0

Contents

1 INTRODUCTION

A tree is a mathematical structure that can be viewed as either a graph or as a data structure.Trees were first studied by Cayley in 1857.

A graph is a drawing or diagram consisting of a collection of vertices (dots or points)together with edges(lines) joining certain pairs of these vertices. If 'v' is a vertex of graph, then the number of times V is an end vertex of an edge is called the degree, d(v). A non-trivial closed trial in a graph is called a cycle if it's origin and internal vertices are distinct. A graph is called acyclic if it contains no cycles and such a connacyclic graph is called a 'TREE'. Trees form one of the most widely used subclass of graphs.

This project contains four chapters. In the first chapter, we discuss about the definitions and properties of trees. The second chapter deals with the different types of trees. Third chapter includes binary trees and it's characteristics also the fourth chapter is all about tree traversal and applications of trees.

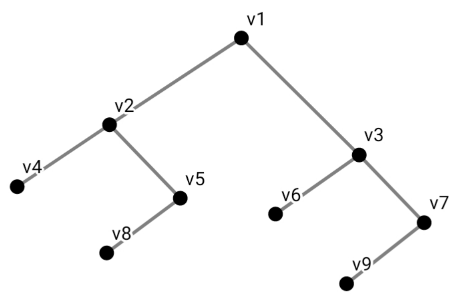
Chapter 1

2 DEFINITIONS AND PROPERTIES OF TREES

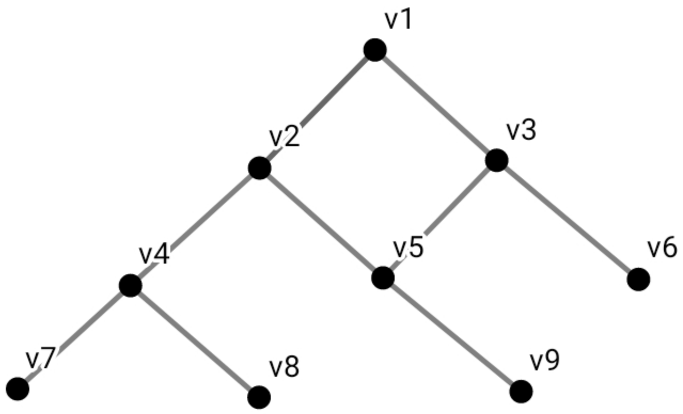
Trees form one of the most widely used subclass of graphs. Tree occures in situations where many elements are to be organised into some sort of hierarchy. In this chapter, we introduce the basic terminology and properties of trees.

Let G(V,E) be a loop-free undirected graph. The graph G is called tree if it is connected and without cycles (loops).

Consider the following graphs



G1



G2

Here graph G_1 is a tree, but graph G_2 is not a tree, because it contains the cycle $\{V_1,V_2\}$, $\{V_2,V_5\}$,$\{V_5,V_3\}$, $\{V_3,V_1\}$.

Trees are also defined as a non-empty finite set of elements called nodes(or vertices) V_i, partitioned into three disjoint subsets. The first subset contains a single element called the root of the tree. The other two subsets are called left and right subtrees(or branches) of the original tree. Any of these subtrees can be empty.

2.1 Basic Terminology of Tree

Branches: The edge of a tree are known as branches.

Nodes: Elements of tree are called nodes.

Leaf: The nodes without child nodes are called leaf nodes. Or leaf is a vertex of degree 1 in an undirected tree.

2.2 Properties of Tree

The relationship among elements of the original tree satisfy the following properties.

i) Every tree which has atleast two vertices should have atleast two leaves.

ii) Trees have many characterizations

Let T be a graph with n vertices, then the following statements are equivalent.

. T is a tree.

. T contains no cycles and has n-1 edges.

. T is connected graph and has (n-1) edges.

. T is connected graph and every edge is a cut-edge

. Any two vertices of graph T are connected by exactly one path.

. T contains no cycles, and for any new edge e, the graph T+e has exactly one cycle.

iii) Every edge of a tree is cut edge.

iv) Adding one edge to a tree defines exactly one cycle.

v) Every connected graph contains a spanning tree.

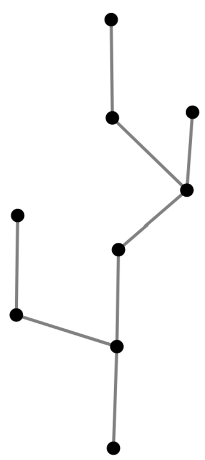
vi) Every tree has atleast two vertices of degree two.

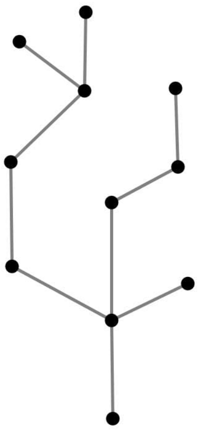
vii) Every pair of vertices is connected by a unique path.

2.3 Forest

In graph theory, a forest is an undirected, disconnected, acyclic graph. In other words, a disjoint collection of trees iis known as forest. Each component of a forest is tree. Any forest consists of a collection of disjoint trees obtained by deleting the root and the corresponding edges connecting the vertices from a given tree.

Example:





The above graph looks like a two subgraphs, but it is a single disconnected graph. There are no cycles in the above graph. Therefore, it is a forest.

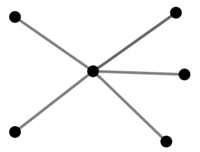
Cayley's Tree Formula

The number of different trees on 'n' labelled vertices is n^{n-2}.

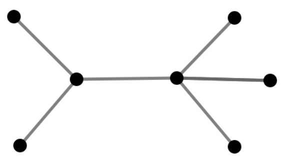
2.4 Centers in Trees

In every tree there will be either one or two vertices are 'innermost' these vertices are called centers of the tree. Centers play an important role in developing recursive algorithms for problems on trees.

For example,



T1



T2

Here T_1 has an obvious centre while T_2 does not.

Chapter 2

3 TYPES OF TREES

Trees are graphs that do not contain even a single cycle. They represent heirarchial structure in a graphical form. Trees belongs to the simplest class of graphs. Despite their simplicity, they have a rich structure.

Trees provide a range of useful applications as simple as a family tree to as complex as trees in data structures of computer science.

There are several types of graphs.

Example:
. A path graph(or linear graph): Consist of n vertices arranged in a line, so that vertices i and i+1 are connected by an edge for i=1,2,...n-1
. A starlike graph: It consists of a control vertex called root and several path graphs attached to it. More formally, a tree is starlike if it has exactly one vertex of degree greater than 2.
. Star tree: It is a tree which consists of a single internal vertex (n-1 leaves), in other words, a star tree of order n is a tree of order n with as many leaves as possible.
. Caterpillas tree: It is a tree which all vertices are within distance one of a central path subgraph.
. Lobster tree: It is a tree which all vertices are within distance 2 of a central path subgraph.

. A regular tree of degree d is the infinite tree with d edges at each vertex. These arise as the Cayley graphs of free groups and in the theory of its buildings.

3.1 Classification of Trees

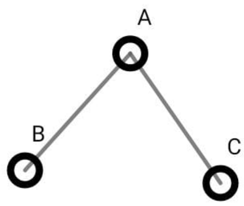
Mainly trees are classified into 3.
1. Ordered tree
2. Binary tree
3. Rooted tree

3.2 Ordered Tree

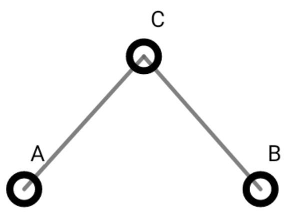
An ordered tree(plane tree) is a rooted tree in which an ordering is specified for the children of each vertex. This is called a 'plane tree' because, an ordering of the children is equivalent to an embedding of the tree in the plane, with the root at the top and the children of each vertex lower than that vertex. Given an embedding of a rooted tree in the plane, if one fixes a direction of children,say left to right then an embedding gives an ordering of the children. Conversely, given an ordered tree and conventionaly drawing the root at the top, then the child vertices in an ordered tree can be drawn left to right, yielding an essentially unique planar embedding. An ordered tree is an oriented tree in which the children of a node are somehow ordered. It is a rooted tree in which an ordering is specified for the children of each vertex. The ordered trees can be further specified as labelled ordered trees and unlabelled ordered trees.

Labelled ordered trees

A labelled ordered tree is a tree where each vertex is assigned a unique number from 1 to n.



T1



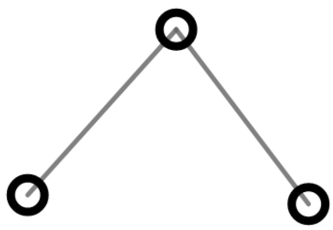
T2

If T_1 and T_2 are ordered trees, then $T_1 != T_2$ else $T_1 = T_2$.

Unlabelled ordered trees

An unlabelled tree is a tree where every vertex is unlabelled. Given below are the possible unlabelled ordered tree having three vertices.

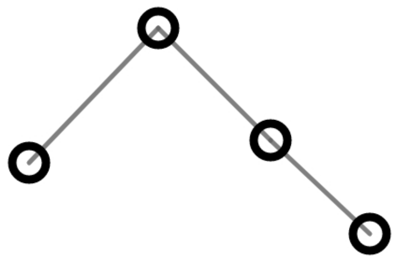
T1

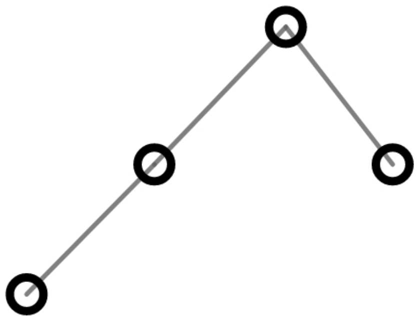


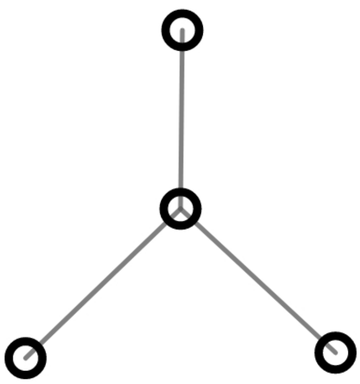
T2

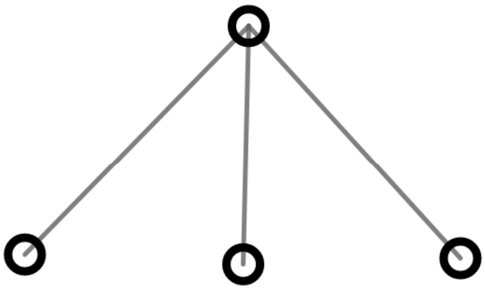
The total number of unlabelled ordered trees having n nodes is equal to the (n-1)th catalan number.

Given below are the possible unlabelled ordered trees having 4 nodes. This diagram will work as a reference example for the next few results.









Some results

1) Number of trees with exactly K leaves.
Let us consider that we have 'n' edges. Then, the solution for the total possible ordered trees having K leaves is given by

$$L_n(k) = \frac{1}{n}\binom{n}{k}\binom{n}{k-1}$$

2) Total number of nodes of degree d in these trees.
Let us consider that we have 'n' edges. Then the solution for the total number of nodes having a degree 'd' is given by

$$D_n(d) = \binom{2n-1-d}{n-1}$$

3) Number of trees in which the root has degree r.
Let us consider that we have 'n' edges. Then, the solution for the total possible ordered trees where root has degree 'r' is given by

$$R_n(r) = \frac{r}{n}\binom{2n-1-r}{n-1}$$

3.3 Binary Tree

A binary tree is a root tree where each vertex V has almost two subtrees. If both subtrees are present, one is called a left subtree of V and the other right subtree of V. If only one subtree is present, it can be designated either as the left subtree or right subtree of V. In other words, a binary tree is a 2-ary tree in which each child is designated as left or right child. In a binary tree, every vertex has two children or no children.

3.4 Rooted Tree

A tree with no vertices is a rooted tree(the empty tree). A single vertex with no children is a rooted tree.

Recursion :

Let T_1, T_2, ...T_r, r $\geq$ 1 be disjoint rooted trees with roots V_1,V_2, ... V_r, respectively, and let V_0 be a vertex that doesn't belong to any of these trees. Then a rooted tree, rooted at V_0, is obtained by making V_0 the parent of the vertices V_1,V_2,...V_r. We call T_1,T_2,...T_r subtrees of the larger tree.

Terminology :

Let T be a rooted tree.
. If there is an edge (u,v) in T, then V is a child of u and u is the parent of V.

. If there is a path of positive length from u to v in T, then V is a descendent of u and u is an ancestor of v.

. A vertex without degree 0 is called a leaf. A vertex that is not a leaf is called an internal vertex.

.If 2 vertices have the same parent, they are called siblings.

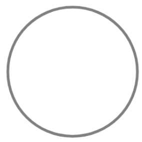
. The subgraph of T containing V and all its descendants is a tree rooted at V. This is called the subtree of T rooted at V.

. For a vertex V, let the length of the path from the root be K, then the depth of V is K.

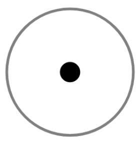
. The height of a vertex is the length of the longest path from it to a leaf. The height of a rooted tree is the height of its root.

Example:

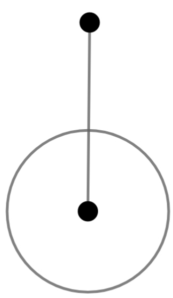
Root:-



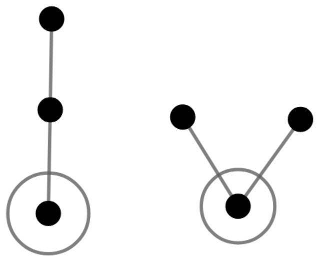
1)



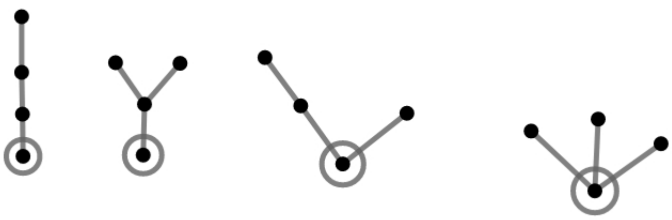
2)



3)



4)



Chapter 3

4 BINARY TREES

Among all three of the different types of trees,binary trees have some special features and a little importance.

A tree is called binary tree if each internal vertex of a rooted tree has almost two children, that is,

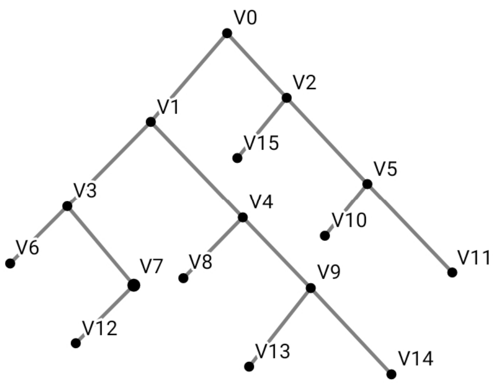
order(v)= 0,1,2

A tree consisting of no nodes is also a binary tree.

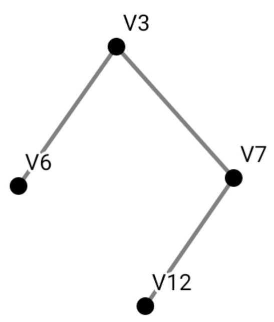
In a binary tree every internal vertex has at most two children.Elder is the left child and other is the right child.

subtree at left child v is left subtree rooted v and the subtree rooted at right child W is the right subtree.

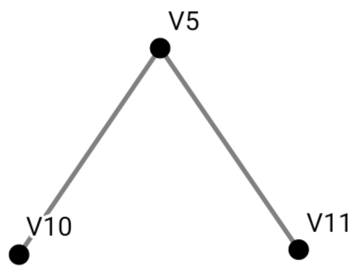
Example for a binary tree :



Left subtree of V1



Right subtree of V2



Left subtree and right subtree can have nodes either be greater, less, or equal to parent's node.

4.1 Properties of a Binary Tree

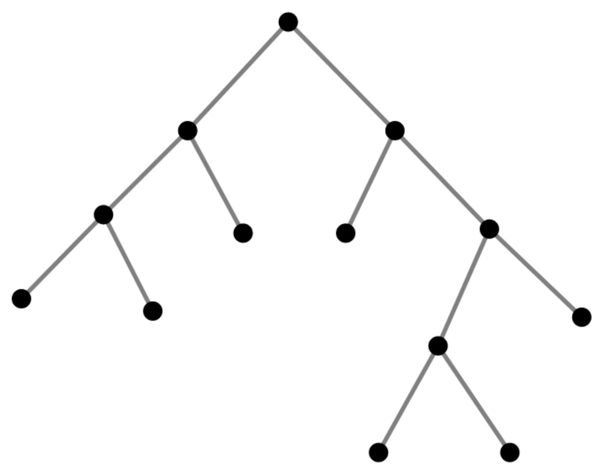
i) The number of vertices in a binary tree is always odd
ii) If the number of vertices (p) in a binary tree of degree 1,then n-p-1 is the number of vertices of degree 3

4.2 Different Types of Binary Trees

Strict / Proper / Full Binary tree

A binary tree is called strict binary tree if each node can have 0 or 2 children

Example :

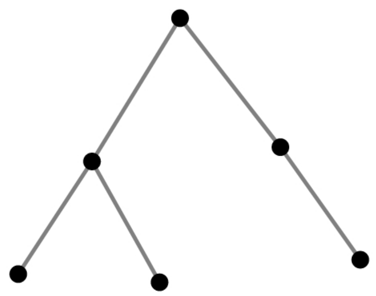


Complete binary tree

A complete binary tree is a binary tree if all level except possibly the left are completely filled and all nodes are as left as possible.

In other words a rooted binary tree of height n is called complete binary tree

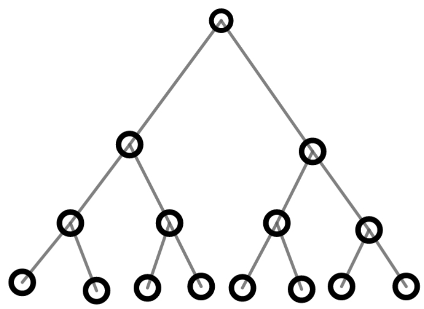
Total number of vertex in a complete binary tree of height (h) equals the sum of number of vertex at each level



level	0	1	2
Vertex	1	2	3

Perfect binary tree

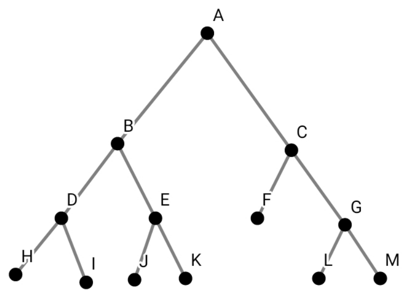
If all the levels of a binary tree is completely filled, then it is known as a perfect binary tree.



Balanced binary tree

A binary tree is called balanced binary tree if absolute difference height of left and right subtree for every node is not more than K(mostly 1)

Difference = |h_{left} - h_{right}|



A=|3-3| = 0

B=|2-2| = 0

C=|1-2| = 1

D=|1-1| = 0

E=|1-1| = 0

F = 0

G=|1-1| = 0

H = 0

I = 0

J = 0

K = 0

L = 0

M = 0

maximum difference in this graph is 1

4.3 Tree of an Algebraic Expression

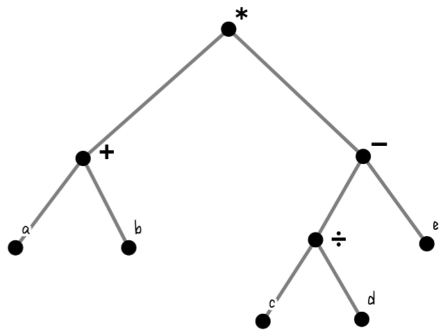
An algebraic expression E containing operands and binary operators can be represented by using a binary tree called expression binary tree. The root of tree contains an operator that is to be applied to the result of evaluating the expressions represented by left and right subtrees. A vertex representing an operator is a non-leaf, whereas a vertex representing an operand is a leaf.

In other words, depending upon the precedence of evaluation, an algebraic expression involving binary operators can be considered as

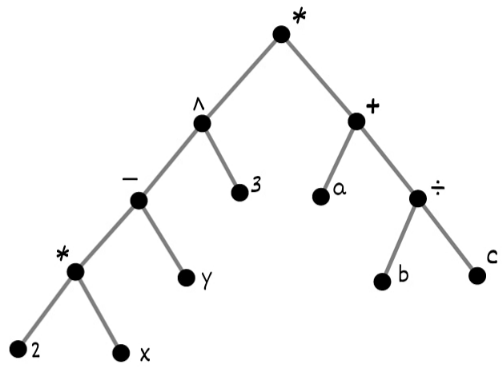
... left operand $\Leftrightarrow$(operator) $\Leftrightarrow$right operand...

Example :

$E = (a+b)*((c \div d)-e)$



$E=(2x-y)^3 * (a+(b \div d))$



4.4 Relationship Between General Trees and Binary Trees

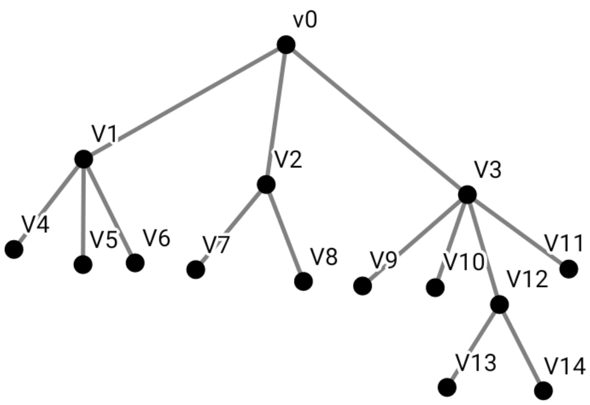
A general tree T_g can be converted into a unique binary T_b tree as follows

(i) The number of nodes in a binary tree T_b is same as in general tree T_g

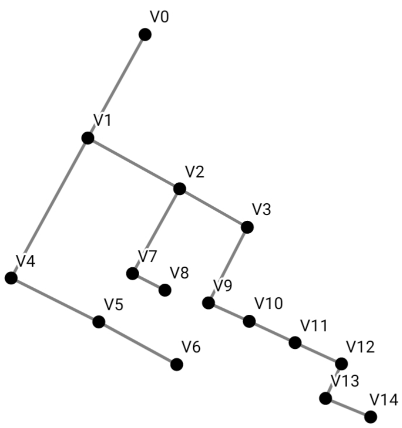
(ii) The root vertex of T_b is also the root vertex of T_g

(iii) The first child (from left) of root vertex V_0 in T_g is the left child of the root vertex in T_b, and right child root vertex in $T_b$is the sibling of root vertex in T_g

(iV) Repeat steps (ii) and (iii) for each new vertex of T_g till T_b is drawn.
Example :



General Tree



Rooted Tree

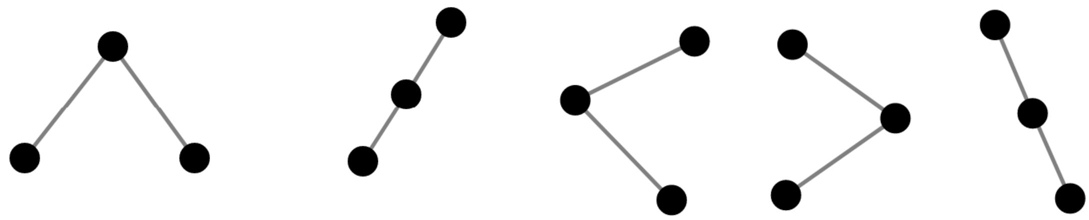
4.5 Counting Binary Trees

let b_n denote the number of binary trees on n vertices, then

$$b_n = b_0\, b_{n-1} + b_1\, b_{n-2} + \ldots + b_{n-1}\, b_0$$

This recurrence relation is known as the Catalan recursion and the quantity b_n is called the n^{th} Catalan number.

* Five different binary trees on 3 vertices :



4.6 Fibanocci Trees

A special class of binary trees, and their close relationship with Fibanocci numbers are said to be Fibanocci trees.

Fibanocci trees are defined recursively as :-
Both T_1 and T_2 are binary trees with exactly one vertex each ; and

T_n is a binary tree with left subtree T_{n-1} and right subtree T_{n-2} where n>=3

Chapter 4

5 TREE TRAVERSAL AND APPLICATIONS

5.1 Traversals

A traversal is a process that visits all the node in a tree. Tree or nonlinear data structure, there is no unique traversal.

There are reversal traversal algorithms with group in 2 types :

1). Depth - First traversal
2). Breadth - First traversal

Breadth Traversal

It is the level order traversal. Visits the nodes by levels from top to bottom and from right to left.

Depth Traversal

Depth traversal is divided into three.

1) Pre order traversal

Visit the parent first and then left and right children.

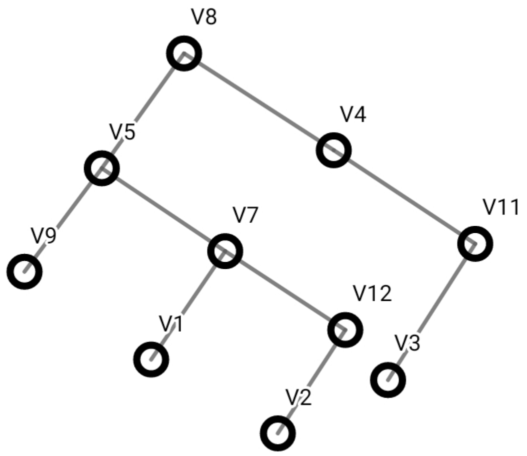
2) In order traversal

In order traversal is the visit of the left child then the parent and the right child.

3) Post order traversal

Post order traversal is the visit of the left child then two right child and then the parent.

Example based on figure :



Pre order : v_8,v_5,v_9,v_7,v_1,v_{12},v_2,v_4,v_{11},v_3

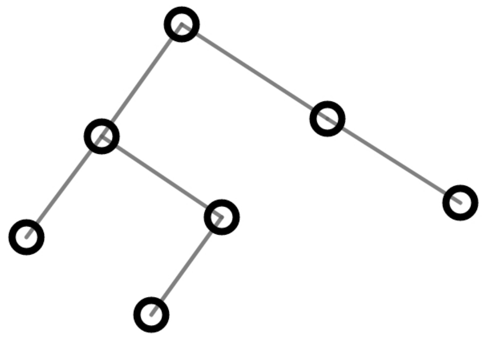
In order : v_9,v_5,v_1,v_7,v_2,v_{12},v_8,v_4,v_3,v_{11}

Post order : v_9,v_1,v_2,v_{12},v_7,v_5,v_3,v_{11},v_4,v_8

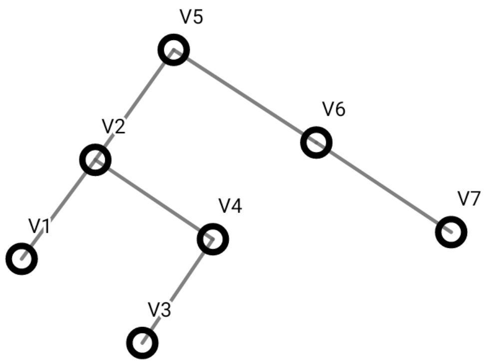
Level order : v_8,v_5,v_4,v_9,v_7,v_{11},v_1,v_{12},v_3,v_2

5.2 Order of node visitance

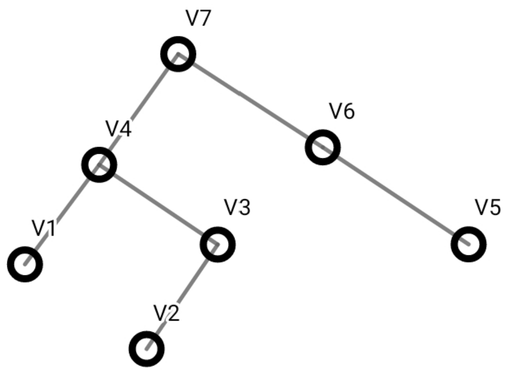
Pre-order



In order



Post order



An Euler tour is a walk around the binary tree where each edge is treated as a walk, which you cannot cross. In this walk each node will be visited either on the left or under the below or on the right
The Euler tour in which we visit nodes on the left produces a pre order traversal. Then we visit nodes from the below we get an in order traversal and when we visit nodes on the right we get a poat order traversal.

5.3 Non Recursive Traversal

Depth - first traversal can be easily implemented recursively. An non-recursive implementation is a fit move difficult. In this section we implemented a pre order traversal as a tree traversal.

5.4 Level order Traversal

Level order traversal process is the nodes level by level. It first processes the root and then it's children, then it's grand children, and so on. Unlike the order traversal methods the recursive version does not exist.
A traversal algorithm is similar to the non recursive pre order traversal algorithm. The only difference is that a stack is replaced with a FIFO queue.

5.5 Applications of Trees

Trees form one of the most widely used sub class of graph. Because many of the applications of graph theory directly or indirectly involve trees. Tree occur in situations where many elements are to be oraganized into some sort of hierarchy.

In computer science trees are useful in organizing and storing data in data base. There are several different form of trees and can be applied for solving a wide variety of problems in computer science, applications such as, organizing and sorting of data in data base, coding theory, language compilers...In addition trees are employed in a wide range of algorithms.

The wide application of m-ary trees embraces the general decision making process. The decisions tree is a hierarchical tree structure or a diagram, it helps us to chose between various actions. It is mostly used for decision making purpose. It is a type of rooted tree in which each internal vertex corresponds to a decision. These vertices contain a sub tree for each possible out come of the decision. The paths to leaves vertex corresponds to the possible solution to the problem.

Tree can also be used in probability and in calculating routes and path lengths in networks. Therefore trees are helpful tools in many branches of both discrete mathematics and computer science.

6 CONCLUSION

This project presents a brief overview of trees in graph theory.Many applications of graph theory depends on tree. Properties of trees were discussed in details.It also includes the characteristics of binary tree and the process of tree traversal. we also illustrated the applications of trees in both Mathematics and Computer Science.

In this project we discussed about different types of trees ,such as ordered trees,rooted trees and about various types of binary trees in detail. It also looked into the characteristics of tree traversal and different types of traversals. Overall,tree has an important role in both Mathematics and Computer Science.It is also an active area of research,with ongoing efforts to develop new techniques with trees.

we hope that this project has provided you with a solid understanding of trees in graph theory and has inspired you to further explore the fascinating world of graph theory.

7 BIBLIOGRAPHY

1.A.Cayley,A Theorem on trees,Quart.J,Math

2.Edgar G.Goodaire,Michael M.Parmenter,Discrete Mathematics with Graph Theory 3rd Edition,2015

3.Robin J.Wilson,Introduction to Graph Theory,1979

4.Seymour Lipschutz,Marc Lars Lopson,Discrete Mathematics Rivised third edition,McGraw Hill Education;2017