Unit 5 : Trees & Graphs.

(weightage - 16 marks).

(Data structures using 'C' - 22317)

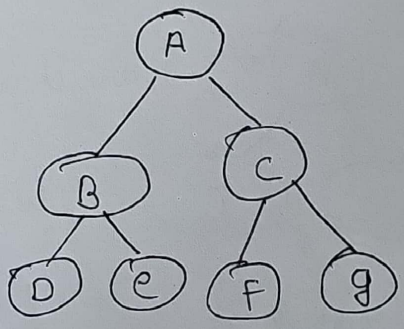
Ish Chaniyara
(SY-COMPS)

5.1 Terminologies:

Tree

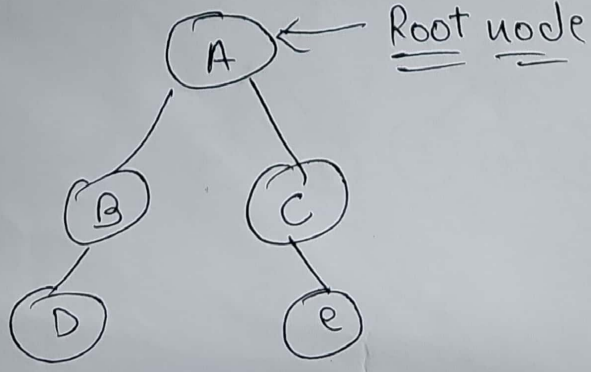
1) Tree is a non-linear Data structure
2) It is very flexible & powerful Data structure.
3) Use to represent hierarchical relationship among data items
4) Data item arranged in a sorted sequence.

eg-



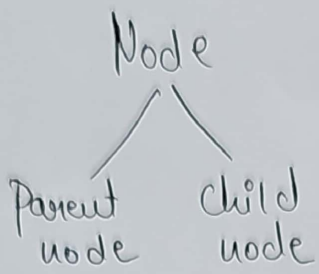
Root

- In a tree data structure, the first node is called as Root Node.
- Every tree has a root node.
- It is the origin of the tree in data structure.
- There can be only one root node.



Node

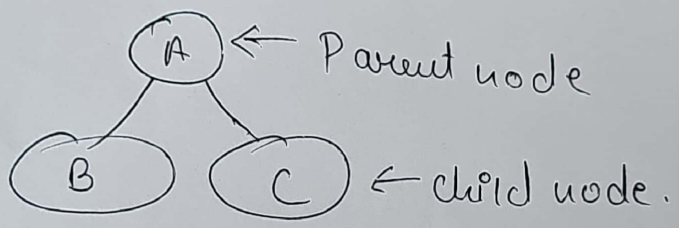
Every elements in the tree is called as Node.



Parent node: the node which is predecessor of any node is called as Parent node. In simple words, the node which has a branch from it to any other node is called a parent node

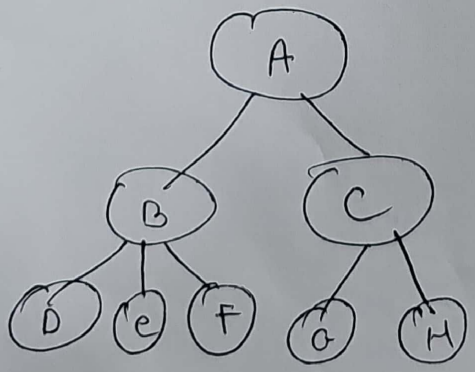
Child node: The node which is dicendant of any node is called as child node. The node which has a link from a parent is known as child node. All nodes expect root are child node.

eg.



Degree

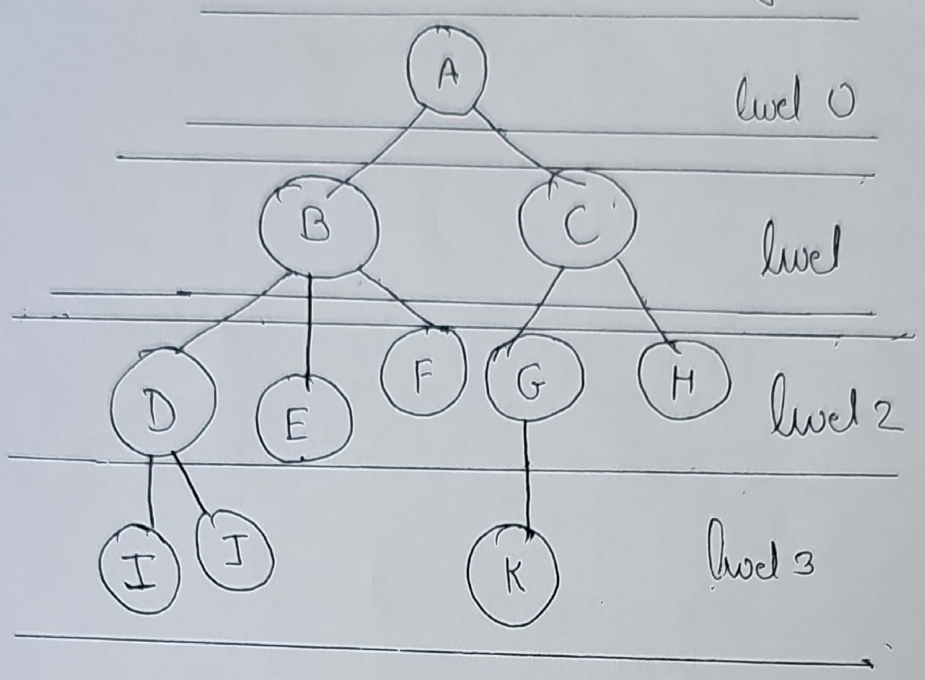
The total no of childrens of a node is called as Degree of that node. The heighest degree of a node among all the nodes in the tree is called Degree of the tree.



Here degree of B = 3
degree of A = 2
degree of F = 0

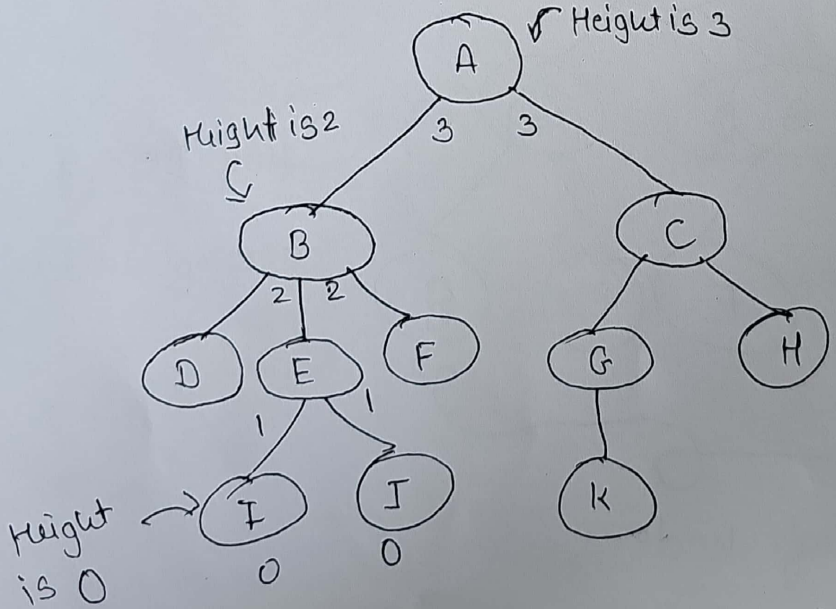
level

In a tree, each step from top to bottom is called as a level & level counts starts with '0' & increment by one at each level



Height

- The total no of edge from ~~edge~~ leaf node to a particular node in the longest path is called as height of that node.
- height of the root node is said to be height of the tree

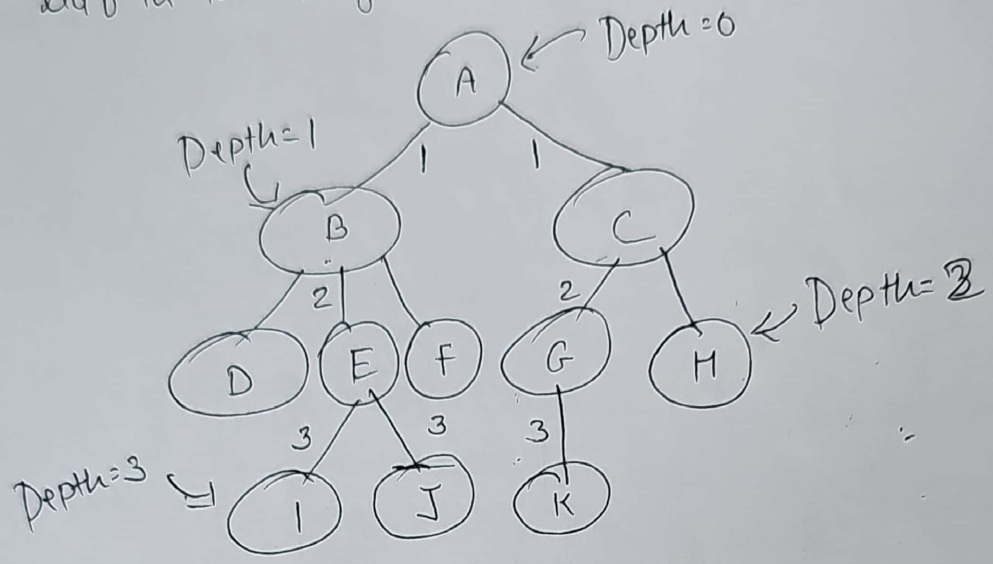


height of leaf node = 0

height of tree = 3

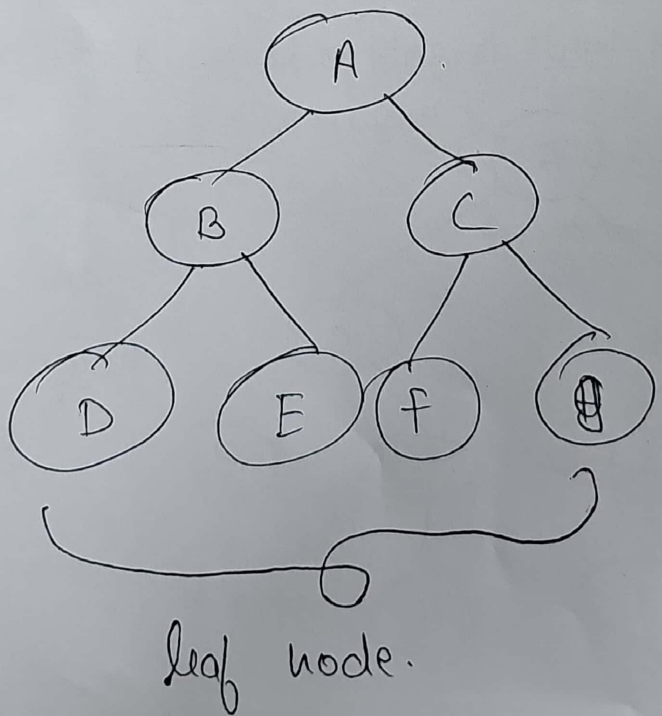
depth

- Depth of Node is total no of edges from root to that node
- Depth of the tree is total no of edges from root to leaf in the longest path



leaf

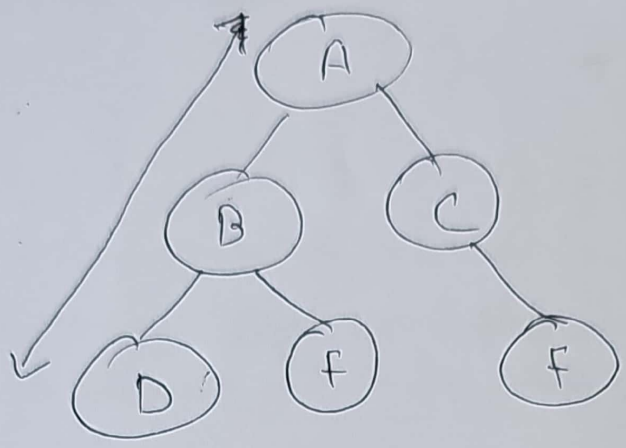
- Trees in the data structure, the node with no child, is known as a leaf node.
- leaf node are called as external node or terminal nodes.



here D, E, F, g are leaf nodes.

Path

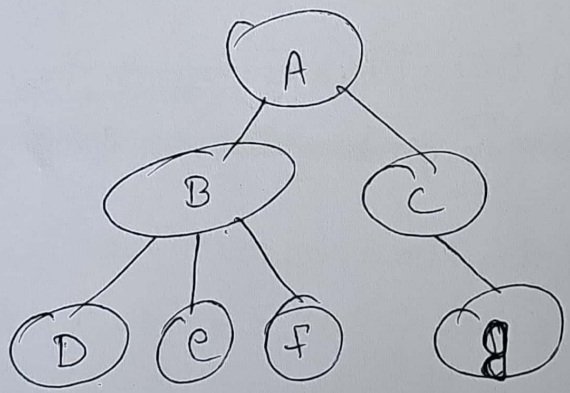
- The sequence of node and edges from one node to another node is called as the path between those two nodes
- The length of the path is the total no of nodes in path, ex.



Here the path between A-D is: A-B-D.

Ancestor node

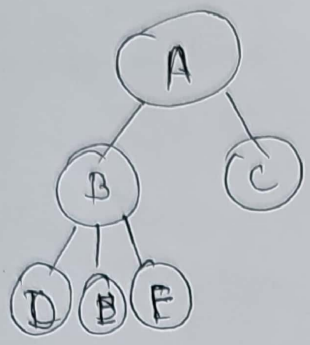
- An ancestor of a node is any predecessor node on a path from the root to that node. The root doesn't have any ancestor



A & B are ancestor of node D.

Descendant node

- The immediate successor of the given node known as a descendant of a node.



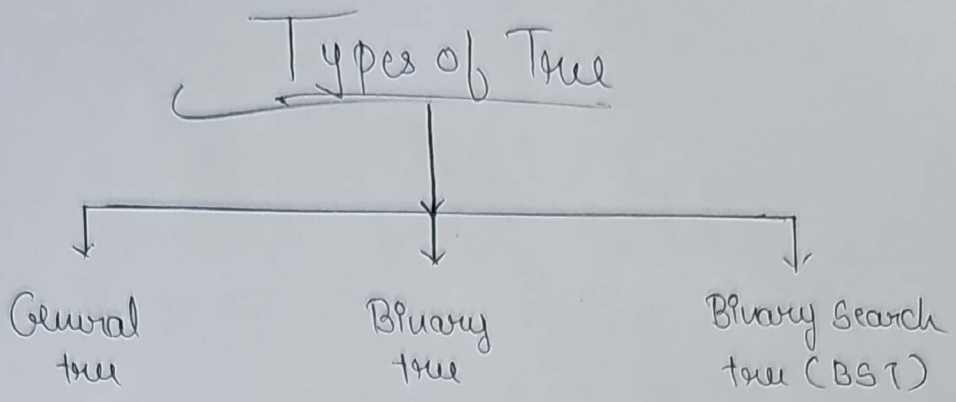
Here, D is the descendant of B.

In degree out degree

- The total no. of edges coming to the node
- The total no of edges going out from the node

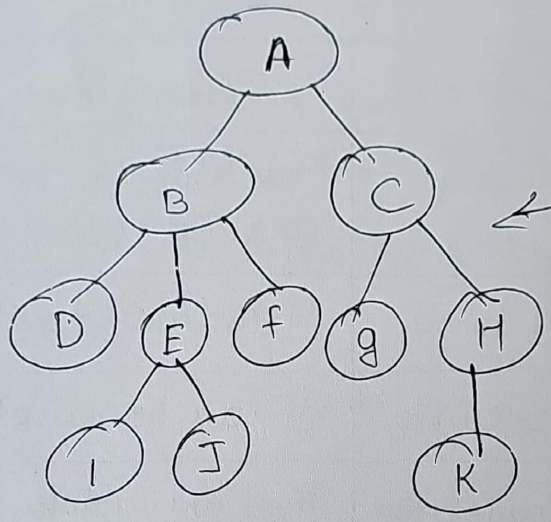
~~5.2~~ Tree types and traversal methods, Types of Trees: General tree, Binary tree, Binary Search tree (BST) Binary tree traversal: In order, Pre-order & post order.

Types of Tree



1) General tree:

- It is a type of tree, where each node can have any no of children
- There is no limitation on children.
- It cannot be empty, It is in unordered way.



← General tree

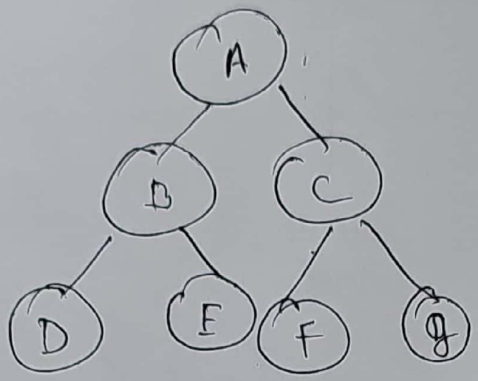
- Here A nodes have 2 child node, B has 3 child node, C has 2 child node, E has 2 node, & K has 1 child node.

2) Binary tree:

- It is a type of tree in which each node can have at most two nodes.
- There is a limitation on the degree of a node, because the node in the binary tree can't have more than 2 child node
- The topmost node of a binary tree is called root node & there are two subtree

Subtree → left subtree

- Unlike general tree, binary tree also cannot be empty.
- The binary tree is ordered
-



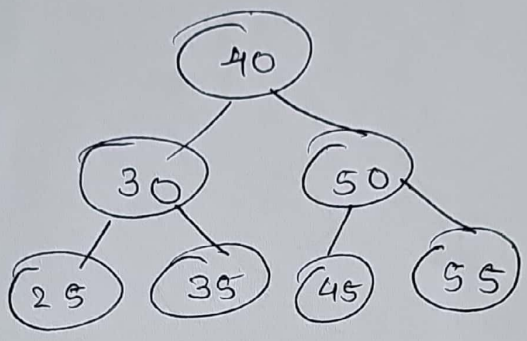
- Here all the nodes (parent nodes) have only 2 child nodes in this binary tree.

Difference between General & Binary tree.

General tree	Binary tree
• Each node can have many child-nodes.	– each node can have only 2- child-nodes.
• It is in unordered form	– It is in ordered form
• General tree cannot be empty	– It can be empty.
• In General tree, there is either Zero Subtree or many Sub-tree	– There are mainly 2 Subtree Subtree → left Subtree → Right Subtree
• There is no limit to the degree of the node	– There are limitation to the degree of the node.
• Here, general tree has an in-degree of one & maximum out-degree of n	– Binary tree has in-degree of one & out degree of 2

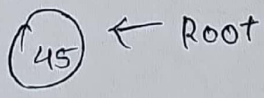
Binary Search tree:

- In binary Search tree, the value of left node must be smaller than root node or Parent node.
- the value of Right node must be greater than root node or parent node.

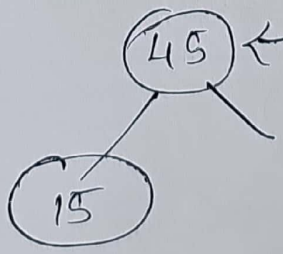


- Let's us create a binary Search tree
  eg- 45, 15, 79, 90, 10, 55, 12, 20, 50
  1st we will Insert 45 to the tree so it will be a root node

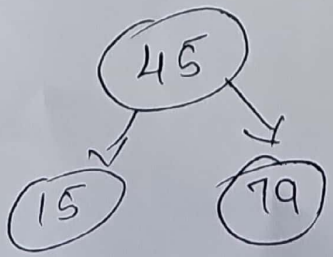
STEP 1: Insert 45 -



STEP 2: Insert 15 -
It is smaller than 45 so it will go to left.

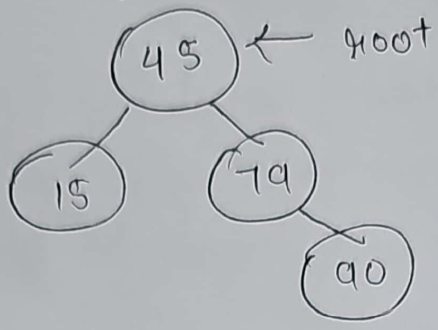


STEP 3: Insert 79
It is larger than 45 so it will go to right



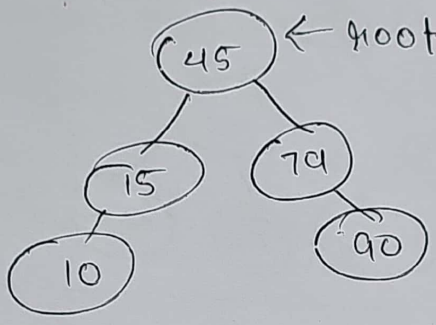
Step4: Insert 90

It is greater than 45 & 75, so it will go to right side



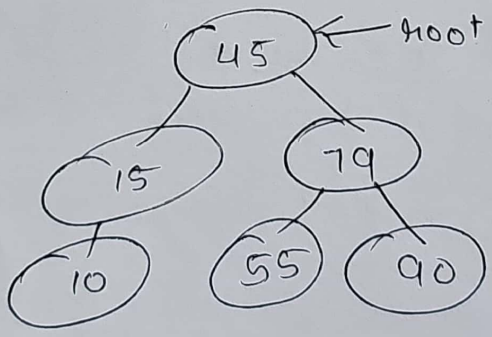
STEP5 : Insert 10

It is smaller than 45 & 15, so it will goto left subtree.



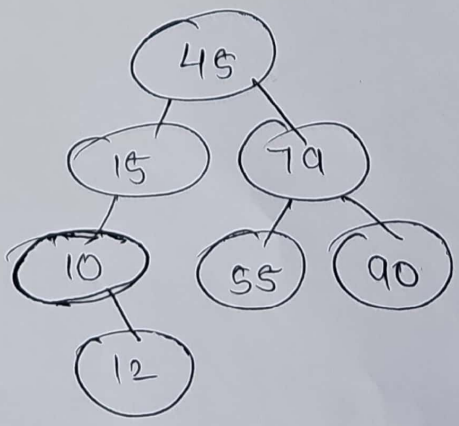
STEP6: Insert 55

It is greater than 45, but smaller than 79 so it will go to left of 75



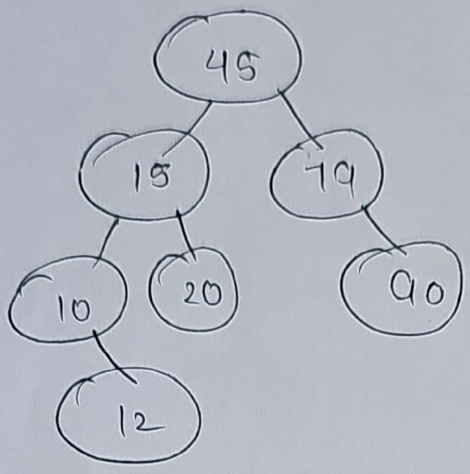
STEP7 : Insert 12

It is smaller than 45, but greater than 10 so it will go to Right of 12



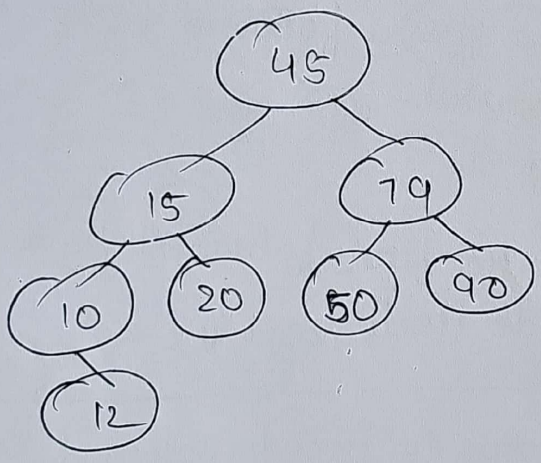
Step 6: Insert 20

20 is smaller than 45, but greater than 15 so it will come at Right of 15



Step 7: Insert 50

50 is greater than 45, smaller than 79 so it will be at left of 79.



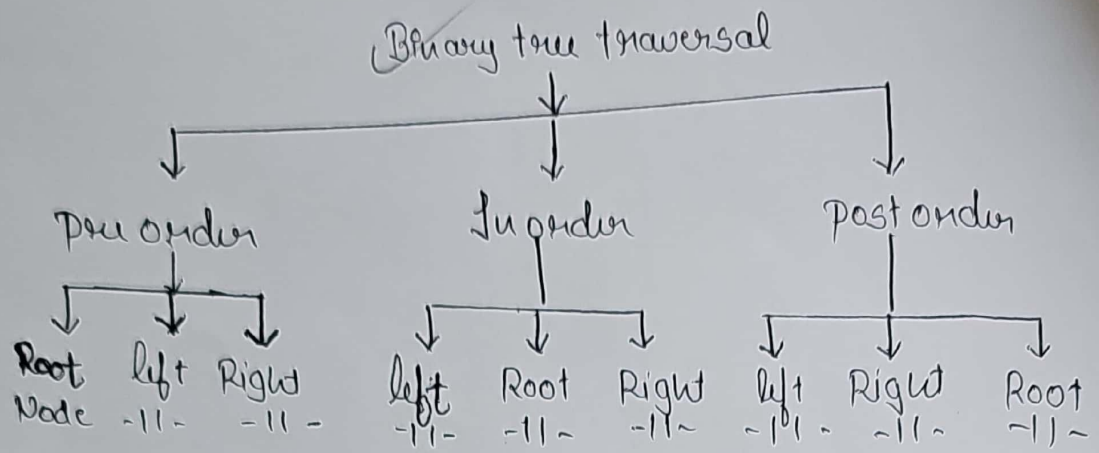
final answer

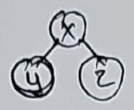
winter 2019

Q1 Draw a Binary Search tree for given no (4M)

50, 33, 44, 22, 77, 35, 60, 40.

Binary tree traversal



eg = 

1) Pre-order traversal

→ the node itself is visited first, followed by the left subtree & the Right Subtree. eg = x y z

2) In-order traversal

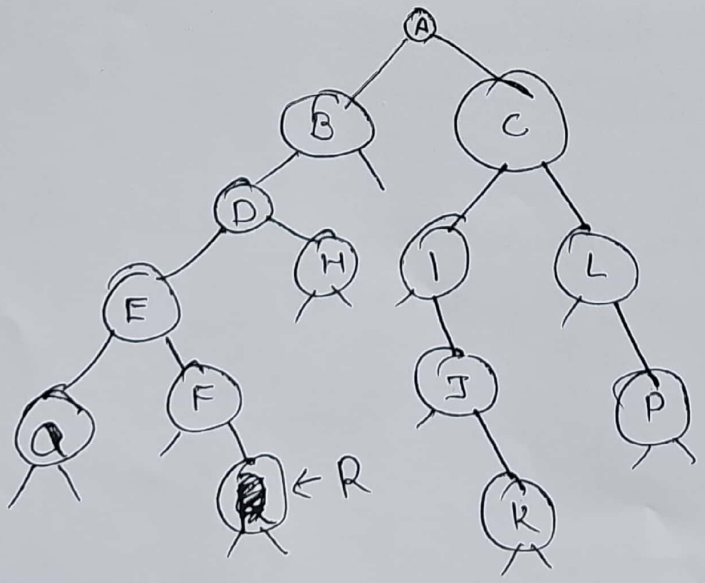
→ the left subtree is visited first, followed by the node itself and finally the Right Subtree. eg = y x z

3) Post-order traversal

→ The left subtree is visited first, followed by the Right subtree & finally the node itself. eg = y z x

---

for eg- Sort the binary tree in in-order, pre-order & post-order



Pre-order = A, B, D, E, Q, F, R, H, C, I, J, K, L, P

In-order = Q, E, F, R, D, H, B, A, I, J, K, C, L, P

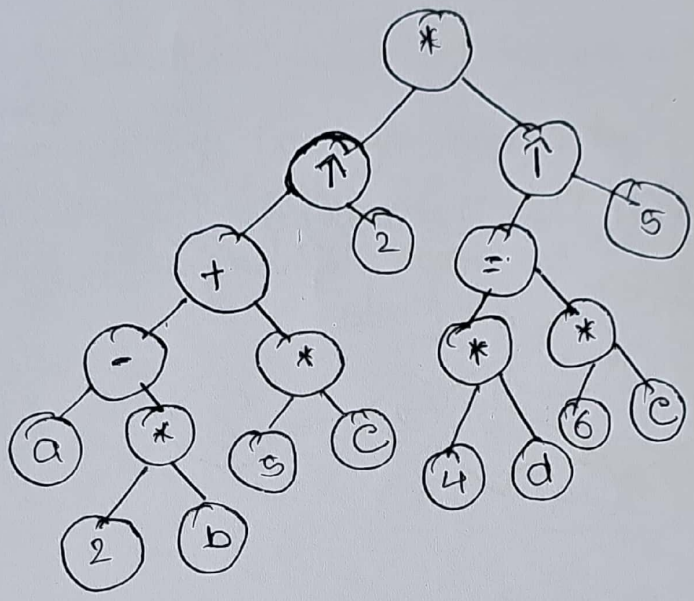
Post-order = Q, R, F, E, H, D, B, K, J, I, P, L, C, A

5.3 Expression tree:

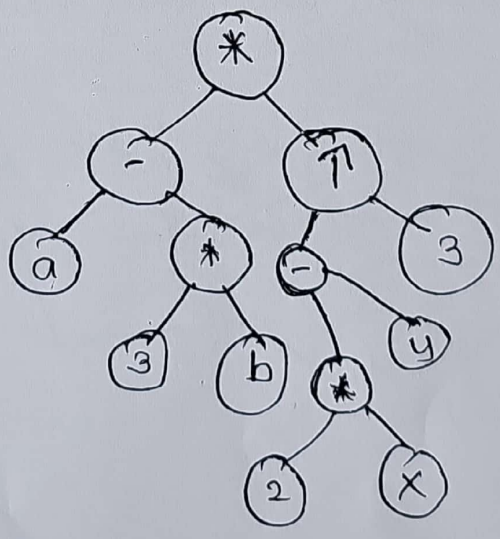
~~Summer~~ winter 2019

Draw the structure of the following expression (4M)

1) $(a-2b+3e)^2 * (4d-6c)^5$



2) $(a-3b) * (2x-y)^3$



5.4 Introduction to Graph terminologies

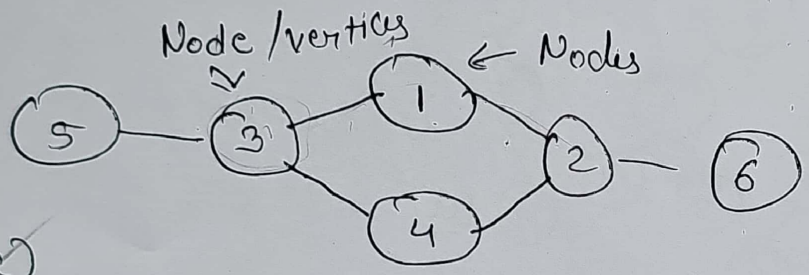
graph, node (vertices), arc (edges), directed graph, un-directed graph, in-degree, out degree, adjacent, successor, predecessor

Graph: A graph is a non-linear data structure consisting of vertices and edges.

- The vertices are also suffixed as nodes & edges are suffixed as arcs that connect two nodes.

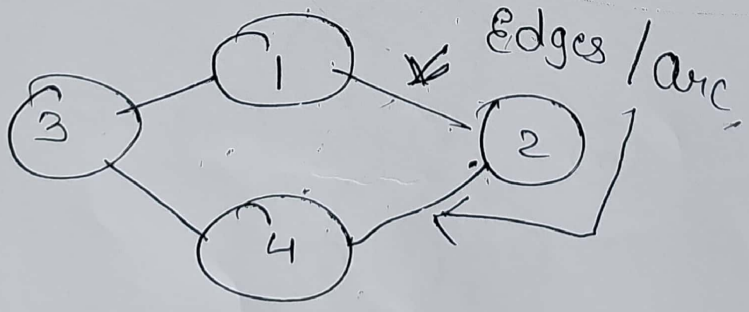
node (vertices): Vertices are the fundamental units of a graph.

- Vertices is nothing but a node of that graph.
- Vertices can be labelled or unlabelled.



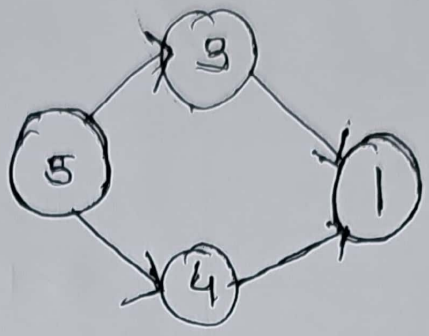
edge (arc):

- Edges are used to connect two nodes of the graph
- Edge can connect any two nodes in any possible way.
- Edges are also known as arc. They can be labelled or unlabelled.



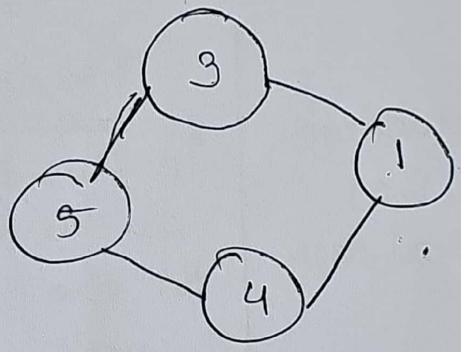
ucted graph — Imp (2M) everytime

- A graph in which edges has direction. That is the nodes are ordered pairs in the definition of every edge.



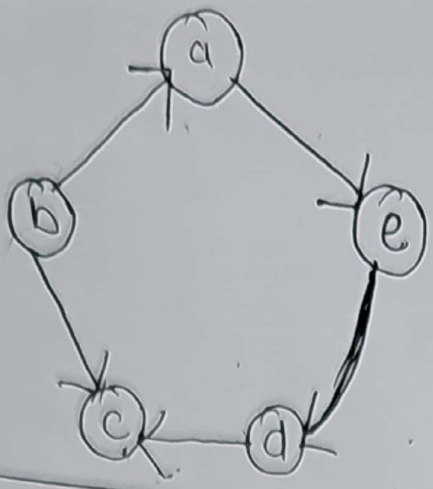
Undirected graph

- A graph in which edges do not have any direction. That is the node are unordered pairs in the definition of every edges



Indegree : Indegree of a vertex is the number of edges which are coming into the vertex

Outdegree : Outdegree of a vertex is the number of edges which are going out from the vertex.

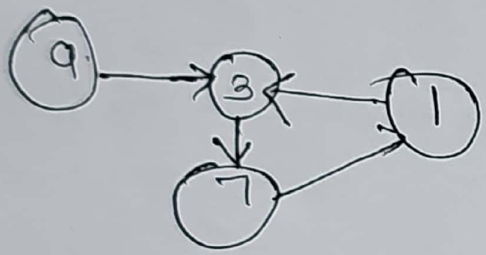


vertex	Indegree	Out degree
a	1	1
b	0	2
c	2	0
d	1	1
e	1	1

Sucessor:

A is a directed graph in which each vertex has outdegree one, i.e, each one edge starts at each node

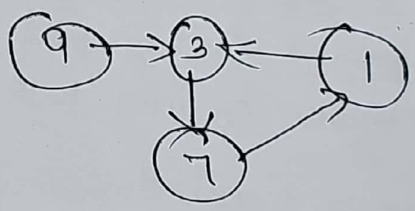
eg-



node	Sucessor
1	3
3	7
7	1
9	3

Predecessor

In this each vertex has an indegree one, i.e the ~~one~~ parent of one child node



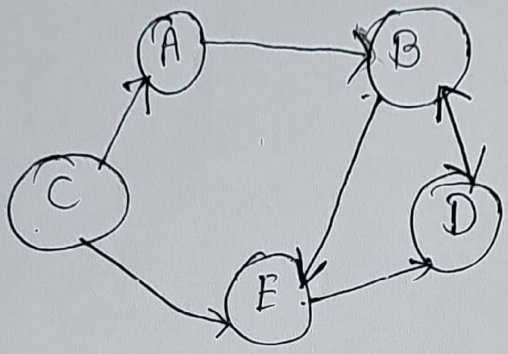
node Predecessor

5 Adjacency list, Adjacency Matrix of directed / undirected graph

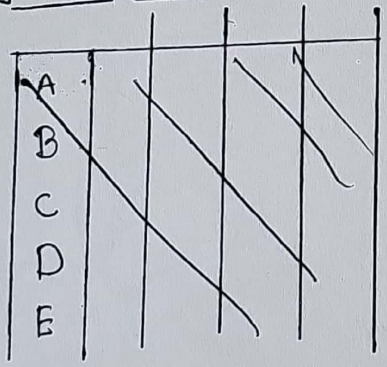
1) Adjacency Matrix:

- Adjacency Matrix is a 2-D Array of size V × V, where V is no of vertices in a graph.

eg- Directed Matrix



Adjacent list:



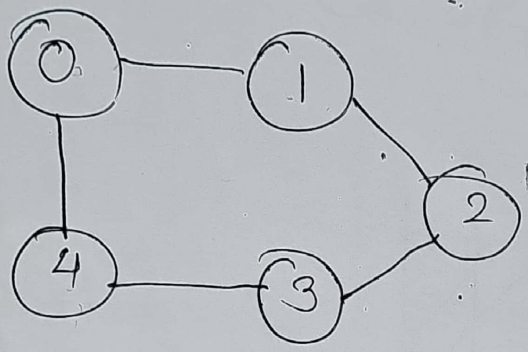
	A	B	C	D	E
A	0	1	0	0	0
B	0	0	0	1	1
C	1	0	0	0	1
D	0	1	0	0	0
E	0	0	0	1	0

OR

Nodes	Adjacent Nodes
A	B
B	D, E
C	A, E
D	B
E	D

2) Adjacent list :

- In the adjacency list representation, a graph is represented as an array of linked list. The index of the array represents a vertex and each element in it's linked list represents the vertices from an edge with the vertex



Ans =