
BOOK REVIEW- “Discrete Mathematics and Its Applications”

Author: Kenneth Rosen Edition: Seventh

Dr. Madhukar Mhaluba Palve

Department of Mathematics,
Prof. Ramkrishna More ACS College, Akurdi, Pune, India- 44.

Discrete Mathematics and its Applications is a focused introduction to the primary themes in a discrete mathematics course, as introduced through extensive applications, expansive discussion, and detailed exercise sets. These themes include mathematical reasoning, combinatorial analysis, discrete structures, algorithmic thinking, and enhanced problem-solving skills through modeling. Its intent is to demonstrate the relevance and practicality of discrete mathematics to all students. This enhanced coverage will provide students with a solid understanding of the material as it relates to their immediate field of study and other relevant subjects. The inclusion of applications and examples to key topics has been significantly addressed to add clarity to every subject.

The book is appropriate for a one- or two-term introductory discrete mathematics course to be taken by students in a wide variety of majors, including computer science, mathematics, and engineering. College Algebra is the only explicit prerequisite.

Additionally, the author Kenneth Rosen provides a lot of examples for each of its theorems and topics. The book also fleshes out the key subjects (counting, proofs, graphs, etc.) while also providing a high level overview of their applications. These combine to make it an excellent first textbook for learning discrete mathematics.

I still highly recommend it for those not familiar with the topics covered in the book. I have summarized the topics of content of this reference book below:

1. The Foundations: Logic and Proofs

This chapter introduces propositional (sentential) logic, predicate logic, and proof theory at a very introductory level. It starts by introducing the propositions of propositional logic (!), then go on to introduce applications of propositional logic, such as logic puzzles and logic circuits. It then goes on to introduce the idea of logical equivalence between sentences of propositional logic, before introducing quantifiers and predicate logic and its rules of inference. It then ends by talking about the different kinds of proofs one is likely to encounter – direct proofs via repeated modus ponens, proofs by contradiction, proof by cases, and constructive and non-constructive existence proofs.

This chapter illustrates exactly why this book is excellent as an introductory text. It doesn't just introduce the terms, theorems, and definitions; it motivates them by giving applications. For example, it explains the need for predicate logic by pointing out that there are inferences that can't be drawn using only propositional logic.

2. Basic Structures: Sets, Functions, Sequences, Sums, and Matrices

This chapter introduces the different objects one is likely to encounter in discrete

mathematics. Most of it seemed pretty standard, with the following exceptions: functions are introduced without reference to relations; the “cardinality of sets” section provides a high level overview of a lot of set theory; and the matrices section introduces zero-one matrices, which are used in the chapters on relations and graphs.

3. Algorithms

It starts by introducing the notion of algorithms, and gives a few examples of simple algorithms. It then spends a page introducing the halting problem and showing its undecidability. (!) Afterwards, it introduces big-o, big-omega, and big-theta notation and then gives a (very informal) treatment of a portion of computation complexity theory. It's quite unusual to see algorithms being dealt with so early into a discrete math course, but it's quite important because the author starts providing examples of algorithms in almost every chapter after this one.

4. Number Theory and Cryptography

This section goes from simple modular arithmetic (3 divides 12!) to RSA, which I found extremely impressive. After introducing the notion of divisibility, the book takes the reader on a rapid tour through base-n notation, the fundamental theorem of arithmetic, the infinitude of primes, the Euclidean GCD algorithm, Chinese remainder theorem, Fermat's little theorem, and other key results of number theory. It then gives several applications of number theory: hash functions, pseudorandom numbers, check digits, and cryptography. The last of these gets its own section, and the book spends a large amount of it introducing RSA and its applications.

5. Induction and Recursion

This chapter introduces mathematical induction and recursion, two extremely important concepts in computer science. Proofs by mathematical induction, basically, are proofs that show that a property is true

of the first natural number (positive integer in this book), and if it is true of an integer k it is true of $k+1$. With these two results, we can conclude that the property is true of all natural numbers (positive integers). The book then goes on to introduce strong induction and recursively defined functions and sets. From this, the book then goes on to introduce the concept of structural induction, which is a generalization of induction to work on recursively-defined sets. Then, the book introduces recursive algorithms, most notably the merge sort, before giving a high level overview of program verification techniques.

6. Counting

The book now changes subjects to talk about basic counting techniques, such as the product rule and the sum rule, before moving on to the pigeonhole principle. It then moves on to permutations and combinations, while introducing the notion of combinatorial proof, which is when we show that two sides of the identity count the same things but in different ways, or that there exists a bijection between the sets being counted on either side. The textbook then introduces binomial coefficients, Pascal's triangle, and permutations/combinations with repetition. Finally, it gives algorithms that generate all the permutations and combinations of a set of n objects.

7. Discrete Probability

In this section the book covers probability, it begins by introducing the notion of sample spaces and events as sets, before defining probability of an event E as the ratio of the cardinality of E to the cardinality of S . We are then introduced to other key concepts in probability theory: conditional probabilities, independence, and random variables, for example. The textbook takes care to flesh out this section with a discussion about the Birthday Problem and Monte Carlo algorithms. Afterwards, we are treated to a

section on Bayes theorem, with the canonical example of disease testing for rare diseases and a less-canonical-but-still-used- quite-a-lot example of Naïve Bayes spam filters. The chapter concludes by introducing the expected value and variances of random variables, as well as a lot of key results (linearity of expectations and Chebyshev's Inequality, to list two).

8. Advanced Counting Techniques

This chapter, though titled "advanced counting techniques", is really just about recurrences and the principle of inclusion-exclusion. I found this chapter quite helpful nevertheless.

We begin by giving three applications of recurrences: Fibonacci's "rabbit problem", the Tower of Hanoi, and dynamic programming. We're then shown how to solve linear homogenous relations, which are relations of the form

$$a_n = c_1 a_{n-1} + c_2 a_{n-2} + \dots + c_k a_{n-k} + F(n)$$

Where c_1, c_2, \dots, c_k are constants, $c_k \neq 0$, and $F(n)$ is a function of n . The solutions are quite beautiful. Afterwards, we are introduced to divide-and-conquer algorithms, which are recursive algorithms that solve smaller and smaller instances of the problem, as well as the master method for solving the recurrences associated with them, which tend to be of the form

$$f(n) = a f(n/b) + cn^d$$

After these algorithms, we're introduced to generating functions, which are yet another way of solving recurrences.

9. Relations

Relations are defined as sets of n -tuples, but the book also gives alternative ways of representing relations, such as matrices and directed graphs for binary relations. We're then introduced to transitive closures and Warshall's algorithm for computing the

transitive closure of a relation. We conclude with two special types of relations: equivalence relations, which are reflexive, symmetric, and transitive; and partial orderings, which are reflexive, anti-symmetric, and transitive.

10. Graphs

A graph is defined as a set of vertices and a set of edges connecting them. Edges can be directed or undirected, and graphs can be simple graphs (with no two edges connecting the same pair of vertices) or multigraphs, which contain multiple edges connecting the same pair of vertices. We're then given a ton of terminology related to graphs, and a lot of theorems related to these terms. The treatment of graphs is quite advanced for an introductory textbook – it covers Dijkstra's algorithm for shortest paths, for example, and ends with four coloring. I found this chapter to be a useful review of a lot of graph theory.

11. Trees

Thebook gives a lot of examples of applications of trees, such as binary search trees, decision trees, and Huffman coding. We're then presented with the three ways of traversing a tree – in-order, pre-order, and post-order. Afterwards, we get to the topic of spanning trees of graphs, which are trees that contain every vertex in the graph. Two algorithms are presented for finding spanning trees – depth first search and breadth first search. The chapter ends with a section on minimum spanning trees, which are spanning trees with the least weight. Once again we're presented with two algorithms for finding minimum spanning trees: Prim's Algorithm and Kruskal's algorithm. I found this section to be quite interesting, though they are given a more

comprehensive treatment in most introductory algorithms textbooks.

12. Boolean Algebra

This section introduces Boolean algebra, which is basically a set of rules for manipulating elements of the set $\{0,1\}$. Boolean algebra is directly related to circuit design! This book first introduces the terminology and rules of Boolean algebra, and then moves on to circuits of logic gates and their relationship with Boolean functions.

13. Modeling Computation

This chapter covers phase structure grammars, finite state machines, and closes with Turing machines. However, I found this chapter a lot more poorly motivated than the rest of the book.

If you're not familiar with discrete mathematics, this is a great book that will

get you up to speed on the key concepts, at least to the level where you'll be able to understand the other textbooks. I think that Rosen is hands down the best. However, I think that people familiar with the topics probably should look for other books, especially if they are looking for textbooks that are more concise. It might also not be suitable if you're already really motivated to learn the subject, and just want to jump right in. In general, the rule for the textbook is: read the sections you're not familiar with, and skim the sections you are familiar with, just to keep an eye out for cool examples or theorems.

I'd highly recommend it to anyone who has not had any exposure to discrete mathematics, and I think it's an important prerequisite for the rest of the books.