that can be scrutinised using technology, but that is not to say it is impossible to provide a different marking scheme that could allow for a fairer analysis of students' answers; for instance, DEWIS uses a remarking scheme, where every answer a student submits is "flagged" during the evaluation process[24].

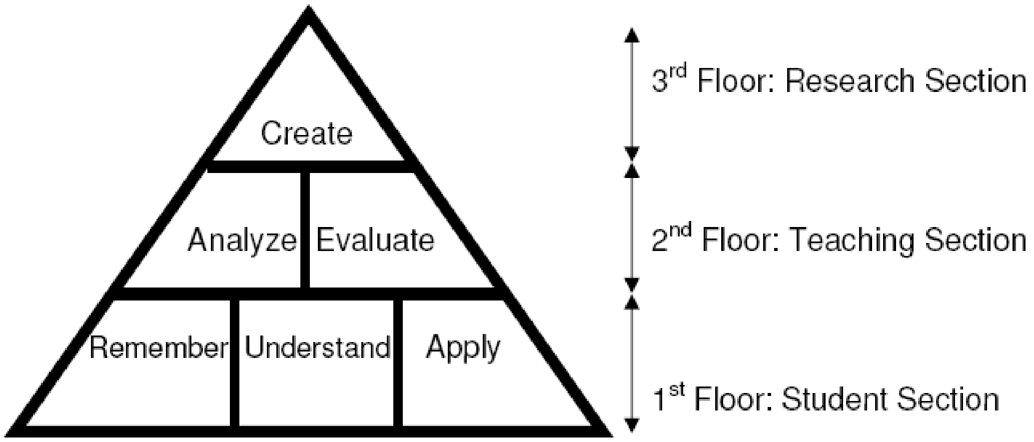


Figure 10.2 The Reconstructed Pyramidal Model of Bloom's Taxonomy for Mathematics.

Bloom's Taxonomy was helpful from a background perspective in that it helped to write structurally sound questions that could compel students to apply knowledge or analyse information within questions; this was rather important to remember as less challenging questions may not have provided valuable information about the quality of question design within graph theory using Mathletics. However, exploring taxonomies does open numerous possibilities for exploring educational theories within online assessment of mathematics. Exploring educational theories in such depth would allow educational researchers opportunities to explore mathematics education using online learning and could provide excellent interdisciplinary opportunities between mathematicians and educationalists.

10.3.3.2 Gardner's Multiple Intelligences

One other consideration could be Gardner's Multiple Intelligences because of visual and logical / mathematical components used in graph theory. Although differences between questions with graphs and questions with adjacency matrices were not as significant throughout all assessments, if future research shows that there could be more significant differences, then this could lead to further research into the consideration and implementation of multiple intelligences within question design for CAA, especially in applied subjects, such as business studies, economics, and engineering. For applied subjects, it would be worth investigating how students answer mathematical questions differently to other applied students and to mathematics students before continuing to explore the implementation of Gardner's Multiple Intelligences; the reason for this being that any differences may highlight advantageous uses for some students (e.g. engineering students may prefer logical / mathematical questions, whereas economics students may prefer visual questions), which may then be implemented within the design of questions for these particular students.

10.3.4 Considerations about Students Using Mathletics

Various characteristics about students, their accessibility to the software, and their opinions on the use of the software, among many other things, were not considered as they were not necessary for the research conducted in this thesis. Student input could be very valuable in providing additional insight on how to improve Mathletics and the graph theory questions, so this should be considered in future research in CAA and/or CAL.

Different groups of students may also have been able to perform better in their final examinations as a result of having Mathletics being made available to them; it would definitely be worth reviewing past examination scripts in future analyses so long as Mathletics is being used in a reasonable capacity as part of a mathematics module. The statistical analyses presented in this thesis does show it is possible that the implementation of Mathletics in graph theory modules could help improve students' overall understanding of learning, leading to better assessment results in their final examinations. Data from cohorts of students who do not use Mathletics could be analysed prior to Mathletics being implemented and full-scale, long-term analyses being conducted. However, this does involve looking at confidential student data and results, meaning that investigating such data would require the approval of an ethics committee at the university.

10.3.5 Structure of Questions

The research and analyses conducted showed that word problems had significantly different facility values than questions not written in a real-world context. Word problems in context are very important in helping students to understand practically the usefulness of the learning material. It was encouraging that question facility values were significantly different, but more research into the use of word problems in context in CAA may provide some further insight into question design, especially with the possibility of including CAA and graph theory into other subjects. A similar strategy to that presented in this thesis could be used, where similar questions could be presented, one without context and others in context. The results of similar questions could be compared to determine what factors exist in understanding how students respond to the different structures of questions.

Responsive questions can provide more detailed feedback if it is assumed that a particular distracter has been implemented in answering a question. Not all questions designed in Mathletics are responsive and this is the case with the graph theory questions, as well. Therefore, it would be preferable for future research to consider modifying any non-responsive questions to responsive questions for the benefit of students attempting to answer them. Also, WI+Check questions had significantly higher facility values than WI questions for the 2008 – 2014 assessments, as was shown in Chapter 8. Not all WI questions in the graph theory set have pop-up windows that alert students to double-check their answers; this is the case for questions on Kruskal’s and Prim’s algorithms, which had lower facility values. Future WI questions to be designed should have the available pop-up windows to remind students to double-check their own work prior to submitting it.

Questions on Kruskal’s and Prim’s algorithms also had low discrimination values. Recall these questions used the same graph each time, but with different weights assigned. The lack of available randomisation within the graphs may have been enough to cause some students to have noted the repetition, thus compelling them to look for patterns in answering questions rather than using the proper methods to solve problems. More research is needed to optimise the potential of graphs when designing questions in Kruskal’s and Prim’s algorithms to minimise the chances of students answering questions without using the relevant learning material.

10.3.6 Structure of Assessment Strategy

The two assessments analysed involved different assessment schemes, namely one with partial marking included and one with all-or-none marking included. The partial marking scheme appeared to provide a better understanding of a student's ability to understand and apply knowledge to more challenging problems as a partial score could be awarded if a distracter was triggered. However, this is only speculative and needs more research to verify this claim. Also, the 2008 – 2014 assessments had to be split into two separate assessments as different topics were covered from 2011 onwards. The data was helpful, but where overlaps in questions occurred, it was still unreasonable to look at all six years' worth of questions in the analyses due to discrimination values relating to the modified assessments. Changes are inevitable over time within modules, but the research conducted became more challenging as a result of this particular change, especially as bipartite graph questions were being replaced with newly designed questions on Prim's and Kruskal's algorithms. Additionally, although raw test data was retrievable for the 2007 – 2008 assessments, they were not retrievable for the 2008 – 2014 assessments; this made it impossible to determine some statistical values, including test-retest coefficients and numbers of attempts made by which numbers of students. A different statistical test had to also be used for the 2008 – 2014 analysis because of the assessment scheme implemented by the lecturer.

10.4 In Summary

This research has provided a library of graph theory questions for use in CAA that are versatile and robust, using random parameterisation and SVG graphics to create numerous realisations of the same questions. Questions are tagged using difficulty levels to categorise questions outside of their subject structure, and they are also organised based on the subject structure to make searching for topics easier. The use of different question types provides a good variety of questions to be answered. Different strategies for assessing questions can be implemented depending on the preferences of teachers or lecturers. Availability of statistical analyses provided by QuestionMark Perception allows for detailed information to be made readily available to teachers and lecturers to better understand their students' ability to understand the learning material. The findings of the statistical analyses showed promise as the variety of questions, question types, wordings, and topics is providing excellent versatility and robustness. There is clearly more work to be done to improve this work further, but this is an encouraging and welcoming beginning to a wider range of research in CAA, which can stretch beyond mathematics into other disciplines. It will be very interesting to see what future research will bring to this subject.

Appendix A Content of Topics in Graph Theory

This appendix will discuss the relevant content involved in the graph theory topics that were visited for the assessments created in this thesis.

A.1 Degree

When viewing a graph, it is obviously important to know which edges are connecting which vertices. Some vertices may not be connected to anything at all, whereas a pair of vertices may be connected to each other more than once. Although they seem basic to graphs in general, knowing the properties of these vertices is essential for understanding the nature of a graph. One concept for understanding this is known as the **order** or **degree**[76]:

Def. A.1 The **degree** of a vertex is the number of edges joined to a vertex.

Def. A.2 An **adjacency matrix** is an $\boldsymbol{n} \times \boldsymbol{n}$ matrix, $\boldsymbol{A}$, which represents a graph of $\boldsymbol{n}$ vertices such that each entry, $\boldsymbol{A_{i,j}}, \boldsymbol{1 \leq i, j \leq n}$, represents the number of edges joining from vertex $\boldsymbol{i}$ to vertex $\boldsymbol{j}$.

For directed graphs, however, the concept of order is more complex because each edge will have a particular direction associated to it; this is detailed with the following definitions[77]:

Def. A.3 The number of edges arriving at a vertex is known as the **indegree**.

Def. A.4 The number of edges departing from a vertex is known as the **outdegree**.

Def. A.5 A **weighted** (or **network** or **distance**) **matrix** is like an $n \times n$ adjacency matrix, A, but it has values given by the weight of the edge rather than the number of edges for all $\boldsymbol{a_{i,j}}$, where $\boldsymbol{1 \leq i, j \leq n}$.

This does not cope with non-simple graphs (i.e. those with loops and/or multiple edges between vertices).

To create objective questions from this topic, very specific details need to be investigated. For instance, the degree, indegree, or outdegree of a particular vertex may be determined. Also, the sum of values for a particular row or column may be asked, which could be different if a network matrix is used in the question. Such items can be objectively tested as there can only be one correct solution in each case.

A.2 Adjacency Matrices

Although a visual representation can be more beneficial to many students, especially those struggling in mathematics, the numerical representation of the visualization must also be made to help students progress further in this subject. In graph theory, the best way to achieve this is through the creation of **adjacency matrices**, which can be modified, according to particular needs, such as the number of connections, any connections to *or* from vertices.

Adjacency matrices are very helpful in determining the nature of a graph through any patterns that can be seen in the matrix. For instance, if $a_{1,3} = 0$, but $a_{3,1} = 1$, then it is ‘obvious’ that the adjacency matrix is indicative of a **directed graph** (or **digraph**) as an edge is going from vertex, 3, to vertex, 1, but nothing is going in the opposite direction. If an adjacency matrix is symmetric, then it is very possible that the corresponding graph is undirected, but even this is not a guarantee. However, if the adjacency matrix is not symmetrical, then the adjacency matrix represents a digraph.

A.3 Edge and Vertex Sets

Adjacency matrices help to determine the shape and appearance of a graph. However, it takes the edges and the vertices to make the graph itself rather than a compilation of values inside a matrix; the edges and vertices make the physical structure appear. Therefore, some consideration needs to be taken towards the individual pieces that make up these sometimes puzzling graphs.

Vertices are the essential items to holding a graph together since they act as meeting points for the ends of the line segments. They can be moved in different ways to reshape graphs and any number of edges can connect each vertex. Also, different graphs have different numbers of vertices and sometimes, graphs with equal numbers of vertices will have different labels on the vertices, depending upon the application of the graphs. Therefore, there needs to be a way of illustrating the vertices together in a set.

Similarly, the edges are crucial because they determine the final appearance of a graph, so there also needs to be a way to illustrate the edges in a set. The best method for doing this is to introduce **set notation**, which includes all elements of a group to be listed and contained within a curly set of parentheses, namely { }. When using this notation, though, students should practice listing all elements in a particular order, such as increasing, numeric order or alphabetical order. By doing this, they become more organised in the presentation of their work, which might help them to develop a habit of being more organised in other ways, too.

A.4 Simple and Connected Graphs

The previous sections dealt with the understanding of graphs and their individual properties. In the first section, degree was mentioned, including the key terms of degree, indegree, and outdegree. The second section dealt with adjacency matrices and how it is possible to use them to create graphs. The third section dealt with edges and the vertices and how to represent each separately as part of a graph. However, this section begins to look at the graphs as a whole to see what types of graphs can be created and what properties each exhibit.

In order to understand the two main types of graphs in this section, the following definitions are to be implemented[77]:

Def. A.6 A **path** is a finite sequence of edges such that the end vertex of one edge in the sequence is the start vertex of the next edge in the sequence.

Def. A.7 A **simple graph** is a graph with no loops.

Def. A.8 A **connected graph** is a graph where there exists a path from any vertex to any other vertex.

To solve questions regarding these types of graphs, basic understanding of the graph types is required. In a simple and connected graph, a loop between one or two vertices cannot exist, but cycles involving three or more vertices may exist. For a connected graph, each vertex must have an edge that connects to a different vertex and furthermore, a connected graph cannot be formed from disjoint graphs.

A.5 Hamiltonian and Eulerian Cycles

As noted in the previous chapter, cycles can appear in simple and connected graphs. However, some cycles have special characteristics that distinguish them from other cycles. Other interesting facts about these cycles involve the methods by which they were first introduced.

Sir William Rowan Hamilton (1857) posed a problem through an "Icosian game" he introduced, where players had to find various paths and cycles, including spanning cycles, of the regular dodecahedron[78]. Two years prior to this, though, Thomas Penyngton Kirkman posed the question directly and even more generally than Hamilton, but it was Hamilton's game that, although unsuccessful commercially, became popular mathematically and thus, the **Hamilton cycle** was introduced.

In 1736, Euler worked on a famous problem involving the seven bridges of **Königsberg**, known today as Kaliningrad, an exclave of Russia surrounded by Lithuania and Poland[79]. The seven bridges connected four landmasses and Euler wanted to determine whether or not it was possible to walk over all seven bridges once and only once, with the walk starting and ending on the same landmass.

This problem became so popular that it can be coined as the *birth certificate* for graph theory and through this, the **Eulerian cycle** was introduced[80].

Based on these historical accounts, the following definitions can be made:

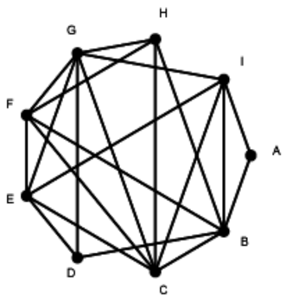
Def. A.9 A **Hamiltonian cycle** is a cycle where every vertex is visited exactly once (Recall that a cycle refers to a path that ends at its starting vertex.).

Def. A.10 A graph containing a Hamiltonian cycle is known as a **Hamiltonian graph**.

Hamiltonian graphs have many properties and many theorems have been created involving these graphs. Some of these theorems will be mentioned later when discussing the generated questions.

The following example maps out a Hamiltonian cycle within a graph.

Example A.1 Find a Hamiltonian cycle within the graph shown below.



Some strategy is needed in order to determine a Hamiltonian cycle within a graph. In this example, the best idea is to start with vertex, A, because it has degree, 2, and thus, the end of the Hamiltonian cycle can easily be determined. If starting with $\overrightarrow{AB}$, then create a list of vertices so that each is visited only once.

Solution: *One possible solution is* $A \to B \to C \to H \to F \to G \to D \to E \to I \to A$. *However, also note that the route,* $A \to B \to C \to H \to F \to G \to D \to E \to I$, *is a path that includes all of the vertices; this is known as a **Hamiltonian path**.* □

Following from a cycle joining all vertices is a cycle joining all edges.

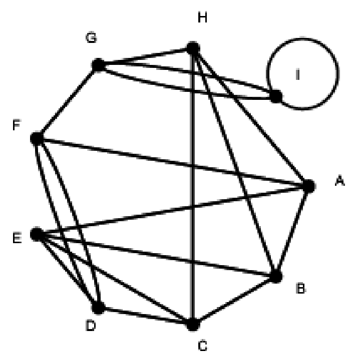
Def. A.11 An **Eulerian cycle** is a cycle that uses each edge of the graph exactly once[80].

Def. A.12 A graph containing an Eulerian cycle is called an **Eulerian graph**[80].

Eulerian graphs have many other terms associated with them, along with some algorithms for either creating Eulerian cycles or for extracting them from graphs. One main property of Eulerian graphs is that the degrees of every vertex are even; this property will be useful in understanding the design of the questions for this section.

The following example maps out an Eulerian cycle within a graph.

Example A.2 Find an Eulerian cycle in the graph shown below.



A lot of strategy is needed in order to find an Eulerian cycle. First, choose a starting vertex, which is to become your end vertex later, as well. From there, draw a path from one vertex to another, using each edge only once. In the case of a loop, it may be preferred to make use of it all at once; for instance, although not always the best strategy, if choosing $\overline{DF}$, then immediately use $\overline{FD}$ as well.

Solution: One possible solution is:

$$\boxed{\begin{array}{l} A \to B \to C \to D \to F \to D \to E \to A \to F \to \\ G \to I \to I \to G \to H \to C \to E \to B \to H \to A \end{array}}.$$

□

A.6 Isomorphisms

The graphs generated using Mathletics all have one main property in that they are all formed in a cyclic pattern. However, not all graphs are drawn in this fashion. For example, ladder graphs are normally drawn as two rows of vertices with edges connecting them to form a distinctive ladder shape. Also, wheel graphs have an additional vertex in the centre, away from the cyclic pattern, which creates a more distinctive wheel shape. Obviously, though, doing this requires specific graphing functions for each graph, which may not be necessary. However, the graphs generated using Mathletics are still ladder graphs, wheel graphs, and so on… it's just that the vertices have moved to different locations. Such graphs, where the features of corresponding vertices are similar, are known as isomorphisms. A formal, mathematical definition of an isomorphism is given, but first, it is necessary to recall the following definitions[81]:

Def. A.13 An **injection** is a mapping, $F : X \to Y$, such that for all $x_1, x_2 \in X$, if $F(x_1) = F(x_2)$, then $x_1 = x_2$.

Def. A.14 A **surjection** is a mapping, $F : X \to Y$, such that for all $y \in Y$, there exists an $x \in X$ such that $F(x) = y$.

Def. A.15 A **bijection** is a mapping, $F : X \to Y$, that is both injective and surjective.

Using the definition of bijection, the definition[80] of an isomorphism can now be given.

Def. A.16 An **isomorphism** between two graphs, *G* and *H*, is a bijection of vertices, $f(V): V_G \to V_H$, and also a bijection of edges, $f(E): E_G \to E_H$, such that for all $u, v \in V_G$, the set of edges, $\{e_{u,v}\} \subseteq E_G$, is a bijection to the set of edges, $\{e_{f(u),f(v)}\} \subseteq E_H$.

Formally, for two graphs, say $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, an isomorphism requires a *bijective* mapping of vertices, $\phi : V_1 \to V_2$, where, for any two vertices, $x_1, x_2 \in V_1$, there is an edge, $\overline{x_1 x_2}$, in G_1 if and only if there is an edge, $\overline{\phi(x)\phi(y)}$, in G_2[82] For the problems that will be seen, the vertices in the mapping, G_2, which correspond to the vertices in G_1 will most likely not be in a fashionable order, such as $\{x_n, x_{n-1}, \ldots, x_1\}$. This, however, will challenge students further to understand the patterns within special graph types so that they may be able to distinguish between different graphs.

There are six questions for this section. As with the previous section, there are questions based on graphs and identical questions based on adjacency matrices. However, in this section, special graphs are used and as such, three of the questions are identical to the other three questions, but are made more difficult by removing detailed information about the graph types.

A.7 Bipartite Graphs

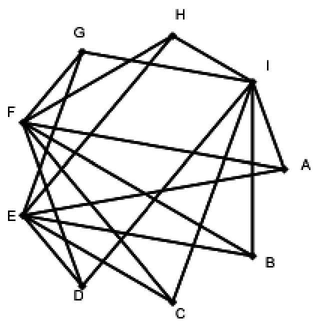


Figure A.1 A bipartite graph with partitions, $V_1 = \{A, B, C, D, G, H\}$ and $V_2 = \{E, F, I\}$.

Throughout this chapter, many properties and characteristics of graphs in graph theory have been mentioned, including details about vertices and edges, different types of special cycles within graphs, and, most recently, similarities between graphs through isomorphisms. Also, some different types of graphs have been mentioned, like simple and connected graphs, and Hamiltonian and Eulerian graphs, which come about by having Hamiltonian or Eulerian cycles respectively. In this section, another type of graph will be introduced.

Many graphs, as have been seen earlier in this chapter, can exhibit special properties. Another type of graph that does this is known as a bipartite graph, which is defined as follows:

Def. A.17 A **bipartite graph**[80] is a graph with vertices, *V*, partitioned as $\boldsymbol{V = V_1 \cup V_2}$, such that all edges are of the form, $\overline{xy}$, where $x \in V_1$ and $y \in V_2$.

An example of such a graph is shown in Figure 3.24.

A.8 Planar Graphs

In the last section, questions revolving around bipartite graphs were given. These graphs are applicable in matching problems, such as matching different sources to a different number of houses (known as a **utility graph**)[48]. However, in this section, one particular type of bipartite graph, namely the complete, bipartite graph, $K_{3,3}$ will be used based on a special property it holds. Similarly, a complete graph, known as K_5, will be used based on the same property and both graphs will help to illustrate the key behind the next type of graph to be shown.

Def. A.18 A **planar graph** is a graph that can be drawn in the plane with its edges connecting *only* at the vertices of the graph (i.e. *no intersections* between any two lines).[48]

In order to determine whether such a drawing exists, though, can be quite difficult if the number of vertices is large. However, in 1930, Polish mathematician, Kasimir Kuratowski, successfully proved a theorem for determining the planarity of a graph, which involves looking at subsets of the graph in question.[48]

Before stating this key theorem, a pair of definitions are first needed:

Def. A.19 A **subdivision** of a graph, G, is a graph, say G', that can be obtained by adding a new vertex to the middle of edge in a subset of edges in G.

Def. A.20 Two graphs, G_1 and G_2, are said to be **homeomorphic** if there is an isomorphism from a subdivision of G_1 to a subdivision of G_2[80].

Please note that all internal vertices of the paths in any subdivision, G', have degree, 2, since they do not intersect any other paths.

Now, using these definitions, along with the two special graphs mentioned at the start of this section, Kuratowski's Theorem can be given.

Theorem A.1 A simple graph is planar if and only if it does not contain a subgraph that is homeomorphic to either the complete graph, K_5, or the complete bipartite graph, $K_{3,3}$[80].

A.9 Spanning Trees

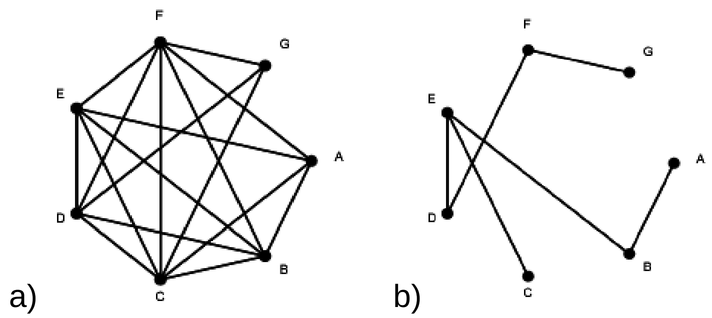


Figure A.2 A graph, G, as shown in (a). The graph shown in (b) is a spanning tree for G.

Graphs can provide a lot of information when they are applied to particular, real-world situations. One essential element in many cases for such graphs is connectedness, but with so much information provided in one graph, it can be necessary to decompose the graph into a subgraph, but while still maintaining connectedness. A good strategy for doing this would be to create a subgraph that has no cycles, but is still connected. Using this strategy, two more definitions are used[83]:

Def. A.21 A graph is a **tree** if and only if it is connected and has no cycles.

Def. A.22 A **spanning tree** for a graph, G, is a subgraph of G that contains every vertex in G and is a tree.

The graph shown in a appears to have many potential spanning trees since there are numerous edges within it; in fact, there are 3,612 possible spanning trees in this graph. Determining the number of spanning trees, though, does not necessarily require any computer system to search for them, but rather a mathematical procedure involving some knowledge in linear algebra. The theorem that generalises this was created by Gustav Robert Kirchhoff[84], a German physicist born in Königsberg, now known as Kaliningrad, Russia… and also known as the setting for the Euler's famous problem on the seven bridges of Königsberg, as detailed in Section 3.5. Kirchhoff's theorem can be given with much detail[83], but is simplified for specific use within introductory linear algebra. However, to state the theorem, one definition[80] needs to be given:

Def. A.23 A **degree** (or **valency**) **matrix** is a diagonal matrix whose entries, $a_{i,i}$, correspond to the degree of the i^{th} vertex of a graph, G.

Now, Kirchhoff's theorem for determining the number of spanning trees of a graph, known as the **Matrix Tree Theorem**[80], may be given.

Theorem A.2 The number of distinct spanning trees of a graph, G, is the absolute value of any cofactor of the difference of the corresponding degree matrix and the corresponding adjacency matrix.

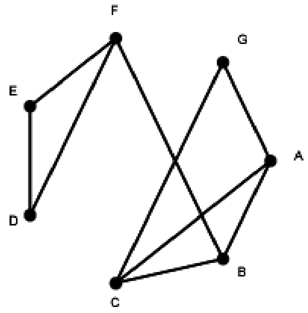
This theorem can be used to prove that there are indeed 3,612 distinct spanning trees for the graph given in a.

Some graphs have special properties that help when determining the number of different spanning trees there are in them. One such property involves the following definition[85]:

Def. A.24 For a graph, G, a **bridge** is any edge such that its removal causes the graph to be disconnected.

For a graph, G, with any number of bridges, since each bridge connects two disjoint subgraphs and therefore, is necessary for creating any spanning tree, determining the number of spanning trees in G can be reduced to first determining the number of spanning trees in each disjoint subgraph of G and then, using combinatorics, multiplying the numbers of spanning trees in each subgraph together to obtain the result. An example of this is given in Example A.3. In this example, notice how the inclusion of a bridge significantly reduces the amount of work to be performed.

Example A.3 Determine the number of spanning trees in the following graph:



Solution: *With the inclusion of a bridge at* $\overline{BF}$ *, the number of spanning trees to be calculated can now be simplified to determining the number of spanning trees in the triangle,* $\triangle DEF$ *, and in the graph formed by the vertices, A, B, C, and G.*

For a cycle graph, C_n*, the number of spanning trees is always n.*

Therefore, the number of spanning trees in $\triangle DEF$ *is 3.*

For the other subgraph, the matrix tree theorem can be applied:

The degree matrix is $\begin{bmatrix} 3 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix}$ *and the adjacency matrix is*

$\begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$*. Therefore, subtraction gives* $\begin{bmatrix} 3 & -1 & -1 & -1 \\ -1 & 2 & -1 & 0 \\ -1 & -1 & 3 & -1 \\ -1 & 0 & -1 & 2 \end{bmatrix}$.

Taking the (1,1)-minor gives $\begin{vmatrix} 2 & -1 & 0 \\ -1 & 3 & -1 \\ 0 & -1 & 2 \end{vmatrix} = \begin{vmatrix} 0 & 5 & 2 \\ -1 & 3 & -1 \\ 0 & -1 & 2 \end{vmatrix} =$

$(-1)^{2+1}(-1)\begin{vmatrix} 5 & -2 \\ -1 & 2 \end{vmatrix} = 8$. *Therefore, the number of spanning trees overall in the graph is* $3 \times 8 = 24$. □

Without noticing the bridge, determining the number of spanning trees would involve calculating the determinant of a 6×6 matrix, which involves much more work than the strategy used in Example A.3. It makes sense to make use of the new terminology and create questions that force students to understand how this terminology affects any calculations they may make.

A.10 Minimal Spanning Trees

Spanning trees are important in connecting networks (e.g. wiring in a house). However, it can be important to minimise the lengths of these networks (e.g. minimise the amount of wiring used in a house). There are two important algorithms which can be used to determine the minimal spanning tree for a given graph.

Kruskal's algorithm[86] selects edges of least weight. If, by choosing an edge, a cycle is formed, then that edge is discounted and the next edge of least weight is considered. This process continues until the moment all vertices in the graph have been selected.

Prim's algorithm[86] begins at a particular vertex and selects the edge of least weight connected to it. There are now two "active" vertices and the edge of least weight connecting either of these vertices is selected, so long as a cycle is not formed. This process continues with one additional active vertex each time until all vertices have been selected.

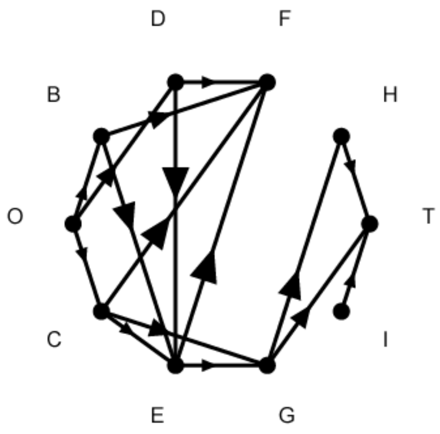
A.11 Shortest Path Algorithm

There is a popular saying that goes, "The shortest distance between any two points is a straight line.", although the source of this saying is unknown; nonetheless, if you really wish to seek a proof on this, then a good suggestion would be to visit the proof by Blochle[87]. However, this is certainly not always the case as there tend to be "obstacles" between the two points, causing the shortest possible path to be elongated by means of detours. In terms of cartography, the shortest distance between two points tends to involve travelling through multiple, different roads, all of which will likely have bends and various methods for turning from one road to another.

Being able to determine the shortest distance between two points, namely an “origin” and a “terminal”, is important and useful in many applications, especially in cartography and in business. Doing this involves looking at all possible points in between the origin and the terminal, whether they are obstacles or destinations. As such, an algorithm is needed to explore all such combinations.

One popular algorithm for solving this problem is **Dijkstra’s algorithm**[88]**,** which involves looking at the shortest possible distance to every vertex along the route to the terminal from the origin. It is a simple algorithm to learn and is known for being one of the most efficient algorithms for finding the minimum distance between a source vertex and a terminal vertex.

George wants to travel between two cities in Ghana. However, the direct route involves a rather costly toll station and so, he wants to therefore find a different route. A map he purchased provides all of the main routes between 10 cities and towns, including the departure and destination cities, as shown below, along with a distance matrix representing the distances (in km) between the communities.



	O	*B*	*C*	*D*	*E*	*F*	*G*	*H*	*I*	*T*
O	---	2	3	9	---	---	---	---	---	---
B	---	---	---	---	5	8	---	---	---	---
C	---	---	---	---	9	6	9	---	---	---
D	---	---	---	---	3	11	---	---	---	---
E	---	---	---	---	---	5	12	---	---	---
F	---	---	---	---	---	---	---	---	---	---
G	---	---	---	---	---	---	---	3	---	11
H	---	---	---	---	---	---	---	---	---	10
I	---	---	---	---	---	---	---	---	---	10
T	---	---	---	---	---	---	---	---	---	---

Avoiding the toll route, what is the minimum distance to be travelled between the two cities, O (departure city) and T (terminal city)?

Figure A.3 An example of an RNI question, asking to find the minimum distance between the departure city (labelled O) and the terminal city (labelled T).

Def. A.25 The **shortest path problem**[89] is a problem that looks to find the shortest distance between two points with various paths between them.

Example A.4 shows a method for determining the shortest distance between source and terminal vertices in a graph, G, such that all edges only move in the *forward* direction (e.g. $\overrightarrow{EG}$ may exist, but $\overrightarrow{GE}$ definitely would not exist.).

Example A.4 Determine the shortest distance and path for the shortest path problem, presented in Figure A**.3**.

Solution: *To find the minimum distance from O to T, a matrix can be used. To produce this matrix, it is important to note the following:*

1. *Determine the original distance matrix for the graph. Notice in this example that since there are no vertices travelling to I, the distance from I to T is ignored.*

$$\begin{bmatrix} & O & B & C & D & E & F & G & H & I & T \\ O & - & 2 & 3 & 9 & - & - & - & - & - & - \\ B & - & - & - & - & 5 & 8 & - & - & - & - \\ C & - & - & - & - & 9 & 6 & 9 & - & - & - \\ D & - & - & - & - & 3 & 11 & - & - & - & - \\ E & - & - & - & - & - & 5 & 12 & - & - & - \\ F & - & - & - & - & - & - & - & - & - & - \\ G & - & - & - & - & - & - & - & 3 & - & 11 \\ H & - & - & - & - & - & - & - & - & - & 10 \\ I & - & - & - & - & - & - & - & - & - & N/A \\ T & - & - & - & - & - & - & - & - & - & - \end{bmatrix}$$

2. *Starting at row O and moving to the right, determine the minimum distance* ***to*** *a particular vertex (i.e. the minimum value located in the column corresponding to a given vertex). For instance, the first minimum value would be for vertex, B, which is two.*

3. *Move to row B. For each entry in row B, add the minimum distance travelled from O to B (which is 2) to each entry. Determine the minimum distance* ***from*** *B to an adjacent vertex (in this case,* $\overrightarrow{OB}+\overrightarrow{BE}=7$*). Vertex B is now considered* ***fixed****.*

	O	B	C	D	E	F	G	H	I	T
O	–	$\boxed{2}$	3	9	–	–	–	–	–	–
B	–	–	–	–	$5+\boxed{2}$ 7	$8+\boxed{2}$ 10	–	–	–	–
C	–	–	–	–	9	6	9	–	–	–
D	–	–	–	–	3	11	–	–	–	–
E	–	–	–	–	–	5	12	–	–	–
F	–	–	–	–	–	–	–	–	–	–
G	–	–	–	–	–	–	–	3	–	11
H	–	–	–	–	–	–	–	–	–	10
I	–	–	–	–	–	–	–	–	–	N/A
T	–	–	–	–	–	–	–	–	–	–

*4. Move to the next, **non-fixed** vertex (which is the minimum value obtained in column C) and determine the minimum distance from C to an adjacent vertex (just like in step 3). In this case, we get*

	O	B	C	D	E	F	G	H	I	T
O	–	**2**	$\boxed{3}$	9	–	–	–	–	–	–
B	–	–	–	–	5+**2** 7	8+**2** 10	–	–	–	–
C	–	–	–	–	$9+\boxed{3}$ 12	$6+\boxed{3}$ 9	$9+\boxed{3}$ 12	–	–	–
D	–	–	–	–	3	11	–	–	–	–
E	–	–	–	–	–	5	12	–	–	–
F	–	–	–	–	–	–	–	–	–	–
G	–	–	–	–	–	–	–	3	–	11
H	–	–	–	–	–	–	–	–	–	10
I	–	–	–	–	–	–	–	–	–	N/A
T	–	–	–	–	–	–	–	–	–	–

5. Repeat step 4 for all vertices, moving from D to T.

Using the matrix in Figure A.3*, we obtain the following results:*

	O	B	C	D	E	F	G	H	I	T
O	–	**2**	**3**	**9**	–	–	–	–	–	–
B	–	–	–	–	5+2 **7**	8+2 10	–	–	–	–
C	–	–	–	–	9+3 12	6+3 **9**	9+3 **12**	–	–	–
D	–	–	–	–	3+9 12	11+9 20	–	–	–	–
E	–	–	–	–	–	5+7 12	12+7 19	–	–	–
F	–	–	–	–	–	–	–	–	–	–
G	–	–	–	–	–	–	–	3+12 **15**	–	11+12 **23**
H	–	–	–	–	–	–	–	–	–	10+15 25
I	–	–	–	–	–	–	–	–	–	N/A
T	–	–	–	–	–	–	–	–	–	–

In this matrix, each value in bold print represents the shortest distance ***to the vertex****, as noted in its corresponding column. Also note that since there was no path to the vertex, I, the distance from I to T was removed as part of the candidate solution.*

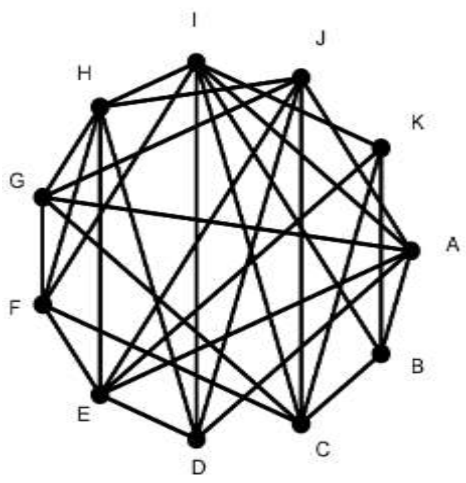
For this problem, the shortest distance is 23. However, using this, we can now also find the shortest path. The value of 23 is at the vertex pairing, (G, T)*. Therefore, the shortest path goes to G, then to T. So, we look at column, G, to find the shortest distance to it. Following this pattern through to the origin, we obtain* $T \leftarrow G \leftarrow C \leftarrow O$*, or, more simply,* $O \rightarrow C \rightarrow G \rightarrow T$*.* □

A.12 Vertex Colouring

One final topic to consider in this subject involves the colouring of vertices. The concept seems simple at first: Colour all *n* vertices with as few colours as possible. However, there is a catch: Ensure that no one colour is used on two vertices that are connected by an edge. Unfortunately, there is no known algorithm that can be used for finding an optimal colouring of a graph, but many procedures can give heuristic solutions (i.e. reasonably close solutions, although not proven to be accurate) that provide a reasonable upper bound.

An example of how vertices may be coloured using a particular, heuristic algorithm is given in Example A.5.

Example A.5 Starting at vertex, *A*, and working clockwise, determine a reasonable upper bound for the number of colours with which to colour the following graph:



Solution: *There are 11 vertices in this graph. Therefore, there will be, at most, 11 colours, say colours 1 to 11. Give each vertex an equivalent set of these 11 colours:*

$$A = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11\}$$
$$B = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11\}$$
$$\vdots$$
$$K = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11\}$$

If we start at A, then we can choose any of the 11 colours to represent it. Therefore, set the smallest element (i.e. colour 1) to represent A. But then, because the vertices adjacent to A cannot receive the same colour as A received, this causes the following changes in the colour sets:

$$A = \{1\}$$
$$B = D = E = G = I = J = \{2, 3, 4, 5, 6, 7, 8, 9, 10, 11\}$$
$$C = F = H = K = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11\}$$

*Next, we go to vertex B and repeat the process: Choose colour 2 to represent it because that is the lowest available colour for it. Now, eliminate colour 2 from all **non-fixed** vertices and we obtain*

$$
\begin{aligned}
&A=\{1\}\\
&B=\{2\}\\
&C=K=\{1,3,4,5,6,7,8,9,10,11\}\\
&D=E=G=J=\{2,3,4,5,6,7,8,9,10,11\}\\
&F=H=\{1,2,3,4,5,6,7,8,9,10,11\}\\
&I=\{3,4,5,6,7,8,9,10,11\}
\end{aligned}
$$

Following from this pattern, we obtain the following solution set:

$$A = C = H = \{1\}, B = D = F = \{2\}, E = G = I = \{3\}, J = K = \{4\}.$$

□

There are many ways with which to choose the vertices that are coloured and when during the algorithmic process. The method used in Example A.5 is one method for selecting vertices. However, another method may be to select those vertices of highest degree first. Doing this with the same graph generates the colour set,

$$I = J = \{1\}, A = C = H = \{2\}, B = E = G = \{3\}, D = F = K = \{4\}.$$

In both cases for this example, the graph could be coloured with four colours and also, the four vertex sets were virtually equivalent in size. Also, vertices, *A*, *C*, and *H*, each of which are of degree, 6, were in one set together each time. However, this could just be coincidence as there is no known proof for defining these patterns.

To understand the logic behind what is happening here, it helps to know the following definitions:

Def. A.26 The **chromatic number**[80] of a graph is the minimum number of colours needed to colour a graph so that no two vertices joined by an edge share the same colour.

Def. A.27 The **chromatic polynomial**[80] of a graph, *G*, is a polynomial which represents the colouring of *G* so that the number of ways to colour *G* with a particular number of colours can be determined.

Additionally, the following theorems help to better understand the logic behind vertex colouring:

Theorem A.3 **(Brooks' Theorem[90])** For any graph, G, with n vertices, except for complete graphs and cycle graphs where n is odd, the upper limit of the chromatic number is the maximum vertex degree present in G.

Theorem A.4 **(Four Colour Theorem[91])** Any planar graph can be coloured with, at most, four colours.

Also, note the chromatic numbers for the following graph types shown in Table A.1.

Graph Type	**Chromatic Number,** $\gamma(G)$
Cycle graph, C_n	2 (if n is even) 3 (if n is odd)
Complete graph, K_n	n
Bipartite graph, $G=(V_1 \cup V_2, E), V_1 \cap V_2 = \varnothing$	2

Table A.1 Table of graph types with their corresponding chromatic numbers.

From Example A.5 and using Theorem A.3, it could have been shown that the upper limit of the chromatic number is seven. However, do note that although the graph in Example A.5 could be coloured with fewer than five colours and that the complete graph, K_5, requires five colours, this does not make it planar. According to Kuratowski's Theorem[51], a graph, G, is not planar if and only if either K_5 or $K_{3,3}$ is homeomorphic to a subgraph of G. According to Table A.1, $\gamma(K_{3,3}) = 2$. In fact, this example does have a complete, bipartite subgraph, $K_{3,3}$, using the vertices, $V_1 = \{A, C, H\}$ and $V_2 = \{G, I, J\}$, thus making it non-planar.

Vertex colouring has valuable uses in cartography, especially in colouring neighbouring regions on maps. One good exercise (for practice) would be to find a way to colour every country in a particular continent (i.e. preferably one of Africa, Asia, or Europe) with, at most, four colours.

Appendix B T-testing Results for MC Question on Spanning Trees

The tables below detail the t-test results for the MC questions on spanning trees. The settings are coded as follows:

1. Adjacency matrices are given as options and a graph is initially given.
2. Graphs are given as options and an adjacency matrix is initially given.

A. Scenario involves business departments.
B. A directed question is given.
C. Scenario involves a link between towns.
D. Scenario involves university student services.

So, for example, if a listing says “Test 1D”, then the question was given with a graph initially, with adjacency matrices as MC options, and the indirected scenario chosen involved university student services.

	Practice-Graphs	*vs. Practice-Matrices*	*vs. Test 1A*	*vs. Test 1B*	*vs. Test 1C*	*vs. Test 1D*
Mean	8.875	9.75	2.625	2.5	1.25	1.625
Variance	66.41071	55.64286	5.125	6.857143	1.071429	7.410714
Observations	8	8	8	8	8	8
t Statistic		**-0.28103**	**2.369847**	**2.736697**	**2.764985**	**3.342271**
One-tailed critical value for t		**1.894579**	**1.894579**	**1.894579**	**1.894579**	**1.894579**
Two-tailed critical value for t		**2.364624**	**2.364624**	**2.364624**	**2.364624**	**2.364624**
			vs. Test 2A	*vs. Test 2B*	*vs. Test 2C*	*vs. Test 2D*
Mean			2.5	2.5	2.125	1.375
Variance			8.857143	12	8.125	3.410714
Observations			8	8	8	8
t Statistic			**3.093142**	**2.595351**	**2.871227**	**3.24037**
One-tailed critical value for t			**1.894579**	**1.894579**	**1.894579**	**1.894579**
Two-tailed critical value for t			**2.36462**	**2.36462**	**2.36462**	**2.36462**

Table B.1 Table of T distribution results for all style pairings with the visual practice set for MC questions on spanning trees. Results highlighted in red indicate where the null hypothesis is rejected in favour of $\boldsymbol{H_1}: \boldsymbol{\mu_1} > \boldsymbol{\mu_2}$ and the result highlighted in pink indicate a narrower rejection of the null hypothesis.

	Practice-Matrices	vs. Test 1A	vs. Test 1B	vs. Test 1C	vs. Test 1D
Mean	9.75	2.625	2.5	1.25	1.625
Variance	55.64286	5.125	6.857143	1.071429	7.410714
Observations	8	8	8	8	8
t Statistic		**2.511024**	**2.777696**	**3.156821**	**3.322584**
One-tailed critical value for t		**1.894579**	**1.894579**	**1.894579**	**1.894579**
Two-tailed critical value for t		**2.364624**	**2.364624**	**2.364624**	**2.364624**
		vs. Test 2A	vs. Test 2B	vs. Test 2C	vs. Test 2D
Mean		2.5	2.5	2.125	1.375
Variance		8.857143	12	8.125	3.410714
Observations		8	8	8	8
t Statistic		**2.7998**	**2.92731**	**3.251971**	**3.303718**
One-tailed critical value for t		**1.894579**	**1.894579**	**1.894579**	**1.894579**
Two-tailed critical value for t		**2.364624**	**2.364624**	**2.364624**	**2.364624**

Table B.2 Table of T distribution results for all style pairings with the logical / mathematical practice set for MC questions on spanning trees. Results highlighted in red indicate where the null hypothesis is rejected in favour of $\boldsymbol{H_1}: \boldsymbol{\mu_1} > \boldsymbol{\mu_2}$.

	Test 1A	vs. Test 1B	vs. Test 1C	vs. Test 1D
Mean	2.63	2.5	1.25	1.625
Variance	5.13	6.85714	1.07143	7.41071
Observations	8	8	8	8
t Statistic		**0.1286**	**1.59009**	**1.01835**
One-tailed critical value for t		**1.8946**	**1.8946**	**1.89458**
Two-tailed critical value for t		**2.3646**	**2.3646**	**2.36462**
	vs. Test 2A	vs. Test 2B	vs. Test 2C	vs. Test 2D
Mean	2.5	2.5	2.125	1.375
Variance	8.85714	12	8.125	3.41071
Observations	8	8	8	8
t Statistic	**0.12864**	**0.09706**	**0.48305**	**1.78377**
One-tailed critical value for t	**1.89458**	**1.89458**	**1.8946**	**1.8946**
Two-tailed critical value for t	**2.3646**	**2.3646**	**2.3646**	**2.3646**

Table B.3 Table of T distribution results for all style pairings with the test question set 1A for MC questions on spanning trees.

	Test 1B	vs. Test 1C	vs. Test 1D	vs. Test 2A	vs. Test 2B	vs. Test 2C	vs. Test 2D
Mean	2.5	1.25	1.625	2.5	2.5	2.125	1.375
Variance	6.85714	1.07143	7.41071	8.85714	12	8.125	3.41071
Observations	8	8	8	8	8	8	8
t Statistic		**1.41827**	**1.50716**	**0**	**0**	**0.75337**	**1.68797**
One-tailed critical value for t		**1.89458**	**1.89458**	**1.89458**	**1.89458**	**1.89458**	**1.89458**
Two-tailed critical value for t		**2.36462**	**2.36462**	**2.36462**	**2.36462**	**2.36462**	**2.36462**

Table B.4 Table of T distribution results for all style pairings with the test question set 1B for MC questions on spanning trees.

	Test 1C	vs. Test 1D	vs. Test 2A	vs. Test 2B	vs. Test 2C	vs. Test 2D
Mean	1.25	1.625	2.5	2.5	2.125	1.375
Variance	1.0714286	7.4107143	8.8571429	12	8.125	3.4107143
Observations	8	8	8	8	8	8
t Statistic		**-0.362662**	**-1.138550**	**-0.947331**	**-0.788990**	**-0.174078**
One-tailed critical value for t		**1.8945786**	**1.8945786**	**1.8945786**	**1.8945786**	**1.8945786**
Two-tailed critical value for t		**2.364624**	**2.364624**	**2.364624**	**2.364624**	**2.364624**

Table B.5 Table of T distribution results for all style pairings with the test question set 1C for MC questions on spanning trees.

	Test 1D	vs. Test 2A	vs. Test 2B	vs. Test 2C	vs. Test 2D
Mean	1.625	2.5	2.5	2.125	1.375
Variance	7.4107143	8.8571429	12	8.125	3.4107143
Observations	8	8	8	8	8
t Statistic		**-3.861741**	**-1.593970**	**-1.322876**	**0.606977**
One-tailed critical value for t		**1.8945786**	**1.8945786**	**1.8945786**	**1.8945786**
Two-tailed critical value for t		**2.3646243**	**2.3646243**	**2.3646243**	**2.3646243**

Table B.6 Table of T distribution results for all style pairings with the test question set 1D for MC questions on spanning trees. Results highlighted in red and green (one-tailed test, where $H_1 : \mu_1 < \mu_2$) indicate where the null hypothesis is rejected.

	Test 2A	vs. Test 2B	vs. Test 2C	vs. Test 2D
Mean	2.5	2.5	2.125	1.375
Variance	8.8571429	12	8.125	3.4107143
Observations	8	8	8	8
t Statistic		**0**	**0.6637465**	**2.5528888**
One-tailed critical value for t		**1.8945786**	**1.8945786**	**1.8945786**
Two-tailed critical value for t		**2.3646243**	**2.3646243**	**2.3646243**

Table B.7 Table of T distribution results for all style pairings with the test question set 2A for MC questions on spanning trees. Results highlighted in red indicate where the null hypothesis is rejected in favour of $\boldsymbol{H_1}: \boldsymbol{\mu_1} > \boldsymbol{\mu_2}$.

	Test 2B	*vs. Test 2C*	*vs. Test 2D*
Mean	2.5	2.125	1.375
Variance	12	8.125	3.4107143
Observations	8	8	8
t Statistic		**0.6637465**	**1.3502411**
One-tailed critical value for t		**1.8945786**	**1.8945786**
Two-tailed critical value for t		**2.3646243**	**2.3646243**

Table B.8 Table of T distribution results for all style pairings with the test question set 2B for MC questions on spanning trees.

	Test 2C	*vs. Test 2D*
Mean	2.125	1.375
Variance	8.125	3.4107143
Observations	8	8
t Statistic		**1.1577675**
One-tailed critical value for t		**1.8945786**
Two-tailed critical value for t		**2.3646243**

Table B.9 Table of T distribution results for all style pairings with the test question set 2C for MC questions on spanning trees.

Appendix C Statistical Analysis of Student Responses from 2008 to 2014

The following tables show the facility and discrimination statistics for all questions answered by students between 2008 and 2014.

	2008 - 2009				
Question description	**Number of Times Answered**	**Facility**	**Discrimination**	**Number of Correct Answers**	**OVERALL FACILITY**
Bipartite adjacency matrix search; MC	43	0.279	0.288	12	0.2345
Bipartite graph search; MC	38	0.211	0.492	8	0.2595
Given graph, input edges; WI+check	20	0.25	0.285	5	0.2688
Bipartite graph / adjacency matrix search; MC	36	0.278	0.385	10	0.2710
Indegree and Outdegree; RNI	28	0.25	0.471	7	0.3117
Number of vertices in a partite set of a bipartite graph; NI	35	0.257	0.538	9	0.3418
Generate the degree sequence; RWI	38	0.184	0.653	7	0.3486
Shortest distance between two towns; RNI	142	0.401	0.487	57	0.4234
Determining degree; NI	47	0.511	0.429	24	0.5063
Find the simple connected graph given the adjacency matrices; RandMC	54	0.685	0.609	37	0.5727
Find the simple connected graph given the graphs or adjacency matrices; RandMC	60	0.583	0.425	35	0.5926
Given graph, find matching adjacency matrix; MC	46	0.609	0.595	28	0.6311
Given graph, input vertices (with disconnected vertices); WI+check	25	0.52	0.476	13	0.6596
Given digraph, input edges; RWI+check	21	0.619	0.524	13	0.6667
Sum of entries (Introduction to Degree); NI	49	0.735	0.501	36	0.6667
What is wrong with the adjacency matrix; RWI+check	63	0.73	0.622	46	0.6814
Find the simple connected graph given the graphs; RandMC	42	0.643	0.218	27	0.6897
Given adjacency matrix, find matching graph; MC	48	0.708	0.512	34	0.7150
Given graph, input vertices; WI+check	19	0.632	0.372	12	0.7158
OVERALL STATISTICS	854	0.4918		420	0.5035

Table C.1 Results for 2008 – 2009 academic year, with overall facility values for 2008 – 2011; questions are ordered based on their Overall Facility values.

Facility and discrimination entries are highlighted using a variation of colours from green to yellow to red. Facility values range from 0 to 1 with 0 being in red and 1 being in green. Discrimination values range from -1 to 1, with -1 to 0 being in red, 0.5 being in green, and 1 being in yellow.

	2009 - 2010				
Question description	**Number of Times Answered**	**Facility**	**Discrimination**	**Number of Correct Answers**	**OVERALL FACILITY**
Bipartite adjacency matrix search; MC	53	0.208	0.467	11	0.2345
Bipartite graph search; MC	42	0.214	0.195	9	0.2595
Given graph, input edges; WI+check	33	0.212	0.481	7	0.2688
Bipartite graph / adjacency matrix search; MC	54	0.259	0.318	14	0.2710
Indegree and Outdegree; RNI	67	0.284	0.481	19	0.3117
Number of vertices in a partite set of a bipartite graph; NI	59	0.288	0.644	17	0.3418
Generate the degree sequence; RWI	59	0.254	0.607	15	0.3486
Shortest distance between two towns; RNI	192	0.385	0.466	74	0.4234
Determining degree; NI	53	0.472	0.532	25	0.5063
Find the simple connected graph given the adjacency matrices; RandMC	90	0.544	0.396	49	0.5727
Find the simple connected graph given the graphs or adjacency matrices; RandMC	61	0.508	0.619	31	0.5926
Given graph, find matching adjacency matrix; MC	87	0.54	0.543	47	0.6311
Given graph, input vertices (with disconnected vertices); WI+check	34	0.765	0.508	26	0.6596
Given digraph, input edges; RWI+check	27	0.778	0.557	21	0.6667
Sum of entries (Introduction to Degree); NI	57	0.649	0.487	37	0.6667
What is wrong with the adjacency matrix; RWI+check	59	0.508	0.602	30	0.6814
Find the simple connected graph given the graphs; RandMC	67	0.612	0.48	41	0.6897
Given adjacency matrix, find matching graph; MC	73	0.795	0.542	58	0.7150
Given graph, input vertices; WI+check	32	0.719	0.55	23	0.7158
OVERALL STATISTICS	1199	0.4621		554	0.5035

Table C.2 Results for 2009 – 2010 academic year, with overall facility values for 2008 – 2011; questions are ordered based on their Overall Facility values.

	2010 - 2011				
Question description	**Number of Times Answered**	**Facility**	**Discrimination**	**Number of Correct Answers**	**OVERALL FACILITY**
Bipartite adjacency matrix search; MC	49	0.224	0.586	11	0.2345
Bipartite graph search; MC	51	0.333	0.437	17	0.2595
Given graph, input edges; WI+check	40	0.325	0.464	13	0.2688
Bipartite graph / adjacency matrix search; MC	65	0.277	0.408	18	0.2710
Indegree and Outdegree; RNI	59	0.373	0.585	22	0.3117
Number of vertices in a partite set of a bipartite graph; NI	64	0.438	0.702	28	0.3418
Generate the degree sequence; RWI	78	0.5	0.542	39	0.3486
Shortest distance between two towns; RNI	214	0.472	0.473	101	0.4234
Determining degree; NI	58	0.534	0.536	31	0.5063
Find the simple connected graph given the adjacency matrices; RandMC	83	0.53	0.49	44	0.5727
Find the simple connected graph given the graphs or adjacency matrices; RandMC	68	0.676	0.468	46	0.5926
Given graph, find matching adjacency matrix; MC	92	0.728	0.415	67	0.6311
Given graph, input vertices (with disconnected vertices); WI+check	35	0.657	0.689	23	0.6596
Given digraph, input edges; RWI+check	30	0.6	0.747	18	0.6667
Sum of entries (Introduction to Degree); NI	74	0.635	0.486	47	0.6667
What is wrong with the adjacency matrix; RWI+check	82	0.768	0.639	63	0.6814
Find the simple connected graph given the graphs; RandMC	94	0.766	0.545	72	0.6897
Given adjacency matrix, find matching graph; MC	79	0.646	0.479	51	0.7150
Given graph, input vertices; WI+check	44	0.75	0.386	33	0.7158
OVERALL STATISTICS	1359	0.5475		744	0.5035

Table C.3 Results for 2010 – 2011 academic year, with overall facility values for 2008 – 2011; questions are ordered based on their Overall Facility values.

	2011 - 2012				
Question description	**Number of Times Answered**	**Facility**	**Discrimination**	**Number of Correct Answers**	**OVERALL FACILITY**
minimum spanning tree 7 vertices Prim; WI	46	0.152	0.435	7	0.1892
was AB edge added rejected unconsidered & at what step 7 vertices Prim; WI	46	0.196	0.334	9	0.1913
was AB edge added rejected unconsidered & at what step 5-6 vertices Prim; WI	47	0.234	0.393	11	0.2477
indegree of the vertex of the network matrix of a digraph; NI	14	0.214	0.468	3	0.2787
minimum spanning tree 5-6 vertices Kruskal; WI	29	0.172	0.687	5	0.2885
minimum spanning tree 5-6 vertices Prim; WI	40	0.2	0.447	8	0.2903
minimum spanning tree 7 vertices Kruskal; WI	25	0.2	0.638	5	0.2963
was AB edge added rejected unconsidered & at what step 5-6 vertices Kruskal; WI	27	0.296	0.594	8	0.3226
n'th edge minimum spanning tree 7 vertices Prim; WI	42	0.452	0.492	19	0.3248
was AB edge added rejected unconsidered & at what step 7 vertices Kruskal; WI	23	0.174	0.098	4	0.3377
outdegree of the vertex of the network matrix of a digraph; NI	11	0.273	0.527	3	0.3585
n'th edge minimum spanning tree 7 vertices Kruskal; WI	28	0.5	0.703	14	0.4286
n'th edge minimum spanning tree 5-6 vertices Kruskal; WI	21	0.524	0.265	11	0.4478
degree of the vertex of the network matrix (symmetric graph); NI	12	0.25	0.313	3	0.4630
n'th edge minimum spanning tree 5-6 vertices Prim; WI	40	0.55	0.53	22	0.5299
OVERALL STATISTICS	980	0.4735		464	0.5372

Table C.4 Some results for 2011 - 2012 academic year, with overall facility values for 2011 – 2014; questions are ordered based on their Overall Facility values.

	2011 - 2012				
Question description	**Number of Times Answered**	**Facility**	**Discrimination**	**Number of Correct Answers**	**OVERALL FACILITY**
indegree of the vertex of the adjacency matrix; NI	12	0.25	0.569	3	0.5439
degree sequence of the adjacency matrix (simple, disconnected graph); WI	16	0.688	0.62	11	0.5741
degree sequence of the graph (with multi edges and loops); WI	13	0.538	0.366	7	0.5778
Given disconnected graph_input vertex set; WI+check	44	0.591	0.757	26	0.6158
Given simple, connected graph_input edge set; WI+check	50	0.62	0.657	31	0.6205
Given graph, find matching adjacency matrix; MC	88	0.614	0.639	54	0.6400
Given graph with loops_input edge set; WI+check	53	0.604	0.778	32	0.6548
degree sequence of the adjacency matrix (with multi edges); WI	17	0.471	0.575	8	0.6667
outdegree of the vertex of the adjacency matrix; NI	16	0.438	0.626	7	0.6939
degree sequence of the adjacency matrix (with multi edges and loops); WI	13	0.692	0.499	9	0.7073
degree sequence of the graph (simple, disconnected graph); WI	15	0.533	0.514	8	0.7111
degree sequence of the adjacency matrix (simple, connected graph); WI	10	0.6	0.824	6	0.7119
degree sequence of the graph (with multi edges); WI	14	0.5	0.648	7	0.7193
degree sequence of the graph (simple, connected graph); WI	9	0.778	0.689	7	0.7193
What is wrong with the adjacency matrix; RWI+check	95	0.684	0.653	65	0.7346
Given connected graph_input vertex set; WI+check	64	0.797	0.518	51	0.8387
OVERALL STATISTICS	980	0.4735		464	0.5372

Table C.5 Some results for 2011 – 2012 academic year, with overall facility values for 2011 – 2014; questions are ordered based on their Overall Facility values.

	2012 - 2013				
Question description	**Number of Times Answered**	**Facility**	**Discrimination**	**Number of Correct Answers**	**OVERALL FACILITY**
minimum spanning tree 7 vertices Prim; WI	32	0.281	0.671	9	0.1892
was AB edge added rejected unconsidered & at what step 7 vertices Prim; WI	33	0.242	0.647	8	0.1913
was AB edge added rejected unconsidered & at what step 5-6 vertices Prim; WI	32	0.344	0.428	11	0.2477
indegree of the vertex of the network matrix of a digraph; NI	18	0.222	0.715	4	0.2787
minimum spanning tree 5-6 vertices Kruskal; WI	42	0.333	0.781	14	0.2885
minimum spanning tree 5-6 vertices Prim; WI	55	0.345	0.639	19	0.2903
minimum spanning tree 7 vertices Kruskal; WI	24	0.25	0.625	6	0.2963
was AB edge added rejected unconsidered & at what step 5-6 vertices Kruskal; WI	26	0.423	0.658	11	0.3226
n'th edge minimum spanning tree 7 vertices Prim; WI	36	0.25	0.554	9	0.3248
was AB edge added rejected unconsidered & at what step 7 vertices Kruskal; WI	25	0.32	0.651	8	0.3377
outdegree of the vertex of the network matrix of a digraph; NI	14	0.429	0.402	6	0.3585
n'th edge minimum spanning tree 7 vertices Kruskal; WI	29	0.448	0.528	13	0.4286
n'th edge minimum spanning tree 5-6 vertices Kruskal; WI	26	0.5	0.671	13	0.4478
degree of the vertex of the network matrix (symmetric graph); NI	30	0.5	0.675	15	0.4630
n'th edge minimum spanning tree 5-6 vertices Prim; WI	31	0.516	0.553	16	0.5299
OVERALL STATISTICS	1235	0.5733		708	0.5372

Table C.6 Some results for 2012 – 2013 academic year, with overall facility values for 2011 – 2014; questions are ordered based on their Overall Facility values.

	2012 - 2013				OVERALL FACILITY
Question description	**Number of Times Answered**	**Facility**	**Discrimination**	**Number of Correct Answers**	
indegree of the vertex of the adjacency matrix; NI	22	0.591	0.75	13	0.5439
degree sequence of the adjacency matrix (simple, disconnected graph); WI	21	0.619	0.554	13	0.5741
degree sequence of the graph (with multi edges and loops); WI	16	0.688	0.315	11	0.5778
Given disconnected graph_input vertex set; WI+check	77	0.675	0.702	52	0.6158
Given simple, connected graph_input edge set; WI+check	68	0.618	0.67	42	0.6205
Given graph, find matching adjacency matrix; MC	137	0.657	0.459	90	0.6400
Given graph with loops_input edge set; WI+check	76	0.658	0.632	50	0.6548
degree sequence of the adjacency matrix (with multi edges); WI	17	0.647	0.455	11	0.6667
outdegree of the vertex of the adjacency matrix; NI	19	0.842	0.029	16	0.6939
degree sequence of the adjacency matrix (with multi edges and loops); WI	16	0.75	0.063	12	0.7073
degree sequence of the graph (simple, disconnected graph); WI	21	0.81	0.607	17	0.7111
degree sequence of the adjacency matrix (simple, connected graph); WI	20	0.7	0.812	14	0.7119
degree sequence of the graph (with multi edges); WI	20	0.8	0.526	16	0.7193
degree sequence of the graph (simple, connected graph); WI	23	0.696	0.734	16	0.7193
What is wrong with the adjacency matrix; RWI+check	137	0.686	0.654	94	0.7346
Given connected graph_input vertex set; WI+check	92	0.859	0.476	79	0.8387
OVERALL STATISTICS	1235	0.5733		708	0.5372

Table C.7 Some results for 2012 – 2013 academic year, with overall facility values for 2011 – 2014; questions are ordered based on their Overall Facility values.

	2013 - 2014				
Question description	**Number of Times Answered**	**Facility**	**Discrimination**	**Number of Correct Answers**	**OVERALL FACILITY**
minimum spanning tree 7 vertices Prim; WI	33	0.152	0.651	5	0.1892
was AB edge added rejected unconsidered & at what step 7 vertices Prim; WI	36	0.139	0.09	5	0.1913
was AB edge added rejected unconsidered & at what step 5-6 vertices Prim; WI	30	0.167	0.451	5	0.2477
indegree of the vertex of the network matrix of a digraph; NI	29	0.345	0.255	10	0.2787
minimum spanning tree 5-6 vertices Kruskal; WI	33	0.333	0.408	11	0.2885
minimum spanning tree 5-6 vertices Prim; WI	29	0.31	0.732	9	0.2903
minimum spanning tree 7 vertices Kruskal; WI	32	0.406	0.576	13	0.2963
was AB edge added rejected unconsidered & at what step 5-6 vertices Kruskal; WI	40	0.275	0.392	11	0.3226
n'th edge minimum spanning tree 7 vertices Prim; WI	39	0.256	0.612	10	0.3248
was AB edge added rejected unconsidered & at what step 7 vertices Kruskal; WI	29	0.483	0.699	14	0.3377
outdegree of the vertex of the network matrix of a digraph; NI	28	0.357	0.407	10	0.3585
n'th edge minimum spanning tree 7 vertices Kruskal; WI	20	0.3	0.692	6	0.4286
n'th edge minimum spanning tree 5-6 vertices Kruskal; WI	20	0.3	0.738	6	0.4478
degree of the vertex of the network matrix (symmetric graph); NI	12	0.583	0.557	7	0.4630
n'th edge minimum spanning tree 5-6 vertices Prim; WI	46	0.522	0.576	24	0.5299
OVERALL STATISTICS	1143	0.5529		632	0.5372

Table C.8 Some results for 2013 – 2014 academic year, with overall facility values for 2011 – 2014; questions are ordered based on their Overall Facility values.

	2013 - 2014				
Question description	**Number of Times Answered**	**Facility**	**Discrimination**	**Number of Correct Answers**	**OVERALL FACILITY**
indegree of the vertex of the adjacency matrix; NI	23	0.652	0.571	15	0.5439
degree sequence of the adjacency matrix (simple, disconnected graph); WI	17	0.412	0.787	7	0.5741
degree sequence of the graph (with multi edges and loops); WI	16	0.5	0.378	8	0.5778
Given disconnected graph_input vertex set; WI+check	56	0.554	0.627	31	0.6158
Given simple, connected graph_input edge set; WI+check	48	0.625	0.611	30	0.6205
Given graph, find matching adjacency matrix; MC	125	0.64	0.556	80	0.6400
Given graph with loops_input edge set; WI+check	68	0.691	0.498	47	0.6548
degree sequence of the adjacency matrix (with multi edges); WI	20	0.85	0.608	17	0.6667
outdegree of the vertex of the adjacency matrix; NI	14	0.786	0.053	11	0.6939
degree sequence of the adjacency matrix (with multi edges and loops); WI	12	0.667	0.665	8	0.7073
degree sequence of the graph (simple, disconnected graph); WI	9	0.778	0.114	7	0.7111
degree sequence of the adjacency matrix (simple, connected graph); WI	29	0.759	0.59	22	0.7119
degree sequence of the graph (with multi edges); WI	23	0.783	0.727	18	0.7193
degree sequence of the graph (simple, connected graph); WI	25	0.72	0.224	18	0.7193
What is wrong with the adjacency matrix; RWI+check	141	0.816	0.629	115	0.7346
Given connected graph_input vertex set; WI+check	61	0.852	0.417	52	0.8387
OVERALL STATISTICS	1143	0.5529		632	0.5372

Table C.9 Some results for 2013 – 2014 academic year, with overall facility values for 2011 – 2014; questions are ordered based on their Overall Facility values.

Appendix D Analysis of Final Examinations for MA2920

Tables in this section summarise errors made by 265 students when attempting graph theory questions in MA2920: Algebra and Discrete Mathematics from 2005 to 2008. There were four questions on each examination. For each examination, questions three and four represent questions on graph theory; questions one and two represent questions from another subject within the module and were not reviewed. Numbers of attempts made on all questions during MA2920 examinations are included. Correct answers are also included, along with errors made by the assessor(s) when marking examination scripts.

	2004-2005	2005-2006	2006-2007	2007-2008	TOTAL
Question One	8	1	1	0	10
Question Two	2	24	23	13	62
Question Three	11	1	0	14	26
Question Four	21	26	18	44	109
None	27	13	26	13	79
TOTAL	69	65	68	84	**286**

Table D.1 Table listing the numbers of students who did not perform which questions, along with the numbers of students who performed all questions during the MA2920 examinations from 2005 to 2008.

SPANNING TREES QUESTION		
Types of Errors Made	**Count**	**Percentage**
No Attempt	26	9.8%
Reading Question	1	0.4%
Methodology	51	19.2%
Accidental	3	1.1%
Guesswork	19	7.2%
Calculation	93	35.1%
Lack of work shown	22	8.3%
None	60	22.6%
Assessor	21	7.9%
Knowledge	1	0.4%
Strategy	13	4.9%
Matrix	1	0.4%
Process	7	2.6%
Setup	5	1.9%

Table D.2 Categorisation of errors made in MA2920 examinations from 2005 to 2008 for questions investigating spanning trees. Significant results are highlighted in various colours.

VERTEX COLOURING QUESTION		
Errors Made – Vertex Colouring Question	**Count**	**Percentage**
Reading question	1	0.4%
Guesswork	10	3.8%
Accidental	8	3.0%
Calculation	11	4.2%
Lack of work shown	20	7.5%
Methodology	56	21.1%
None	43	16.2%
Procedural / Strategy	29	10.9%
Not attempted	109	41.1%
Unknown	1	0.4%
ASSESSOR	3	1.1%

Table D.3 Categorisation of errors in MA2920 examinations from 2005 to 2008 for questions investigating vertex colouring. Significant results are highlighted.

Errors Made- Spanning Trees Question	Count	Percentage
Added (C_4) one too many times	1	0.3774%
Created loop around a vertex; made solving difficult	1	0.3774%
Matrix for calculations incorrect	7	2.6415%
Broke subgraph into 2 independent sets	1	0.3774%
Squared subgraphic portion in calculation	2	0.7547%
Miscalculated trees of subgraph	6	2.2642%
Did not answer question asked; wrong graph drawn	2	0.7547%
Assumed wrong quantities of shapes	2	0.7547%
Assumed odd vertices at tree ends	1	0.3774%
Did not "pinch" properly	46	17.3585%
Work shown not explicit enough	1	0.3774%
Bad use / calculation of matrices	12	4.5283%
Counted tree of n vertices = n (or other) subtrees	3	1.1321%
Accidental change of value in matrix	1	0.3774%
Did not calculate subtrees properly	12	4.5283%
Willing to add instead of multiply	43	16.2264%
Random guess	17	6.4151%
Did not square 11	7	2.6415%
Did not show all work	8	3.0189%
Tried to find subtrees manually	5	1.8868%
Left out portion of calculation	5	1.8868%
Not completed	13	4.9057%
Correct	60	22.6415%
Assessor Error	**7**	2.6415%
Tried to use vertex degrees to solve	1	0.3774%
Used pinching method for chromatics	2	0.7547%
Assumed $T(C^4) = T(K^4)$	1	0.3774%
Assumed subgraph was "near complete"	1	0.3774%
Only viewed cycles of subgraphs	1	0.3774%
Used wrong subgraph	1	0.3774%
Too many cofactors when solving $DET(B_1)$	1	0.3774%
Skipped question	30	11.3208%
Assumed $T(Tree_n)$ to be a different value	2	0.7547%
Assumed $T(K^4)$ to equal some other number	3	1.1321%
Counted too many triangles	1	0.3774%
Squared all values when multiplying	1	0.3774%

Table D.4 Summarisation of errors made by students in MA2920 examinations from 2005 to 2008 while attempting to answer questions on spanning trees. Significant results are highlighted in various colours.

Errors Made- Vertex Colouring Question	Count	Percentage
Assumed complete graph of (n+2) vertices	1	0.3774%
Did not answer 2nd part of question	1	0.3774%
Added unnecessary vertex when pinching	1	0.3774%
Random guess	11	4.1509%
Did not draw correct graph to solve	3	1.1321%
Changed graph during procedure	2	0.7547%
Set to be largest x s.t. _(G) = 0 (or otherwise)	3	1.1321%
Assumed wrong graph types when branches included	1	0.3774%
Pinching method incorrect	42	15.8491%
Not: 2 loops around 2 vertices = 1 branch	1	0.3774%
Deleted branches instead of key edges	1	0.3774%
Did not use proper procedure to solve	5	1.8868%
Did not show all work	5	1.8868%
One subgraph had wrong polynomial	1	0.3774%
Did not complete	14	5.2830%
Correct	43	16.2264%
Removed part of graph when "pinching"	7	2.6415%
Skipped question	109	41.1321%
Added K_3 one too many times	1	0.3774%
Calculations within pinching method incorrect	4	1.5094%
Willing to add instead of subtract or multiply	2	0.7547%
Assessor Error	2	0.7547%
Work shown not explicit enough	15	5.6604%
Assumed another strategy	4	1.5094%
Used wrong graph	3	1.1321%
Left out part of answer, which (s)he had found	1	0.3774%
Work appears correct, but does not match	1	0.3774%
Assumed K_4 from a subgraph of 5 vertices	2	0.7547%
Assumed graph was almost K_5	2	0.7547%
Thought $C_3 = K_2$	1	0.3774%
Thought $T_2 = K_2$	1	0.3774%

Table D.5 Summarisation of errors made by students in MA2920 examinations from 2005 to 2008 while attempting to answer questions on vertex colouring. Significant results are highlighted in various colours.

References

[1] Pearson Education Limited (2013). *Specification: GCE Mathematics*. Pp 97 – 106. http://qualifications.pearson.com/content/dam/pdf/A%20Level/Mathematics/2013/Specification%20and%20sample%20assessments/UA035243_GCE_Lin_Maths_Issue_3.pdf. Referenced on 3 August 2015.

[2] The Assessment and Qualifications Alliance (AQA; 2013). *General Certificate of Education: Mathematics 6360 (2014)*. pp. 84 – 86. http://filestore.aqa.org.uk/subjects/specifications/alevel/AQA-6360-W-SP-14.PDF. Referenced on 3 August 2015.

[3] Oxford, Cambridge and RSA Examinations (OCR; 2010). *OCR Advanced Subsidiary GCE in Mathematics (3890)*. pp. 82 – 85. http://www.ocr.org.uk/Images/67746-specification.pdf. Referenced on 3 August 2015.

[4] AQA (2013). *GCE Mathematics – MD01 Decision 1: Report on the Examination*. http://filestore.aqa.org.uk/subjects/AQA-MD01-WRE-JUN13.PDF. Referenced on 3 August 2015.

[5] Pearson Education Limited (2015). *Results and certification: Grade statistics*. http://qualifications.pearson.com/en/support/support-topics/results-certification/grade-statistics.html?Qualification-Family=A-Level. Referenced on 5 August 2015.

[6] Department of Mathematical Sciences at Brunel University. *MA0422: Discrete and Decision Mathematics.* London, U.K. http://readinglists.brunel.ac.uk/lists/C52EC70A-0F00-318F-02D8-A1E719CE93ED.html. Referenced on 3 August 2015.

[7] Department of Mathematical Sciences at Brunel University. *MA2726: Elements of Combinatorics*. London, U.K. http://readinglists.brunel.ac.uk/lists/E0E8CF03-3C52-96ED-C7B1-61CAE60EAD28.html. Referenced on 3 August 2015.

[8] Higher Education Statistics Agency (not dated). *Free Online Statistics – Students & qualifiers*. https://www.hesa.ac.uk/stats. Referenced on 26 August 2015.

[9] World Wide Web Foundation (2015). *History of the Web*. http://webfoundation.org/about/vision/history-of-the-web/. Referenced on 4 August 2015.

[10] M. Llamas-Nistal, M.. J. Fernández-Iglesias, J. González-Tato, F. A. Mikic-Fonte (2013). *Blended e-assessment: Migrating classic exams to the digital world*. Computers and Education, Volume 62, pp. 72 – 87.

[11] G. Lambert (2004). *What is Computer Aided Assessment and How Can I Use It in My Teaching?* (Briefing Paper). Canterbury Christ Church College, U.K. Available via http://www.canterbury.ac.uk/support/learning-teaching-enhancement-unit/publications/FLT-briefing-notes/Briefing-Note-CAA.pdf. Referenced on 16 July 2009.

[12] Learning and Development Centre at Warwick University (2012). *Introduction to Objective Testing and CAA at Warwick*. Warwick University, Coventry, U.K. http://www2.warwick.ac.uk/services/ldc/resource/eguides/caaintro/#what. Referenced on 4 August 2015.

[13] Teaching for Success National Faculty Success Center (2014). *How to Evaluate & Test Learning*. Pentronics Publishing, Rio Rancho, New Mexico, U.S.A. http://teachingforsuccess.com/QC4Mrk14/TFS_TestingEval_QC-Mrkt.pdf. Referenced on 23 July 2015.

[14] C. J. Sangwin (2013). *Computer Aided Assessment of Mathematics*. Oxford University Press, Oxford, U.K.

[15] Scottish Qualifications Authority (SQA; 2005). *SQA Guidelines on e-assessment for Schools*. Glasgow, Scotland, U.K.

[16] D.R. Sadler (1989). *Formative assessment and the design of instructional systems*. Instructional Science 18, pp. 119 – 144. Kluwer Academic Publishers, Dordrecht, The Netherlands. Available via http://link.springer.com/article/10.1007/BF00117714#page-1.

[17] A.A. Rupp, J. Templin, R. A. Henson (2010). *Diagnostic Measurement: Theory, Methods, and Applications*. The Guilford Press, New York City, U.S.A.

[18] Newcastle University. *Numbas*. http://www.numbas.org.uk. Referenced on 23 July 2015.

[19] University of the West of England. *DEWIS e-Assessment System*. http://www.cems.uwe.ac.uk/dewis/welcome/index.html. Referenced on 23 July 2015.

[20] STACK (System for Teaching and Assessment using a Computer algebra Kernel). http://stack.bham.ac.uk/. Referenced 4 July 2015.

[21] Maplesoft™, a subsidiary of Cybernet Systems Co. Ltd., Japan (based in Waterloo, Ontario, Canada). *Maple T.A.* http://www.maplesoft.com/products/mapleta/. Referenced 1 July 2015.

[22] MyMathLab®. Pearson Education, Inc. http://www.pearsonmylabandmastering.com/global/mymathlab-global/. Referenced on 23 July 2015.

[23] John Wiley & Sons, Inc. *WileyPLUS*. https://www.wileyplus.com/WileyCDA/. Referenced on 23 July 2015.

[24] R. Gwynllyw & K. Henderson (2008). *DEWIS – a computer aided assessment system for mathematics and statistics*. CETL-MSOR Conference 2008, Lancaster University. Available via https://www.heacademy.ac.uk/sites/default/files/CETLMSOR2008_Proceedings.pdf.

[25] J. Ruokokoski (2009). *Automatic Assessment in University-level Mathematics.* Helsinki University of Technology, Helsinki, Finland. Available via https://aaltodoc.aalto.fi/bitstream/handle/123456789/11451/master_ruokokoski_jarno_2009.pdf?sequence=1.

[26] A. Kamavi, M. Greenhow, R. Gwynllyw, et al. *Maths e.g.* (Maths Examples Galore). http://www.mathcentre.ac.uk:8081/mathseg/. Referenced on 4 July 2015.

[27] M. Greenhow (2008). *Mathletics – a suite of computer-assisted assessments.* MSOR Connections, Vol. 8, No. 3. Available via http://journals.heacademy.ac.uk/doi/pdf/10.11120/msor.2008.08030007.

[28] Questionmark Computing Limited (2015). *Questionmark Perception*. https://www.questionmark.co.uk/content/questionmark-perception. Referenced on 6 August 2015.

[29] E. Ellis, M. Greenhow, J. Hatt (2006). *Exportable Technologies: MathML and SVG objects for CAA and web content*. In M. Danson (ed.) 10th CAA International Computer Assisted Assessment Conference: Proceedings of the Conference on 4th and 5th July 2006 at Loughborough University, Loughborough, Loughborough University, Loughborough, U.K., pp. 181 – 194.

[30] J. Hatt, M. Greenhow (2008). *Implementing graph theory into Mathletics*. CETL-MSOR Conference 2007 Proceedings, Birmingham, pp. 56 – 59. Available via http://www-new1.heacademy.ac.uk/assets/documents/rc_import/CETLMSOR2007_Proceedings.pdf#page=56.

[31] J. Hatt (2007). *Computer-Aided Assessment and Learning in Decision-Based Mathematics*. Proceedings of the IADIS International Conference on e-Learning, Lisbon, Portugal, pp. 382 – 385.

[32] J. Bull & C. McKenna (2004). *Blueprint for computer-assisted assessment*. RoutledgeFalmer, London, U.K.

[33] A. Quint (2003). *Scalable vector graphics*. Multimedia, IEEE, Volume 10, Issue 3, pp. 99 – 102.

[34] M. Greenhow, K. Zaczek, A. Kamavi (2011). *Development and integration of computer-aided assessment of discrete mathematics*. MSOR Connections, Volume 11, Number 3, pp. 31 – 34.

[35] J. Hatt (2008). *Alternative Methods to Summative, Online Assessments in Mathematics and Other Applications*. Proceedings of IADIS International Conference e-Learning 2008, Amsterdam, The Netherlands, pp. 419 – 426.

[36] M. Gill, M. Greenhow (2007). *Computer-aided assessment in mechanics: question design and test evaluation*. Teaching Mathematics and Its Applications, Volume 26, Number 3, pp. 124 – 133.

[37] The British Dyslexia Association (not dated). *Eyes and Dyslexia*. http://www.bdadyslexia.org.uk/dyslexic/eyes-and-dyslexia. Referenced on 8 August 2015.

[38] S. A. Trewin (2007). *Robert Yerkes' Multiple-Choice Apparatus 1913 – 1939*. The American Journal of Psychology, Volume 120, Number 4, pp. 645 – 660.

[39] M. R. Harrower-Erickson (1943). *A Multiple-choice Test for Screening Purposes*. Psychosomatic Medicine, Vol. 5, Number 4, pp. 331 – 341.

[40] Lt. Comdr. C. L. Wittson, Lt. Comdr. W. A. Hunt, Lt. H. J. Older (1944). *The Use of The Multiple-choice Group Rorschach Test in Military Screening*. The Journal of Psychology, Vol. 17, Number 1, pp. 91 – 94.

[41] C. Torres, A. Lopes, L. Babo, J. Azevedo (2009). *Developing Multiple-Choice Questions in Mathematics*. Proceedings of 2nd International Conference of Education, Research, and Innovation, pp. 6218 – 6229.

[42] N. Baruah, M. Gill, M. Greenhow (2006). *Issues with setting online objective questions and testing their efficacy*. In M. Danson (ed.) 10th CAA International Computer Assisted Assessment Conference: Proceedings of the Conference on 4th and 5th July 2006 at Loughborough University. Loughborough University, Loughborough, U.K., pp. 55 – 70.

[43] Department of Mathematics: MA2920- Algebra and Discrete Mathematics. Brunel University (2008).

[44] K. Zaczek (2015). *Development and evaluation of computer-aided assessment in discrete and decision mathematics.* Ph.D. Thesis, Department of Mathematical Sciences, Brunel University, Uxbridge, U.K.

[45] Question Mark Computing Ltd. (2000). *Question Mark Perception Authoring Manual*. London, U.K.

[46] E. W. Weisstein. Degree Sequence. From *MathWorld*--A Wolfram Web Resource. http://mathworld.wolfram.com/DegreeSequence.html (Referenced 19 July 2015.).

[47] S. Epp (2004). Discrete Mathematics with Applications: Third Edition (International Student Edition). Brooks/Cole, Belmont, U.S.A.

[48] Grimaldi, Ralph P. (1999) *Discrete and Combinatorial Mathematics- An Applied Introduction: Fourth Edition*. Addison-Wesley Longman, Inc., Reading, Massachusetts, U.S.A.

[49] D. B. West (2001). *Introduction to Graph Theory: Second Edition*. Prentice Hall, Inc., Upper Saddle River, New Jersey, U.S.A.

[50] J. Gross and J. Yellen (1999). *Graph Theory and Its Applications*. CRC Press, Boca Raton, United States of America.

[51] T. Nishizeki, N. Chiba (1988). *Planar Graphs: Theory and Algorithms*. Dover Publications, Inc., Mineola, New York, USA.

[52] Hebborn, John (2001). Heinemann Modular Mathematics for Edexcel AS and A-Level: Decision Mathematics 2. Heinemann Educational Publishers, Oxford, U.K.

[53] Weisstein, Eric W. (1999). *Eulerian Circuit*. Mathworld- A Wolfram Web Resource. http://mathworld.wolfram.com/EulerianCircuit.html/ (Accessed on February 14, 2007.).

[54] J. Martin (2008). *Topics in Graph Theory- Lecture Notes I*. University of Kansas, Lawrence, U.S.A. http://www.math.ku.edu/~jmartin/mc2004/graph1.pdf (Accessed on 8 June 2008.).

[55] Troin, P. (2007) *Analysis of Variance*. http://www.fifi.org/doc/gnumeric-doc/html/C/anova.html. Referenced on 12 July 2008.

[56] Masse, D. (2006). *T-tests*. Wake Forest University, Winston-Salem, U.S.A. http://www.wfu.edu/~massd2/T_test.htm. Referenced on 12 July 2008.

[57] Montgomery, D.C. (2005). *Design and Analysis of Experiments- 6th Edition*. John Wiley & Sons, Inc., Danvers and Hoboken, U.S.A.

[58] Lane, D. et al. (2008). *HyperStat Online: Introduction to ANOVA*. Part of the *Rice Virtual Lab in Statistics*, Rice University, Houston, U.S.A. http://davidmlane.com/hyperstat/intro_ANOVA.html. Referenced on 11 July 2008.

[59] Sullivan, J. (2007). *Psychology 290: Research Methods and Statistics 2007-2008*. St. Francis Xavier University, Antigonish, Canada. http://people.stfx.ca/jfsulliv/Psych290/Psychology290Syllabus.htm. Referenced on 30 July 2008.

[60] G.A. Dirac (1952). *Some theorems on abstract graphs.* Proceedings of the London Mathematical Society, Vol. S3-2, Issue 1, pp. 69 – 81.

[61] L. J. Grady, J. R. Polimeni (2010). *Discrete Calculus: Applied Analysis on Graphs for Computational Science*. Springer-Verlag, London.

[62] G. Farrell & Y. Leung (2008). *Convergence of Validity for The Results of a Summative Assessment with Confidence Measurement and Traditional Assessment*. CAA Conference 2008. Proceedings available via http://www.caaconference.com.

[63] F. Wilcoxon (1945). *Individual Comparisons by Ranking Methods.* Biometrics, Vol. 1, pp. 80-83.

[64] H. MacGillivray (2007). *Roles of Assessment in Learning in Statistics and Mathematics*. CETL-MSOR Conference 2007. Proceedings available via http://www.ltsn.gla.ac.uk/index.php?pid=61.

[65] B. Means, Y. Toyama, R. Murphy, M. Bakia, K. Jones (2009). *Evaluation of Evidence-Based Practices in Online Learning: A Meta-Analysis and Review of Online Learning Studies*. U.S. Department of Education, Washington, D.C. Available on the Department's website at http://www.ed.gov/about/offices/list/opepd/ppss/reports.html.

[66] T. B. Huedo-Medina, J. Sánchez-Meca, F. Marín-Martínez, J. Botella (2006). *Assessing Heterogeneity in Meta-Analysis: Q Statistic or I^2 Index?* Psychological Methods, Vol. 11, No. 2, pp. 193-206.

[67] J. Grierson (2015). *Student protest against 'unfair' GCSE maths question goes viral.* Guardian News and Media, Ltd. https://www.theguardian.com/education/2015/jun/05/students-protest-gcse-math-question-viral-trending-twitter.

[68] H. Horton (2016). *'They must have given us the wrong paper' – Students fume after being set 'impossible' maths exam which was 'nothing like' past papers.* The Telegraph: Education. http://www.telegraph.co.uk/education/2016/05/18/they-must-have-given-us-the-wrong-paper---students-fume-after-be/.

[69] Schmeck, Ronald R., ed. (2013). *Learning strategies and learning styles*. Springer Science & Business Media.

[70] Oxford University Press (2015). *MyMaths*. https://www.mymaths.co.uk/index.html.

[71] L. W. Anderson & D. R. Krathwohl (Eds., 2001). *A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives*. Longman, New York City, U.S.A.

[72] B.S. Bloom (1956). *Taxonomy of Educational Objectives, Handbook I: The Cognitive Domain.* David McKay Co. Inc., New York.

[73] D. Krathwohl (2002). *A Revision of Bloom's Taxonomy: An Overview*. Theory Into Practice, Volume 41, Number 4, pp. 212 – 218.

[74] G. Smith, L. Wood, M. Coupland, B. Stephenson (1996). *Constructing mathematical examinations to assess a range of knowledge and skills*. International Journal of Mathematical Education in Science and Technology, Volume 27, Number 1, pp. 65 – 77.

[75] N. Baruah & J. Hatt (2007). *The Reconceptualisation of The Revised Bloom's Taxonomy for Use in Mathematics and Its Implementation into QM Perception and Mathletics*. Pedagogical Research in Maximising Education (PRIME) Journal, Vol. 2 (2). Liverpool Hope University, Liverpool, U.K.

[76] D. Nelson (Ed.). *Dictionary of Mathematics: Second Edition.* Penguin Books, London, U.K. 1998.

[77] M. van Steen. *Graph Theory and Complex Networks: An Introduction*. VU University, Amsterdam, The Netherlands.

[78] B. Hopkins (Ed.) (2009). *Resources for Teaching Discrete Mathematics: Classroom Projects, History Modules, and Articles*. Mathematical Association of America, Washington, D.C., United States of America.

[79] Kaliningrad City Hall (2006). *Official site of Kaliningrad City Hall: History of Kaliningrad*. Kaliningrad City Hall. http://www.klgd.ru/en/city/history/ (Accessed on February 6, 2007.).

[80] J. L. Gross, J. Yellen, P. Zhang (2013). *Handbook of Graph Theory, Second Edition*. CRC Press, Boca Raton, United States of America.

[81] J.E. Graver & M. E. Watkins (1977). *Combinatorics with emphasis on the theory of graphs*. Springer-Verlag, New York City, United States of America.

[82] J. Köbler, U. Schöning, J. Torán (1993). *The Graph Isomorphism Problem: Its Structural Complexity*. Springer Science+Business Media, New York City, United States of America.

[83] R. Bapat, J. Grossman, D. Kulkarni (2001). *Matrix tree theorem*. Springer Online Reference Works. http://eom.springer.de/m/m130130.htm#m130130_00ca9 (Referenced on 20 May 2008.).

[84] JOC/EFR (2002). *Kirchhoff biography*. School of Mathematics and Statistics, University of St. Andrews, Scotland, U.K. http://www-groups.dcs.st-and.ac.uk/~history/Biographies/Kirchhoff.html (Accessed on 20 May 2008.).

[85] Weisstein, Eric W. "Graph Bridge." From *MathWorld*--A Wolfram Web Resource. http://mathworld.wolfram.com/GraphBridge.html (Accessed on 1 June 2008.).

[86] S. G. Jameson (2010). *Decision Mathematics 1: EdExcel AS and A-Level Modular Mathematics D1*. Pearson Education Limited, Harlow, United Kingdom.

[87] P. Blochle (Date unknown.). *Proof shortest distance between two point is a straight line*. Canisius College, Buffalo, U.S.A. http://www.instant-analysis.com/Principles/straightline.htm (Accessed on 9 June 2008.).

[88] E. W. Dijkstra (1959). *A note on two problems in connexion with graphs*. Numerische Mathematik, Vol. 1, Issue 1, pp. 269 – 271.

[89] L. Liu (2003). Lecture notes for Math 3333- Operations Research – Winter 2003. Lakehead University, Thunder Bay, Canada.

[90] W. D. Wallis (2007). *A Beginner's Guide to Graph Theory: Second Edition*. Birkhäuser, Boston, United States of America.

[91] K. Appel, W. Haken (1989). *Every Planar Map is Four Colorable*. Contemporary Mathematics, v. 98. American Mathematical Society, Providence, Rhode Island, USA.