GRAPH AND TREE TRANVERSAL

(R.RAMYA,P.PRIYA,Dr.S.SANGEETHA,M.MAHALAKSHMI)
(priyaperiyasammy13@gmail.com,sangeethasankar2016@gmailcom)
Department of Mathematics
Dhanalakshmi Srinivasan College of
Arts and Science for Women (Autonomous)
Perambalur

ABSTRACT

Graph representation in computers has always been a hot topic thanks to the amount of applications that directly enjoy graphs. Multiple methods have emerged in computing to represent graphs in numerical/logical formats; most current business applications also rely heavily on relational databases as a primary source of storing information.

KEYWORDS

Loop, finite, infinite, dots, vertices, edges, BFS, DFS

INTRODUCTION

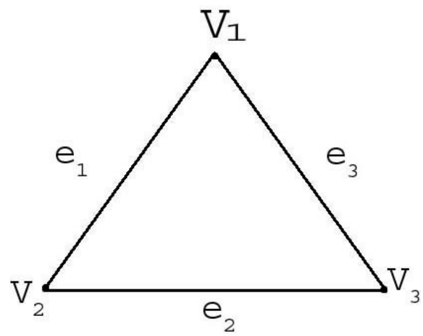
Graph theory in mathematics mean the study of graphs. Graphs are **one among** the prime objects of study in discrete mathematics. In general a graph is represented as a set of vertices (notes or points connected by line). Graphs are therefore mathematical structures **wont to** model pair wise relations between object .They are fount on road maps, constellations, when constructing schemes and drawing.

GRAPH

Definition 1.1

A Graph G consists of a pair sets V (G), E (G)
Where V (G) is a non empty finite set. Where element are called points (or) Vertices.Where E (G) is a set of unordered pair of distinct elements of V (G) whose elements are called lines (or) edges.

Example



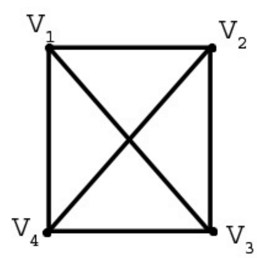
3 - Vertices

3 - Edges

Definition 1.2

A graph with a finite number of vertices as well as finite number of edges is called finite graph otherwise it is said to be infinite graph.

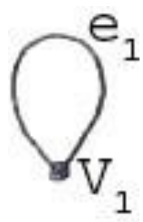
Example

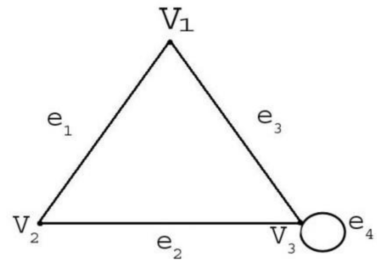


Definition 1.3

A line joining a point to itself is called a loop (or)
An edge with same end vertices is known as a loop (or) self loop.

Example



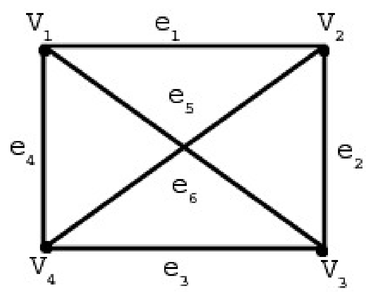


Definition 1.4

A graph in which there exists on edge between every pair of vertices is called a complete graph.

A complete graph with P points and it is denoted by Kp

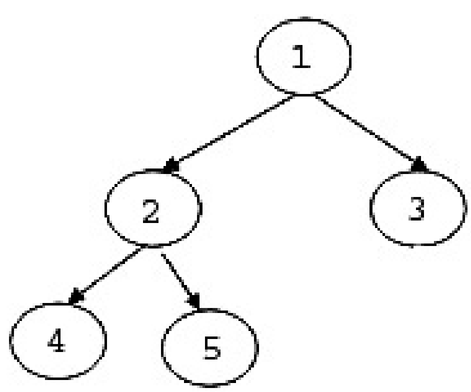
Example



Definition:

Level order traversal of a tree s breadth first traversal for the tree.

Example



Level order traversal of the tree is 1 2 3 4 5.

Traversing a Graph

Generally traversing a graph means visiting all the vertices of the graph exactly once. Graph traversal can start from any vertex of the graph. Since a vertex may have more than one adjacent vertex, the developer should take extra care ‘not to visit a vertex multiple times. Several graph traversing algorithms have been proposed. Out of them two are accepted as standards and discussed below. Both the traversal algorithms have the same purpose

These algorithms are:

(1) DFS (Depth – First Search)

(2) BFS (Breadth – First Search)

Common Traversal Steps

Step 1: Start from any vertex of a graph.

Step 2: From this starting vertex, traverse as deep as you can go. Whenever **you can't** go further, then backtrack one vertex and do **an equivalent** traversal from this vertex until **you can't** traverse further, and so on.

Step 3: Process the information contained in that vertex.

Step 4: Then move along an edge to process a neighbor (adjacent vertex).

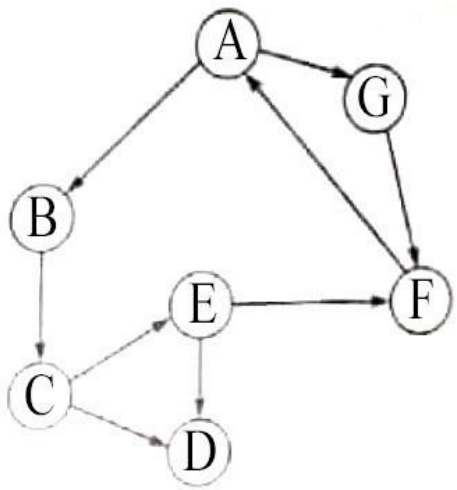
Step 5: When the traversal finishes, all the above vertices that can be reached from the start vertex are processed.

It may be noted that all the traversal algorithms require the systematical examination of the nodes and edges of the Graph.

Example: 2.2

Consider the graph we start with the vertex A. Then, we visit vertex G as it is adjacent to A. (However, it may be noted that one can visit B instead of G as the path of the traversal is not unique). From G, we visit to vertex F. Then, we back track to A, as there is no path from F to any vertex other than A. we visit the next adjacent vertex B. then from B to C, then C to E. and finally from E to D. As there is no path from D to any other vertex which is not visited, the algorithms stops at this point.

In order to keep track of the adjacent vertices, we use stack as the data structure.



An example showing depth – first search

List Visited

A	G	F				
A	G	F	B			
A	G	F	B	C		
A	G	F	B	C	E	
A	G	F	B	C	E	D

In may be noted that, a stack is an ordered collection of homogeneous data elements where the insertion and deletion operations take place at one end only. The insertion operation **is named** PUSH **and therefore the** deletion operation **is named** POP.

Now, we present the detailed traversal steps.

Breadth – First Search

The general idea behind a breadth – first search beginning at a starting node A is as follows. First we have to examine the starting node (Say A). Then we examine all the neighbors of A. Then **we've to look at** all the neighbors of the neighbors of A **then** on. Generally **we've to stay** track of the neighbors of a node and care must be taken that no node is visited or processed **quite** once. This can be done by using a data structure QUEUE to keep track of the nodes visited.

Steps

Step 1: First choose a starting vertex.

Step 2: Find all the vertices which are connected to the starting vertex.

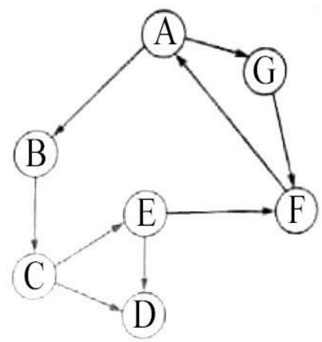
Step 3: Then choose one of the connected vertices and find all the vertices that are connected to this vertex.

Step 4: Continue this procedure until all the vertices are visited.

The following example illustrates the breadth – first search technique.

Example 2.3

The graph first, we start with vertex A. Then, we visit the vertices in the next level, i.e. vertices B and G. Next, we visit B’s adjacent vertex C, and G’s adjacent vertex F. Finally, we visit D and E, the adjacent vertices of C.



List Visited

A	G	F				
A	G	F	B			
A	G	F	B	C		
A	G	F	B	C	E	
A	G	F	B	C	E	D

In order to keep track of the vertices at a level, we use queue as the data structure.

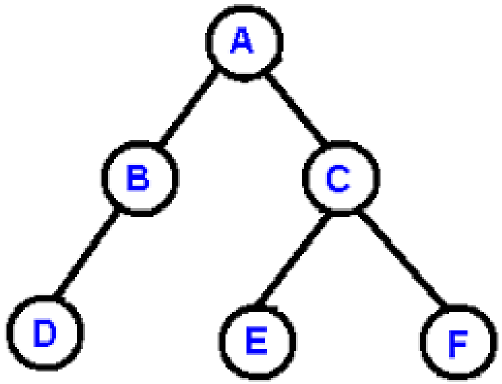
- It may be noted that, like stack, a queue is an ordered collection of homogeneous data elements, but in contrast to the stack, in case of the latter, insertion and deletion operations take place at two extreme ends.

Now, we present the detailed traversal steps of the breadth – first search for the above example.

Tree Traversals

Definition:

A binary tree is a tree in which there is only one vertex of degree 2 and all other vertices are of degree 1 (or) degree 3.



We can use trees to store information in a computer. Therefore some procedures are needed for a accessing the information easily (or for visiting each vertex easily).

BFS and DFS provide ways to walk a true, that is traverse a tree in a systematic way so that each vertex is visited exactly once. Now, we consider three additional tree traversal methods. We define these traversals recursively. They are:

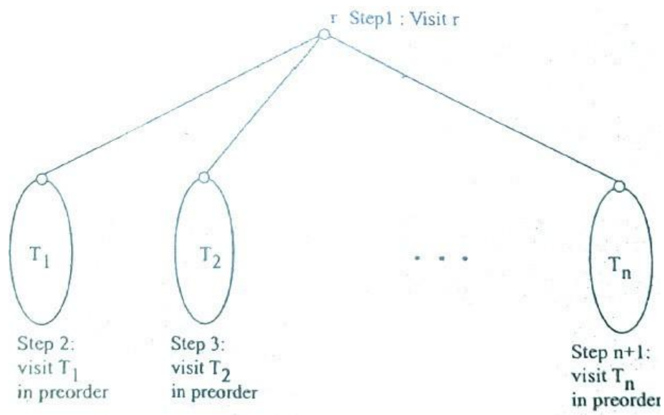
(1) Pre – order traversal

(2) In – order traversal

(3) Post – order traversal

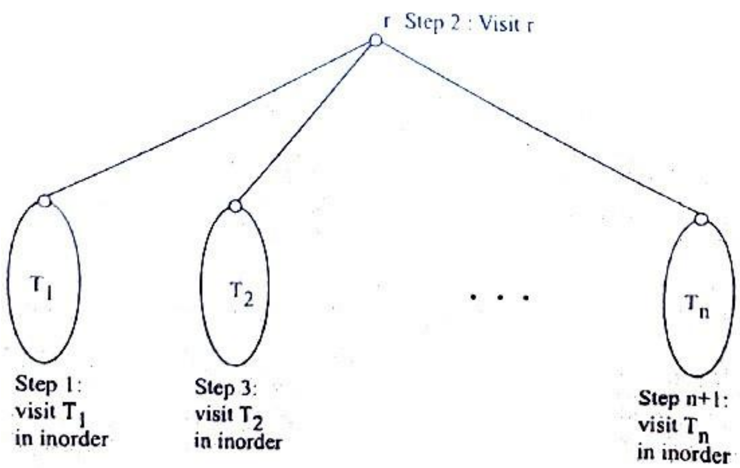
Pre – order traversal

If T is an ordered rooted tree with root r. If T consists **just for** r, then r **is that the** preorder traversal of T. Otherwise suppose that T1, T2,…,Tn are the subs tree at r from left to right in T. The preorder traversal begins by visiting r. It continues by traversing T1 in preorder, then T2 in preorder **then** on until Tn is traversed in preorder.



In order traversal

Let T be an ordered rooted tree with r. If T consists only of r, then r **is that the so as** traversal of T. otherwise suppose

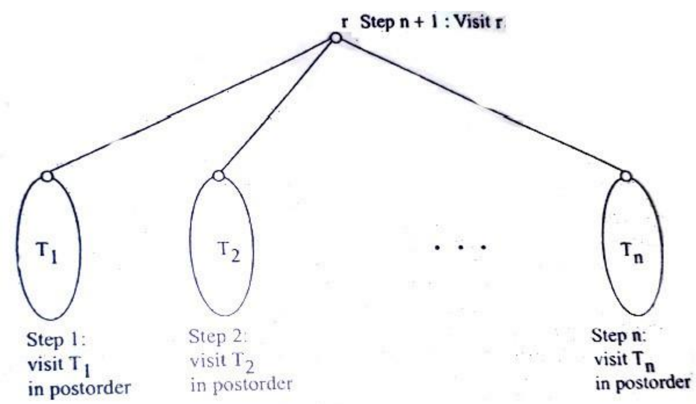


The in order traversal begins by traversing T1 in order, then visiting r. It continues by traversing T2 in order, then T3 in order, and finally T_n In- order.

In order traversal: Visit left most sub tree, visit root, and visit other sub tree left to right.

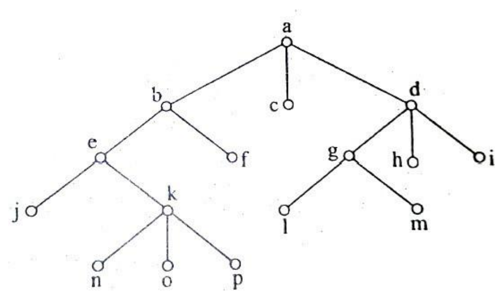
Post order traversal

Let T be an ordered rooted tree with root r. If T consists only of r, then r **is that the** post order traversal of T.. Otherwise, suppose that T_1, $T_{2,}$....T_n are the sub trees at r from left to right. The post order traversal begins by traversing T_1 In post order, then T_2 in post order … then T_n in post order, and ends by visiting r.

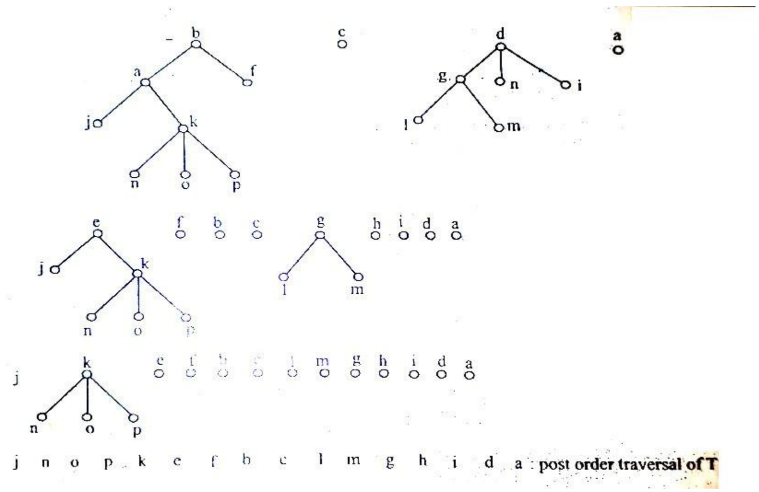


Example: 2.5

Visit the Tree in the post order.



Solution



CONCLUSION

This paper gives **a thought** about traversal techniques like BFS and DFS and also the complexities involved. An idea about graph, tree **and therefore the** different algorithms **employed by** graph are discussed here. While traversing a graph it is important to keep a note on time and space complexity. A proper understanding about complexity will given an idea to the author about different traversals

REFERENCES

1. Graphs and Application an Introductory Approach Joan M.Aldous and Robin J.Wilson
2. Discrete mathematics with graph theory & combinatorial T.veerarajan
3. Breadth first search Wikipedia
4. Depth first search Wikipedia