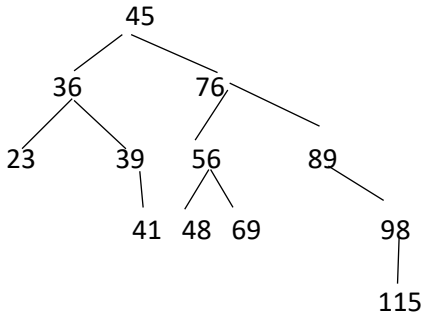
Data structure and algorithm
RCS 121
Assignment 4, deadline of submission is on Tuesday 21/ 6/2022

Qn1 and qn4 missing

2; What is a Binary Search Tree (BST)? Make a BST for the following sequence of numbers. 45, 36, 76, 23, 89, 115, 98, 39, 41, 56, 69, 48 Traverse the tree in Preorder In order and post order.

A binary search tree B is a binary tree whose each node satisfies the following *conditions:*
The value obtained of the left-subtree of 'x' is less than the value at 'x'
The value obtained of the right-subtree of 'x' is greater than value at 'x'
The left-subtree and right-subtree of the binary search tree are again binary search tree



3: With example explain the function Recursion

Recursion in C

Recursion is the process, which comes into existence when a function calls a copy of itself to work on a smaller problem.

Any function, which calls itself, is called recursive function, and such function calls are called recursive calls.

Recursion involves several numbers of recursive calls. However, it is important to impose a termination condition of recursion.

Example

```
#include <stdio.h>
int fact (int);
int main()
{
   int n,f;
   printf("Enter the number whose factorial you want to calculate?");
   scanf("%d",&n);
   f = fact(n);
   printf("factorial = %d",f);
}
int fact(int n)
{
```

```
if (n==0)
{
   return 0;
}
else if ( n == 1)
{
   return 1;
}
else
{
   return n*fact(n-1);
} }
```

4: Given a sequence of number below: 50,60,40,70,45,55,30,80,65,35,25,75,85 when creating a binary search tree, what is the height of the tree?

5: Discuss the real world applications of tree and graph data structure.

Applications of Graph Theory

Graph Theory is used in vast area of science and technologies. Some of them are given below:

- Computer Science
- Electrical Engineering
- Computer Network
- Social science
- mathematics

Applications of trees

The following are the applications of trees:

- **Storing naturally hierarchical data:** Trees are used to store the data in the hierarchical structure. For example, the file system. The file system stored on the disc drive, the file and folder are in the form of the naturally hierarchical data and stored in the form of trees.
- **Organize data:** It is used to organize data for efficient insertion, deletion and searching. For example, a binary tree has a logN time for searching an element.
- **Trie:** It is a special kind of tree that is used to store the dictionary. It is a fast and efficient way for dynamic spell checking.
- **Heap:** It is also a tree data structure implemented using arrays. It is used to implement priority queues.
- **B-Tree and B+ Tree:** B-Tree and B+ Tree are the tree data structures used to implement indexing in databases.
- **Routing table:** The tree data structure is also used to store the data in routing tables in the routers.

6: With example discuss Terminologies in Tree Data Structure

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 Terminology-

The important terms related to tree data structure are

- **Root**
- **Edge**
- **Parent**
- **Children**
- **Sibling**
- **Degree**
- **Internal node**
- **Leaf node**
- **Level**
- **Height**
- **Depth**
- **Subtree**
- **Forest**
- **path**

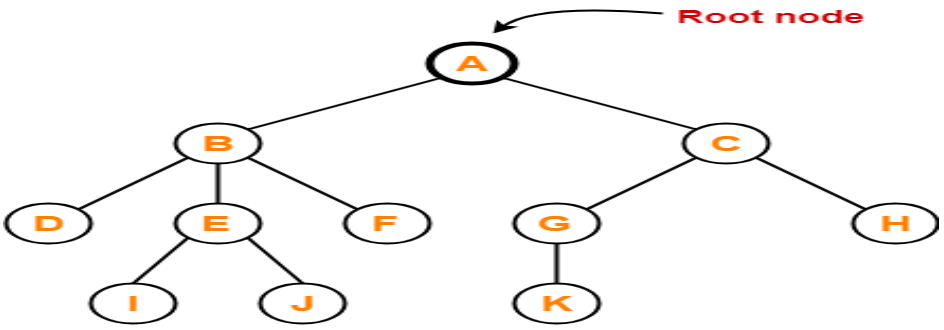
1. Root

The first node from where the tree originates is called as **a root node**.

In any tree, there must be only one root node.

We can never have multiple root nodes in a tree data structure.

example



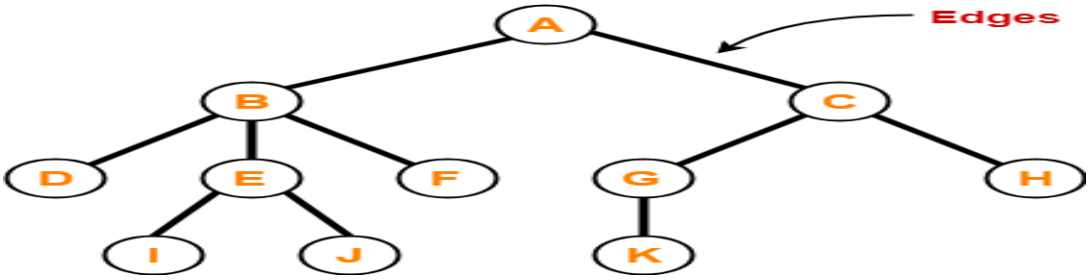
Here, node A is the only root node.

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.

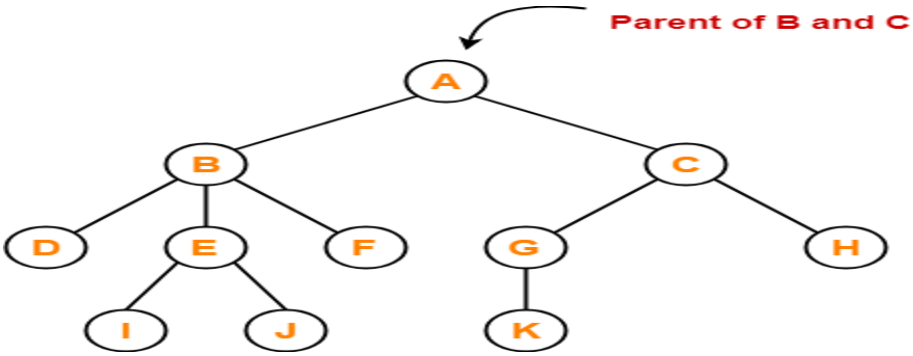
Example



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.

Example

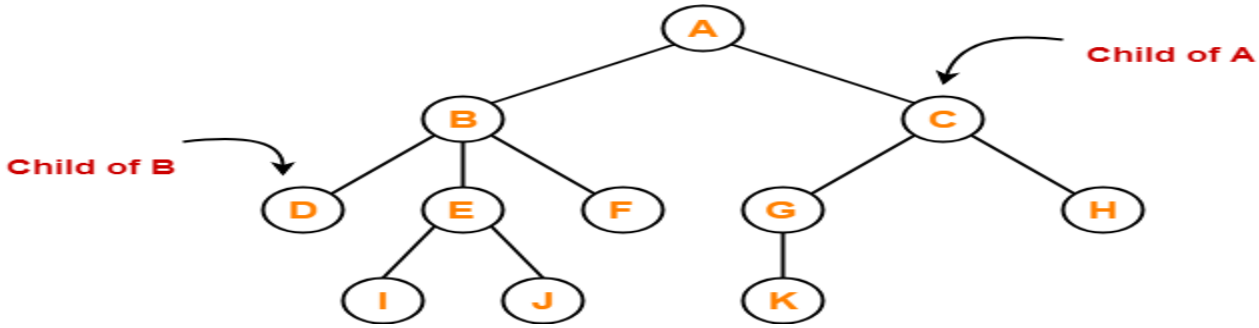


Here,
Node A is the parent of nodes B and C
Node B is the parent of nodes D, E and F
Node C is the parent of nodes G and H
Node E is the parent of nodes I and J
Node G is the parent of node K

4. Child

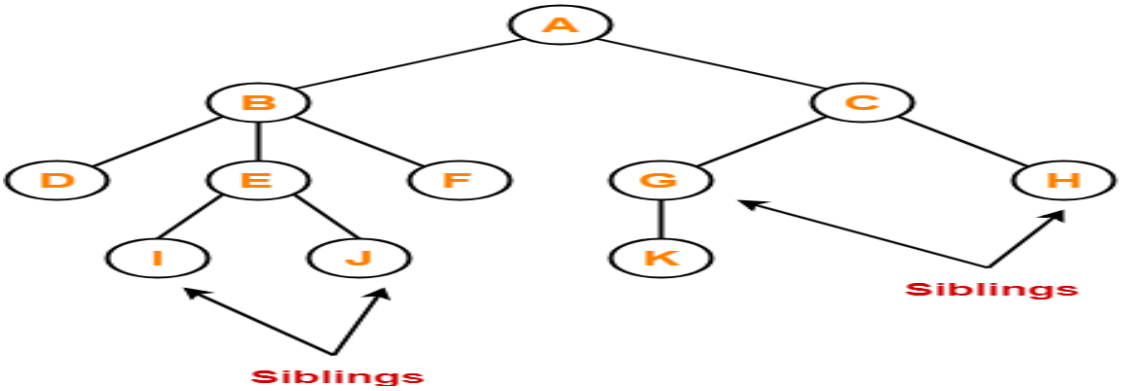
The node, which is a descendant of some node, is called as a child node. All the nodes except root node are child nodes.

Example



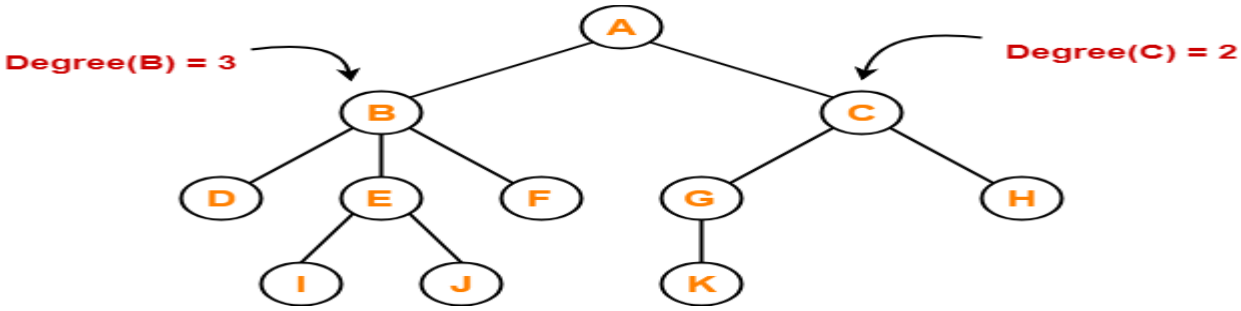
Here,
Nodes B and C are the children of node A
Nodes D, E and F are the children of node B
Nodes G and H are the children of node C
Nodes J and I are the children of node E
Node K is the child of node G

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



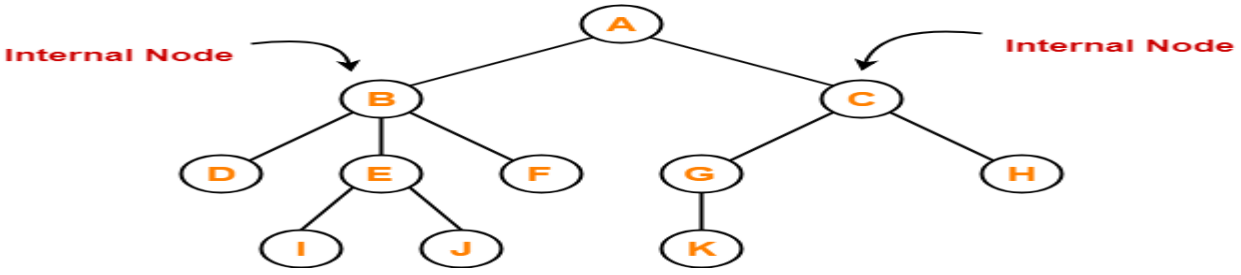
Here,
Nodes B and C are siblings
Nodes D, E and F are siblings
Nodes G and H are siblings
Nodes I and J are siblings

6. 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.
Example



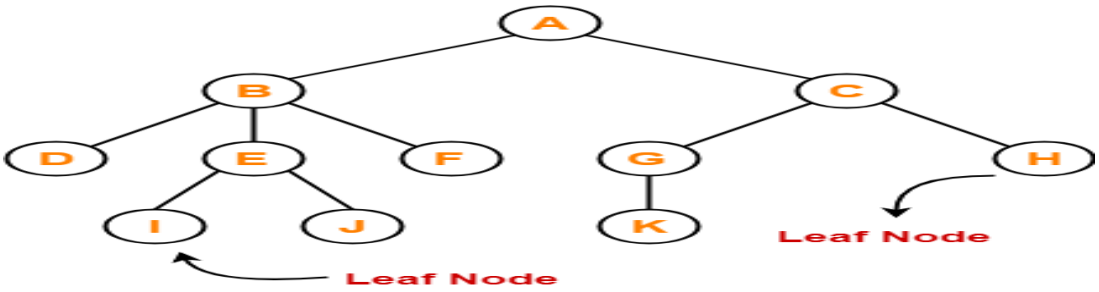
Here,
Degree of node A = 2
Degree of node B = 3
Degree of node C = 2
Degree of node D = 0
Degree of node E = 2
Degree of node F = 0
Degree of node G = 1
Degree of node H = 0
Degree of node I = 0
Degree of node J = 0
Degree of node K = 0

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.
Example



Here, nodes A, B, C, E and G are internal nodes.

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.
Example



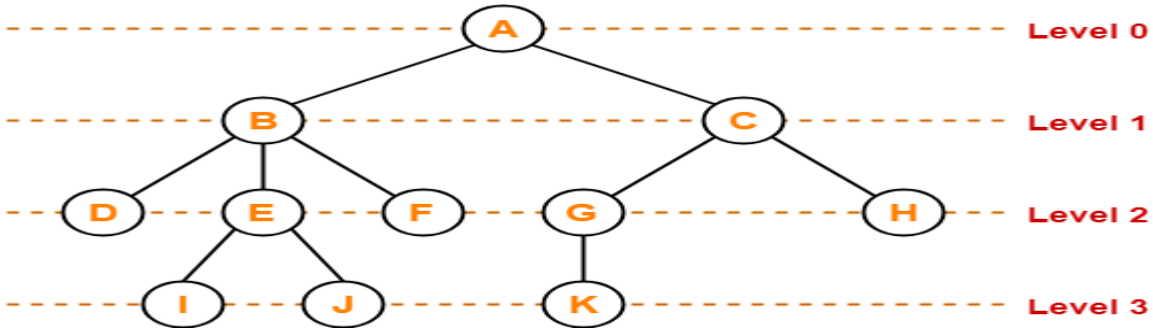
Here, nodes D, I, J, F, K and H are leaf nodes.

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.

Example



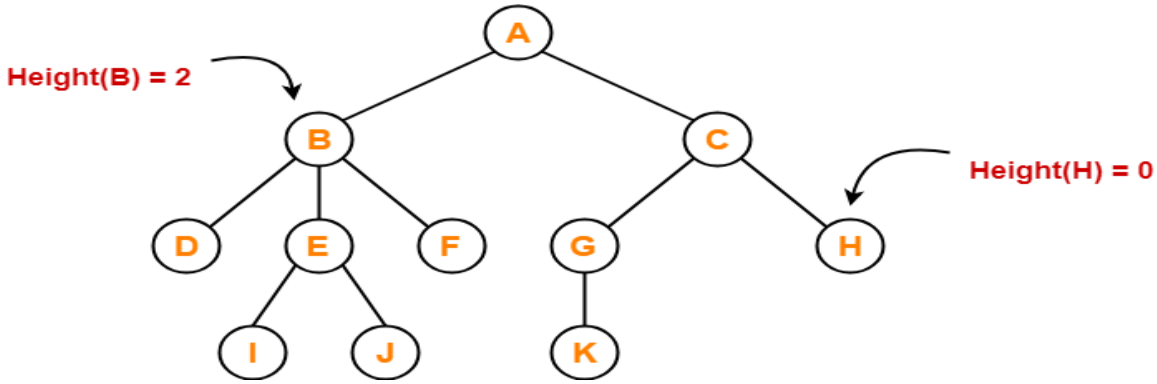
10. Height

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

Example



Here,

Height of node A = 3

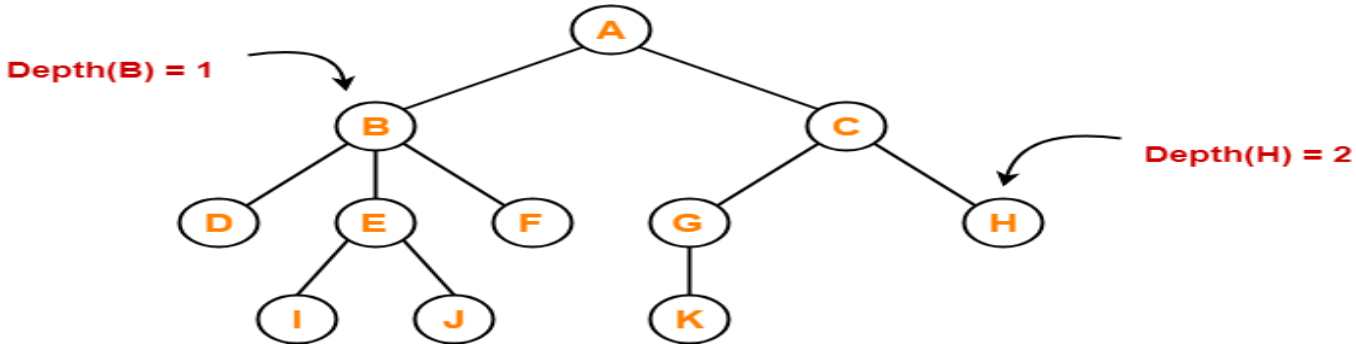
Height of node B = 2

Height of node C = 2
Height of node D = 0
Height of node E = 1
Height of node F = 0
Height of node G = 1
Height of node H = 0
Height of node I = 0
Height of node J = 0
Height of node K = 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.

Example

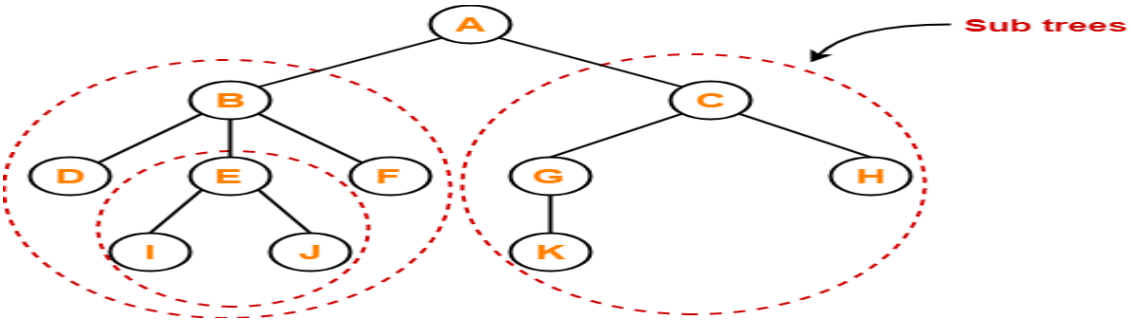


Here,
Depth of node A = 0
Depth of node B = 1
Depth of node C = 1
Depth of node D = 2
Depth of node E = 2
Depth of node F = 2
Depth of node G = 2
Depth of node H = 2
Depth of node I = 3
Depth of node J = 3
Depth of node K = 3

12. Subtree

In a tree, each child from a node forms a subtree recursively.
Every child node forms a subtree on its parent node.

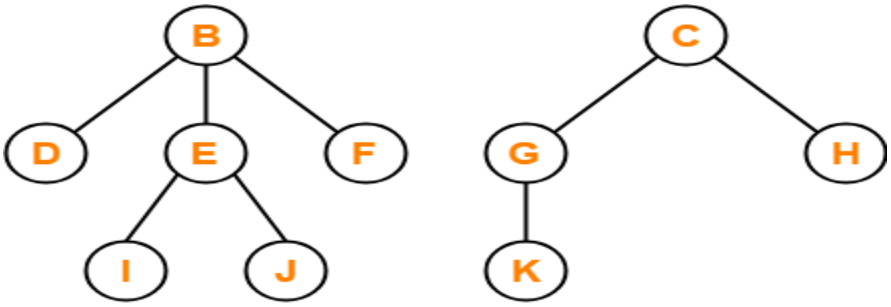
Example



13. Forest

A forest is a set of disjoint trees.

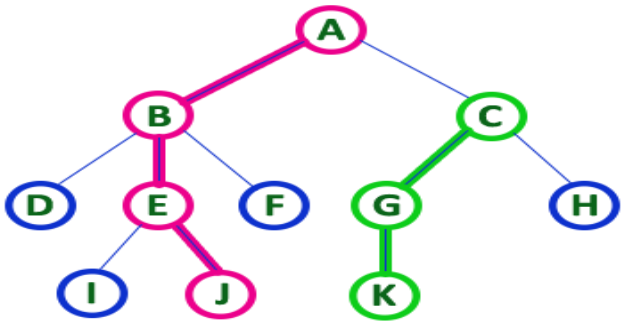
Example



Forest

14. Path

In a tree data structure, the sequence of Nodes and Edges from one node to another node is called as PATH between that two Nodes. Length of a Path is total number of nodes in that path. In below example the path A - B - E - J has length 4.



- In any tree, 'Path' is a sequence of nodes and edges between two nodes.

Here, 'Path' between A & J is

A - B - E - J

Here, 'Path' between C & K is

C - G - K

15. Sub Tree

In a tree data structure, each child from a node forms a subtree recursively. Every child node will form a subtree on its parent node.