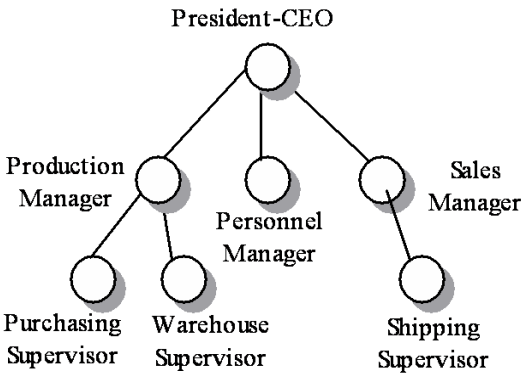
INTRODUCTION TO TREES

A Tree T is a non-linear data structure consisting of a finite set of one or more nodes such that

- **There is one specially designated node called ROOT NODE and**
- **And the remaining nodes are partitioned into a collection of**

  Disjoint sub-trees of the root each of which is also a tree.



HIERARCHICAL TREE STRUCTURE

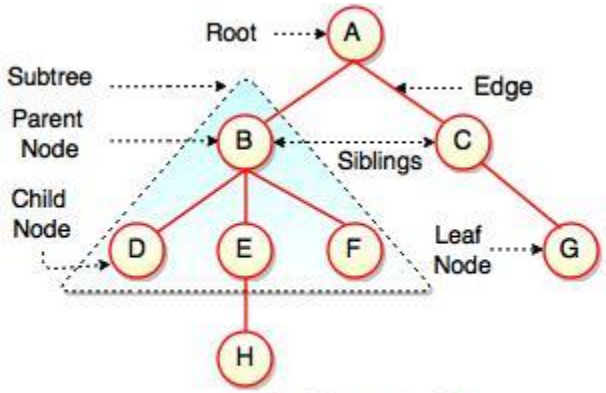
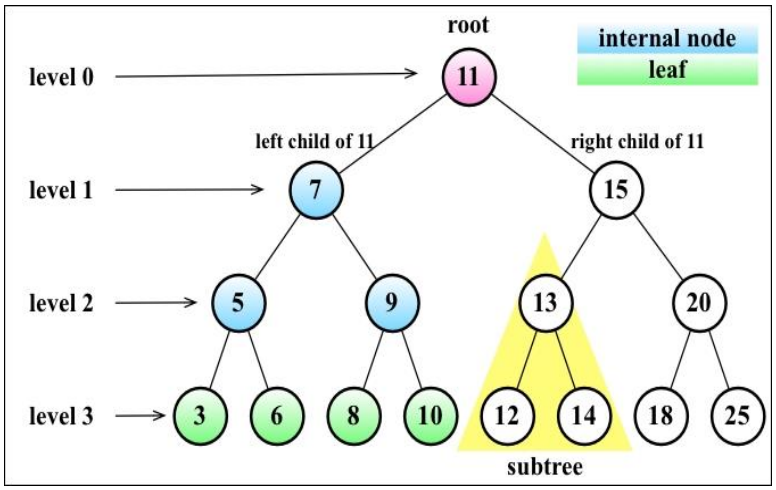


Fig. Structure of Tree

A tree T **can also be defined as a non-linear data structure consisting of a collection of data items called as nodes where the nodes are arranged in hierarchical fashion recursively.**



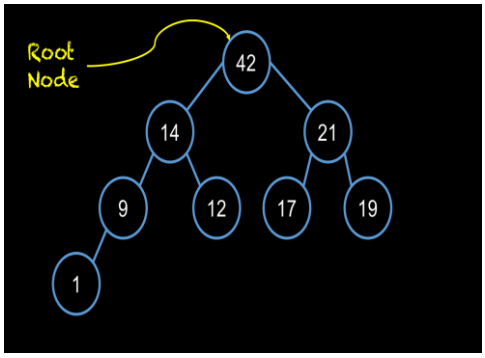
The important properties of tree data structure are-

- **A tree T has only one root node which is the first node of the tree.**
- **A pointer or reference variable called as 'ROOT', which always refers to root (1st node) node of the tree by which we access any node of the tree.**
- **A tree T consisting of only a single node is also a tree.**
- **If *'ROOT'* pointer refers to 'null' value, then the tree is called null tree or empty tree.**
- **There is one and only one path or edge between every pair of nodes in a tree.**
- **A tree with n nodes has exactly (n-1) edges.**

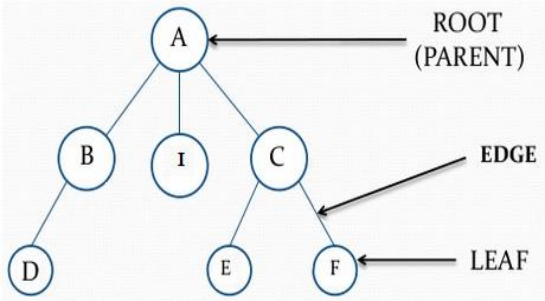
TREE TERMINOLOGY

In a tree data structure we use the following terminology:

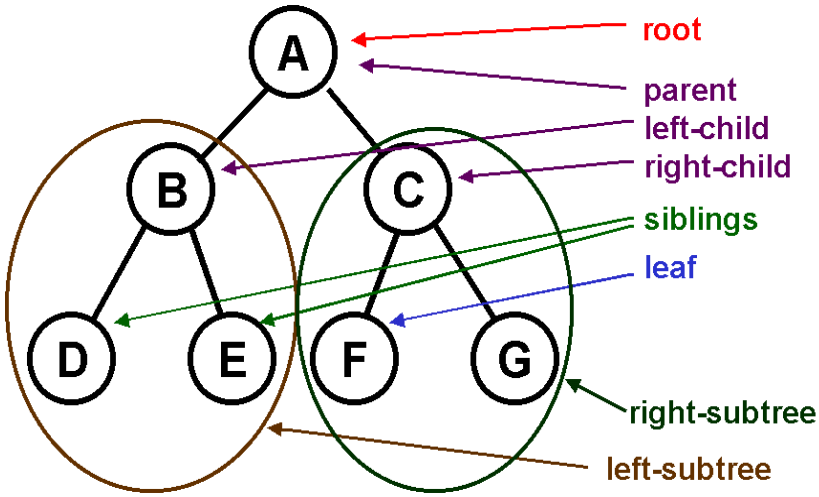
1. **Root: This is first node of the tree, referred by the 'ROOT' pointer.**
   - **➔ Root node has no parent.**
   - **➔ A tree can't have more than one root node.**



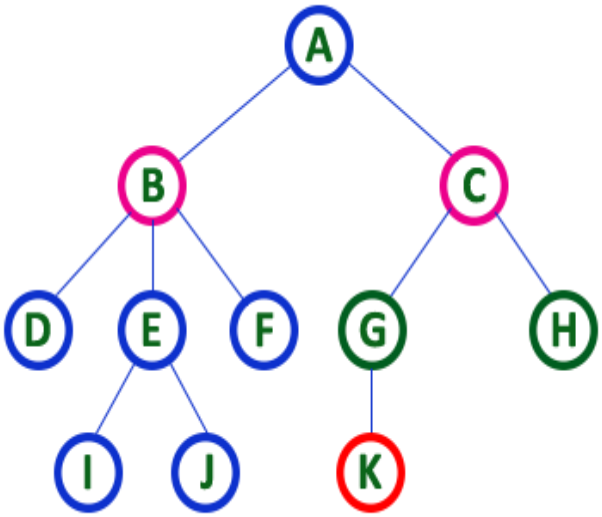
2. **Edge: The connecting link between any pair nodes in a tree is called an edge.**
   - **➔ A tree consisting of N nodes, can have maximum N-1 edges.**



3. **Parent node: The immediate predecessor node of node N is called its parent node.**
   - **➔ The parent of the parent node of a node N is called its grandparent.**
   - **➔ Parent node can also be defined as: a node having some child nodes.**



4. **Child node: The descendant node of a node N is called child node.**
   - **➔ The nodes which are connected by a link from a node N are called child nodes.**
   - **➔ A node N can have any number of child nodes.**
   - **➔ In a tree all nodes except the root node are child nodes.**



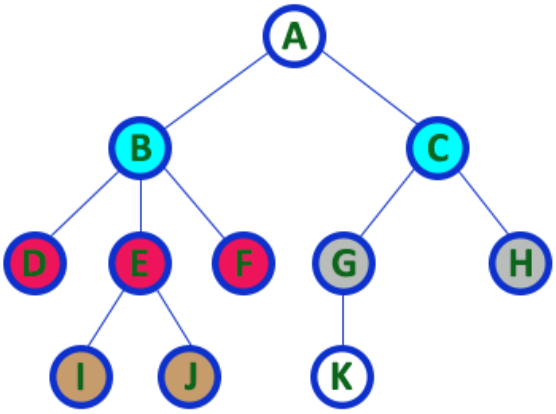
Here B & C are Children of A

Here G & H are Children of C

Here K is Child of G

- descendant of any node is called as CHILD Node

5. **Siblings: The nodes with the same parent are called siblings.**



Here B & C are Siblings

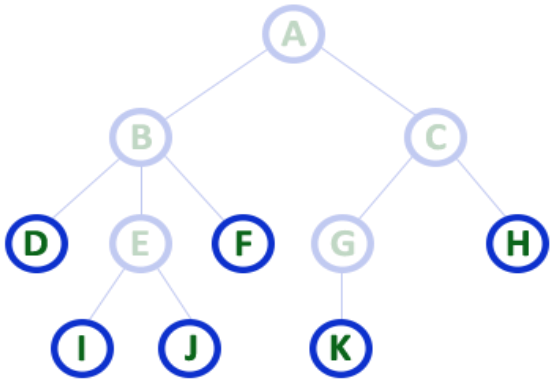
Here D E & F are Siblings

Here G & H are Siblings

Here I & J are Siblings

- In any tree the nodes which has same Parent are called 'Siblings'
- The children of a Parent are called 'Siblings'

6. **Leaf node: A node which does not have any child is called a leaf node.**
   →Leaf nodes are also called external or non-terminal nodes.

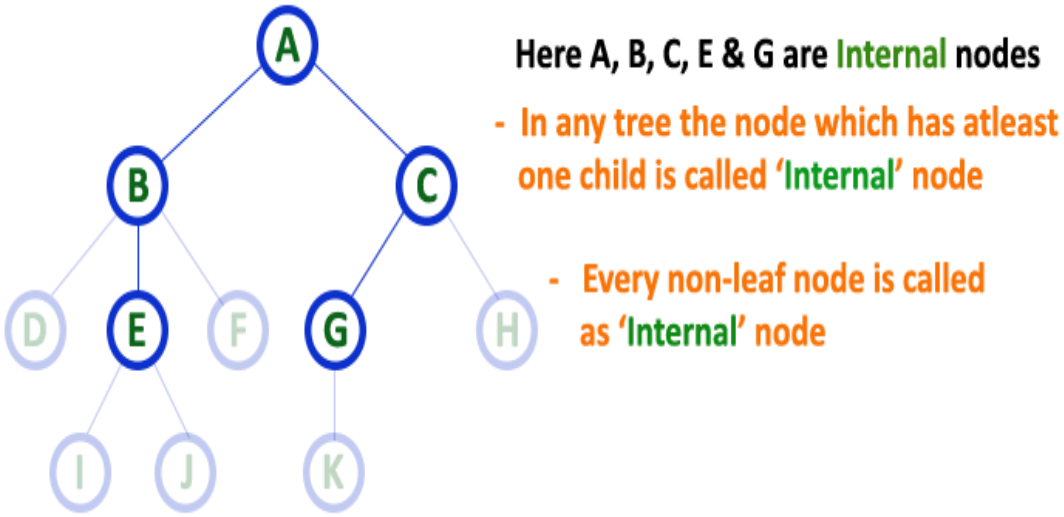


Here D, I, J, F, K & H are Leaf nodes

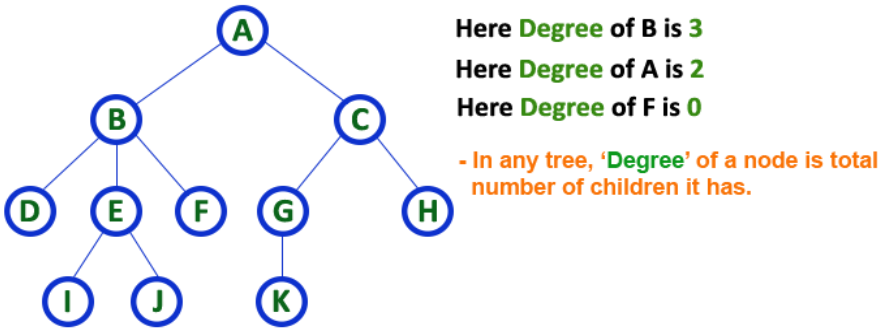
- In any tree the node which does not have children is called 'Leaf'
- A node without successors is called a 'leaf' node

→

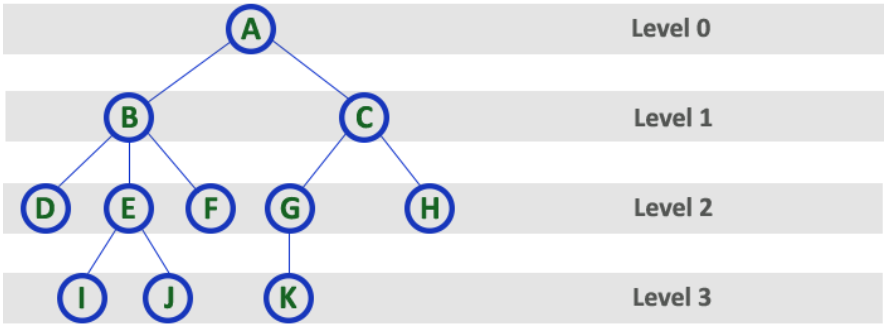
7. **Non-leaf node: The node which has at least one child is called a non-leaf node.**
   → Non-leaf nodes are also called internal or non-terminal nodes.
   → Except leaf nodes all other nodes in a tree are internal-nodes.



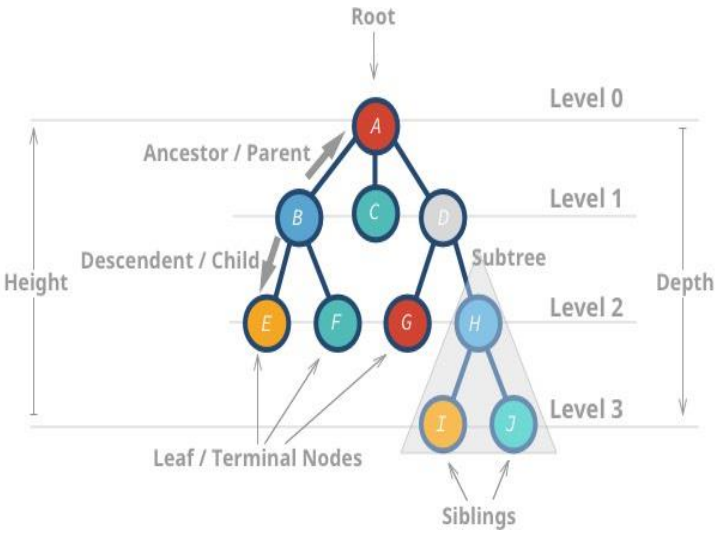
8. **Degree of a node: The total number of children of a node N is called its degree.**
   ➔ The maximum degree of any node in a tree is called degree of the tree.

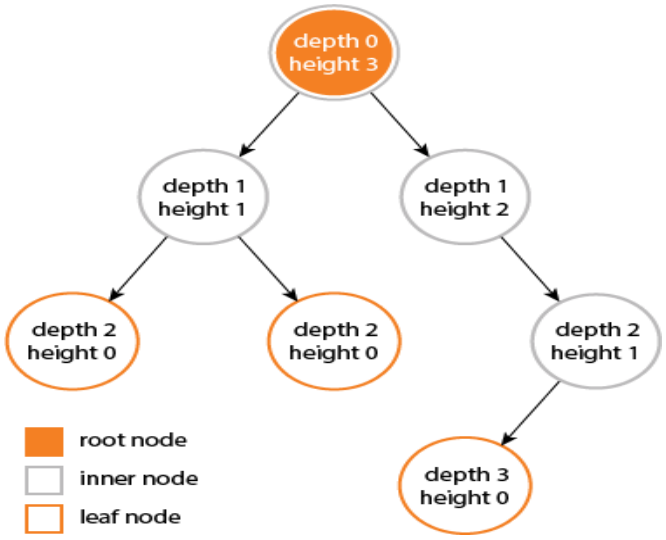


9. **Level: The entire tree structure is levelled such that, the root node lies at level 0, the children of the root node are at level 1 , children of the nodes at level 1 are at level 2 and so on…**
   ➔ The level count starts from 0 and incremented by one at each step.

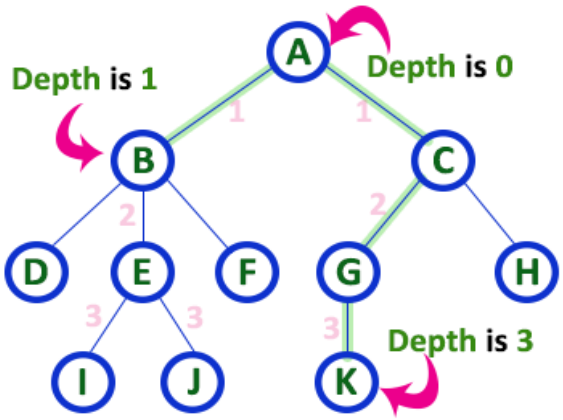


10. **Height of a node: The total number of edges from a node N to any of its descendant leaf node is called height of node N.**
    ➔ Height of root node is maximum in the entire tree.
    ➔ Height of root is called height of the tree.
    ➔ Height of leaf nodes is zero.





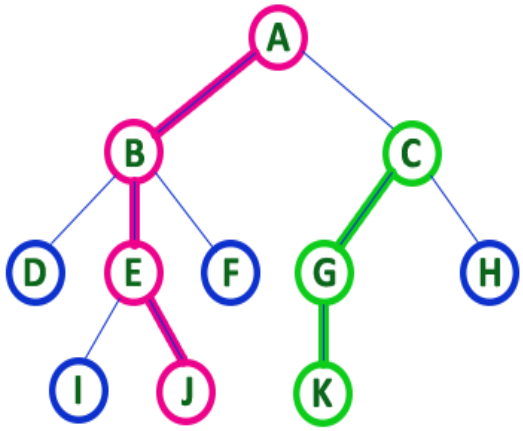
11. **Depth of a node: The total number of edges from root node connecting to a particular node N is called depth of the node N.**
    - ➔ **The level at which a node N lies is also called depth of the node N.**
    - ➔ **The maximum depth of any leaf node in a tree is called Depth of the tree.**
    - ➔ **The maximum level of the tree is also called its depth i.e maximum level of the tree is equal to depth of the tree.**
    - ➔ **Depth of the root node is zero.**
    - ➔ **Depth of the tree=height of the tree=maximum level of the tree**



Here Depth of tree is 3

- In any tree, 'Depth of Node' is total number of Edges from root to that node.
- In any tree, 'Depth of Tree' is total number of edges from root to leaf in the longest path.

12. **Path: The sequence of edges connecting any pair of nodes in a tree is called path.**
    - ➔ **A path is represented by a sequence of ordered pair of nodes.**
    - ➔ **For example : The path connecting nodes B and J in the above tree is represented as $PATH_{(B-J)}$ = { (B,E) , (E,J) }**
    - ➔ **Length of a path is the total number edges in a path.**



- 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

13. Sub-tree: The tree formed by any child of a node N is called a sub-tree.

➔ **The child node is the root of sub-tree.**

➔ **Every child will form a sub-tree on its parent node recursively.**