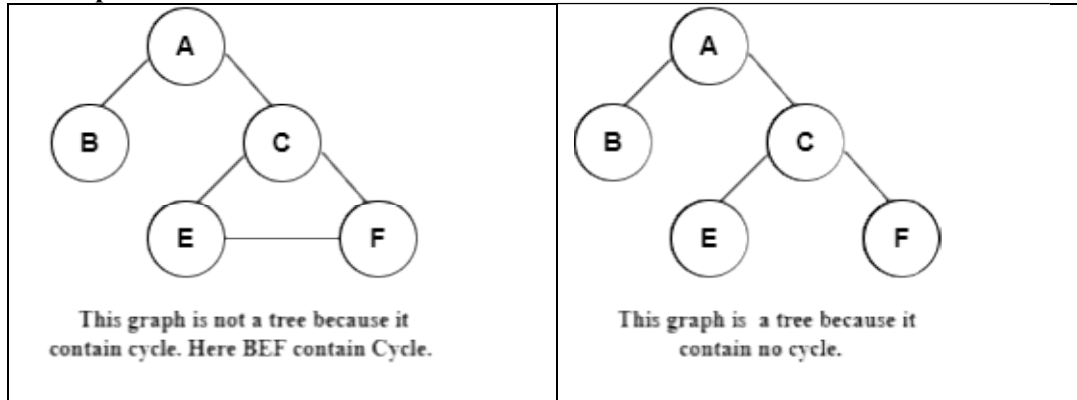


Tree Data Structure

Tree is a non-linear data structure which organizes data in a hierarchical structure and this is a recursive definition. Or a tree is a connected graph without any circuits. Or, if in a graph, there is one and only one path between every pair of vertices, and then graph is called as a tree.

Example:

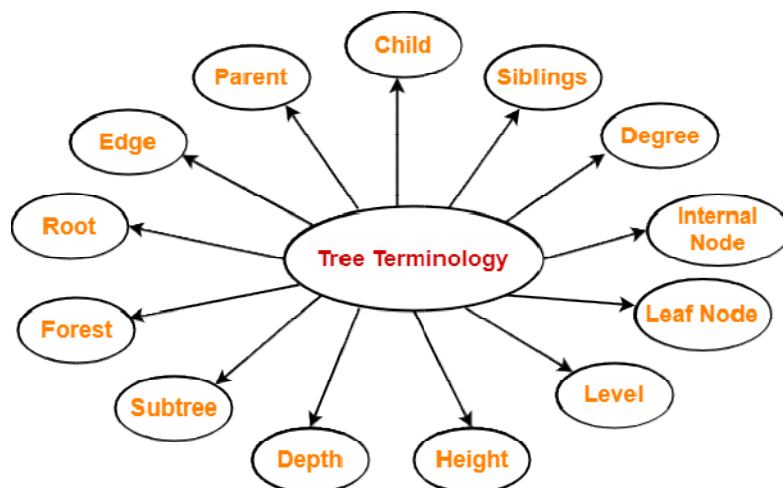


The important properties of tree data structure are-

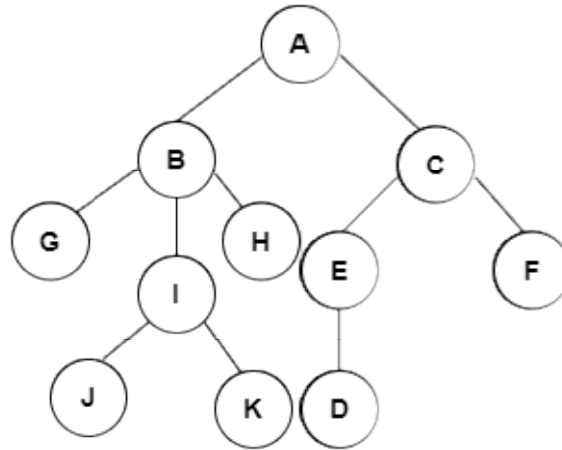
- ✓ There is one and only one path between every pair of vertices in a tree.
- ✓ A tree with n vertices has exactly $(n-1)$ edges.
- ✓ A graph is a tree if and only if it is minimally connected.
- ✓ Any connected graph with n vertices and $(n-1)$ edges is a tree.
- ✓ Tree must not contain cycle.
- ✓ A unique path traverses from root to any node of tree

Tree Terminology-

The important terms related to tree data structure are-



See the following tree and explain all the terminologies:



This is a simple tree. Explain all the terminologies using this tree.

1. Root

- ✓ The first node from where the tree originates is called as a **root node**.
- ✓ In any tree, there must be only one root node. Here A is only the root node.

2. Edge

- ✓ The connecting link between any two nodes is called as an **edge**.
- ✓ In a tree with n number of nodes, there are exactly (n-1) number of edges.

3. Parent

- ✓ The node which has a branch from it to any other node is called as a **parent node**.
- ✓ In other words, the node which has one or more children is called as a parent node.
- ✓ In a tree, a parent node can have any number of child nodes.

Here,

Node A is the parent of nodes B and C

Node B is the parent of nodes G, I and H

Node E is the parent of node D

4. Child

- ✓ The node which is a descendant of some node is called as a **child node**.
- ✓ All the nodes except root node are child nodes.

Here

Node B and C are the child of node A

Node J and K are the child of node I

Node D child of node E

5. Siblings

- ✓ Nodes which belong to the same parent are called as **siblings**.
- ✓ In other words, nodes with the same parent are sibling nodes.

Here

Node B and C are siblings.

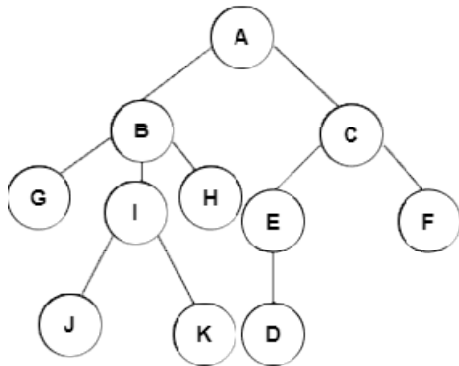
Node J,K are siblings.

6. Degree

There are two types of degree.

- ✓ **Degree of a node** is the total number of children of that node.
- ✓ **Degree of a tree** is the highest degree of a node among all the nodes in the tree.

See the following tree:



Here,

- Degree of node A = 2
- Degree of node B = 3
- Degree of node C = 2
- Degree of node D = 0

So, the highest degree of a node among all the nodes of this tree is $\text{Degree}(B)=3$. So, the degree of this tree is 3.

7. Internal Node

- ✓ The node which has at least one child is called as an **internal node**.
- ✓ Internal nodes are also called as **non-terminal nodes**.
- ✓ Every non-leaf node is an internal node.

Here, A,B,C,I,E is internal node.

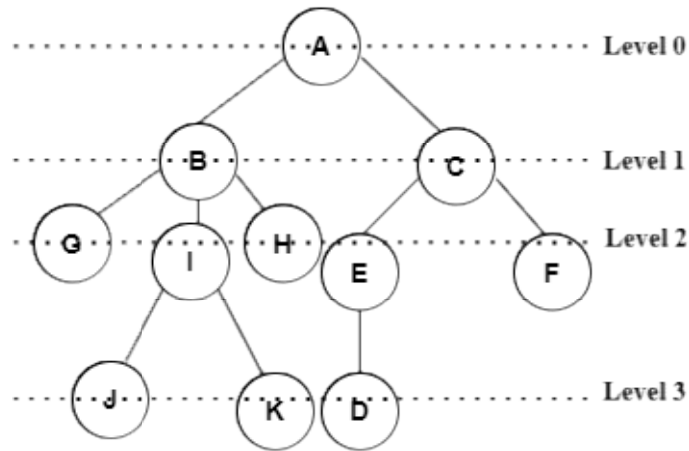
8. Leaf Node

- ✓ The node which does not have any child is called as a **leaf node**.
- ✓ Leaf nodes are also called as **external nodes** or **terminal nodes**.

Here, G,J,K,H,D and F are leaf node.

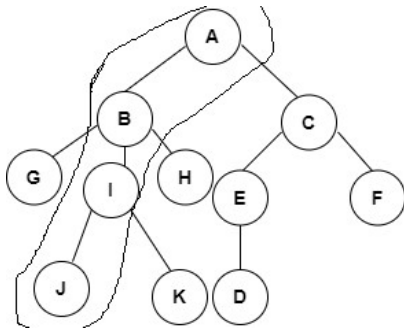
9. Level

- ✓ In a tree, each step from top to bottom is called as **level of a tree**.
- ✓ The level count starts with 0 and increments by 1 at each level or step.



10. Height

- ✓ Total number of edges that lies on the longest path from any leaf node to a particular node is called as **height of that node**.
- ✓ **Height of a tree** is the height of root node.
- ✓ Height of all leaf nodes = 0



- Height of node A = 3
- Height of node B = 2
- Height of J,K,D = 0

11. Depth

- ✓ Total number of edges from root node to a particular node is called as **depth of that node**.
- ✓ **Depth of a tree** is the total number of edges from root node to a leaf node in the longest path.
- ✓ Depth of the root node = 0
- ✓ The terms “level” and “depth” are used interchangeably.
 - Depth of node A = 0
 - Depth of node B = 1
 - Depth of node C = 1
 - Depth of node J = 3

12. Forest: A collection of disjoint trees.

Types of Trees

Types of trees depend on the number of children a node has. There are two major tree types:

i. **General Tree:** A tree in which there is no restriction on the number of children a node has, is called a General tree. Examples are Family tree, Folder Structure.

ii. **Binary Tree:** In a Binary tree, every node can have at most 2 children, left and right. In diagram below, B & D are left children and C, E & F are right children.

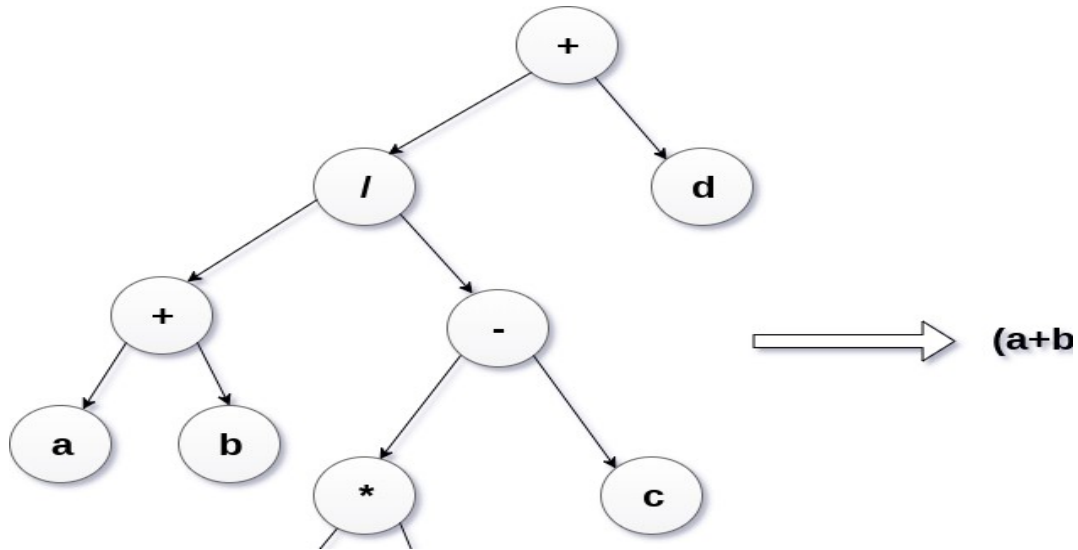
Binary trees are further divided into many types based on its application.

- ✓ Full Binary Tree: If every node in a tree has either 0 or 2 children, then the tree is called a full tree. The tree in the above diagram is **not a full** binary tree as node C has only the right child.
- ✓ Perfect Binary tree: It is a binary tree in which all interior nodes have two children and all leaves have the same depth or same level.
- ✓ Balanced Tree: If the height of the left and right subtree at any node differs at most by 1, then the tree is called a balanced tree.
- ✓ Binary Search Tree: It is a binary tree with binary search property. Binary search property states that the value or key of the left node is less than its parent and value or key of right node is greater than its parent. And this is true for all nodes

Binary search trees are used in various searching and sorting algorithms. There are many variants of binary search trees like AVL tree, B-Tree, Red-black tree, etc.

➤ **Construct an expression tree by using the following algebraic expression.**

$$(a + b) / (a * b - c) + d$$



➤ **BINARY TREE**

It is a special type of tree where each node of tree contains either 0 or 1 or 2 children.

Or

Binary Tree is either empty, or it consists of a root with two binary trees called left-sub tree and right sub-tree of root (left or right or both the sub trees may be empty)

➤ **Properties of binary tree**

- ✓ Binary tree partitioned into three parts.
- ✓ First subset contains root of tree.
- ✓ Second subset is called left subtree.
- ✓ Another subset is called right subtree.
- ✓ Each subtree is a binary tree.
- ✓ Degree of any node is 0/1/2.
- ✓ The maximum number of nodes in a tree with height 'h' is $2^{h+1} - 1$
- ✓ The maximum number of nodes at level 'k' is $2^k - 1$.
- ✓ For any non-empty binary tree, the number of terminal nodes
- ✓ with n_2 , nodes of degree 2 is $N_0 = n_2 + 1$
- ✓ The maximum number of nodes in a tree with depth d is $2^d - 1$.

29. **Maximum how many nodes can be placed in a Binary Tree of N levels?** [Com(AP)-2020]

- a) 2^N b) 2^{N-1} c) $2^{N-1} - 1$ d) N^2 **Ans.: c**

30. **Which algorithm will be most efficient to find out the shortest path between two given nodes in an undirected weighted graph?** [Com(AP)-2020]

- a) Breadth First Search b) Depth First Search
c) Dijkstra's algorithm d) Floyd-Warshall algorithm **Ans.: b**

➤ **Some Graph Algorithm:**

- ✓ **DFS:** The most useful graph algorithms are search algorithms. DFS (Depth First Search) is one of them
- ✓ **BFS:** BFS is another search algorithm (Breadth First Search). It is usually used to calculate the distances from a vertex v to all other vertices in unweighted graphs.
- ✓ **Shortest path:** Shortest path algorithms are algorithms to find some shortest paths in directed or undirected graphs.
- ✓ **Dijkstra:** This algorithm is a single source shortest path (from one source to any other vertices). Pay attention that you can't have edges with negative weight.
- ✓ **Floyd-Warshall:** Floyd-Warshall algorithm is an all-pairs shortest path algorithm using dynamic programming.
- ✓ **Bellman-Ford:** Bellman-Ford is an algorithm for single source shortest path where edges can be negative (but if there is a cycle with negative weight, then this problem will be NP).
- ✓ **SPFA:** SPFA (Shortest Path Faster Algorithm) is a fast and simple algorithm (single source) that its complexity is not calculated yet.

➤ **MST**

- ✓ MST = Minimum Spanning Tree :) (if you don't know what it is, google it).
- ✓ **Best MST algorithms:**
- ✓ **Kruskal:** Kruskal's algorithm is a greedy algorithm, which helps us find the minimum spanning tree for a connected weighted graph, adding increasing cost arcs at each step.
- ✓ **Prim:** Prim's algorithm is a greedy algorithm, which helps us find the minimum spanning tree for a weighted undirected graph.

PREVIOUS YEAR QUESTIONS

31. Which data structure allow deleting data elements front and inserting at rear? *[Assistant programmer (ICB)- 2017]*
 a) Stack b) Queue c) Dequeue d) Binary search tree **Ans.: b**
32. The term push and pop related to *[Combined (AME)-2018]*
 a) Array b) list c) stack d) all of this **Ans: c**
33. Which of the following name does not related to stacks? *[AP(ICB)- 2017, AP(HBFC, KB)-2018]*
 a) FIFO List b) LIFO List c) Piles d) Push Down List **Ans.: a**
- ✍ *Explanation:* FIFO is related to Queue.
34. The term push and pop related to *[Combined (AME)-2018]*
 a) Array b) list c) stack d) all of this **Ans.: c**
35. Which of these data types is used by operating system to manage the Recursion in Java? *[Combined(IT/ICT-2018)]*
 a) Array b) Stack c) Queue d) Tree **Ans.: b**
36. Pushing an element into stack already having five elements and stack size of 5. Result in *[Combined (AP)-2018]*
 a) Overflow b) Crash c) Underflow d) User flow **Ans.: a**
37. The data structure required to check whether an expression contains balanced parenthesis is *[Combined (AP)-2018]*
 Or, The data structure required to evaluate a postfix expression is? *[AP (ICB)- 2017]*
 a) Stack b) Queue c) Array d) Tree **Ans.: a**

38. Find the correct arranged data after stack operation? [BB (AP)-2016]

push(1), push(2), pop, push(1), push(2), pop, pop, pop, push(2)

- a) 2 2 1 1 2 b) 2 2 1 2 1 c) 2 2 2 2 1 d) 2 2 2 1 2

Operation	Stack	Pop Sequence
Push 1	1	
Push 2	1, 2	
Pop	1	2
Push 1	1, 1	
Push 2	1, 1, 2	
Pop	1, 1	2, 2
Pop	1	2, 2, 1
Pop	Empty	2, 2, 1, 1
Push 2	2	
Pop	Empty	2, 2, 1, 1, 2

Ans.: a

39. Stack operations are [AP (SBL)-2016]

- a) delete, insertion b) insertion, delete c) push, pop d) pop, raer

Ans.: c