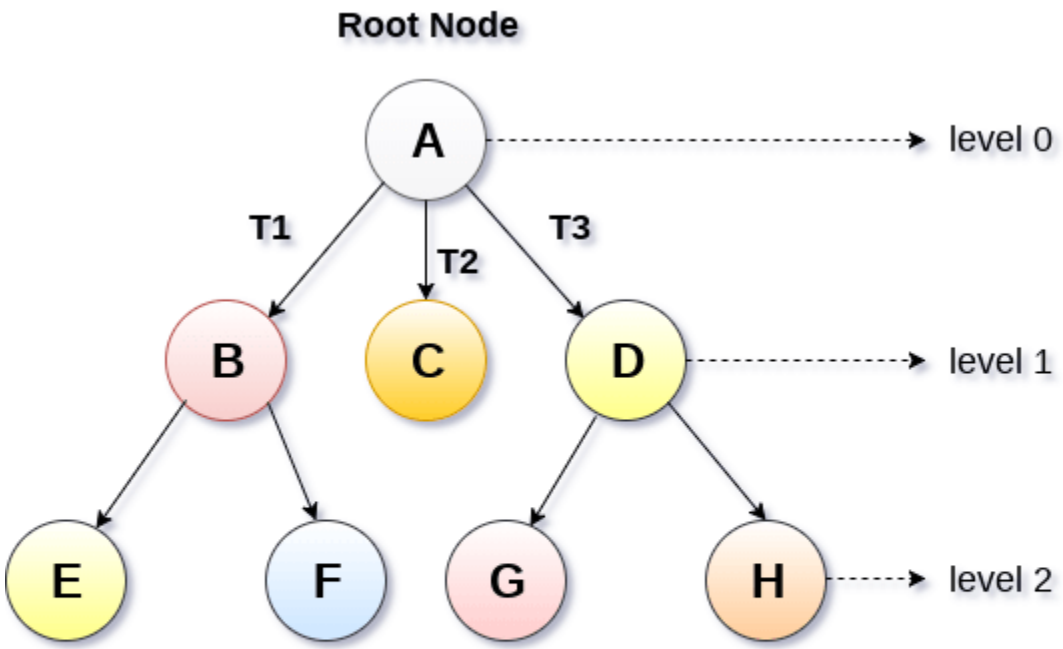
Introduction and Terminology

Tree:

- Tree is a hierarchical data structure which stores the information naturally in the form of hierarchy style.
- Tree is one of the most powerful and advanced data structures.
- It is a non-linear data structure compared to arrays, linked lists, stack and queue.
- It represents the nodes connected by edges.



Tree

Basic terminology:

1. **Root Node** - The root node is the topmost node in the tree hierarchy. In other words, the root node is the one which doesn't have any parent.

2. **Sub Tree**: If the root node is not null, the tree T1, T2 and T3 is called sub-trees of the root node.
3. **Leaf Node**: The node of tree, which doesn't have any child node, is called leaf node. Leaf nodes can also be called external nodes.
4. **Path**: The sequence of consecutive edges is called path. In the tree shown in the above image, path to the node E is A→ B → E.
5. **Ancestor node**: An ancestor of a node is any predecessor node on a path from root to that node. The root node doesn't have any ancestors. In the tree shown in the above image, the node F have the ancestors, B and A.
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.

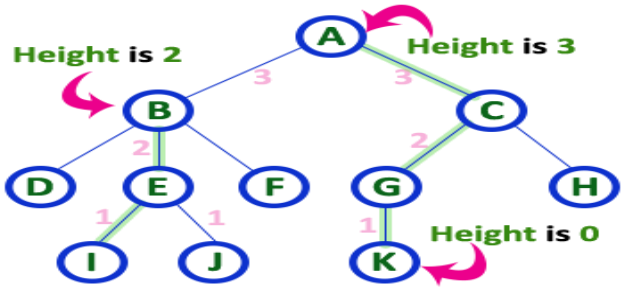
   Degree of a node is equal to number of children, a node have. the degree of node B is 2.

7. **Level Number**: Each node of the tree is assigned a level number in such a way that each node is present at one level higher than its parent. Root node of the tree is always present at level 0.
8. **Edge:** Edge is a connection between one node to another. It is a line between two nodes or a node and a leaf.
9. **Siblings:** Nodes with the same parent are called Siblings.
10. **Internal Nodes:** In a tree data structure, the node which has atleast one child is called as **INTERNAL Node**.

    Internal nodes are also called as '**Non-Terminal**' nodes. **Eg- A, B, C, D**

11. **Height:** In a tree data structure, the total number of edges from leaf node to a particular node in the longest path is called as **HEIGHT** of that Node.

    In a tree, height of the root node is said to be **height of the tree**. In a tree, **height of all leaf nodes is '0'.**



Here Height of tree is 3

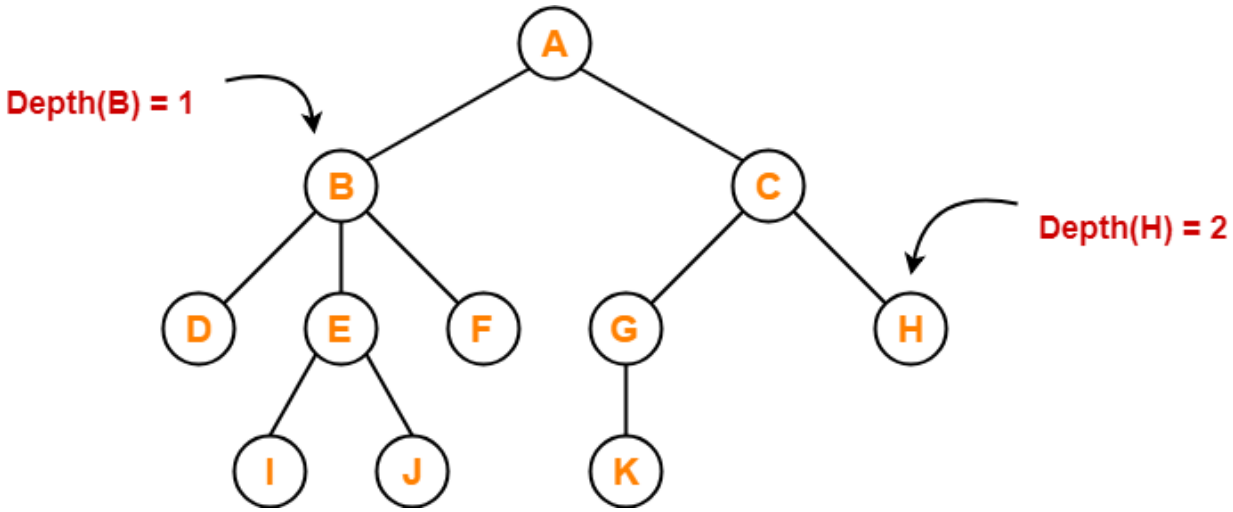
- In any tree, 'Height of Node' is total number of Edges from leaf to that node in longest path.
- In any tree, 'Height of Tree' is the height of the root node.

12. **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

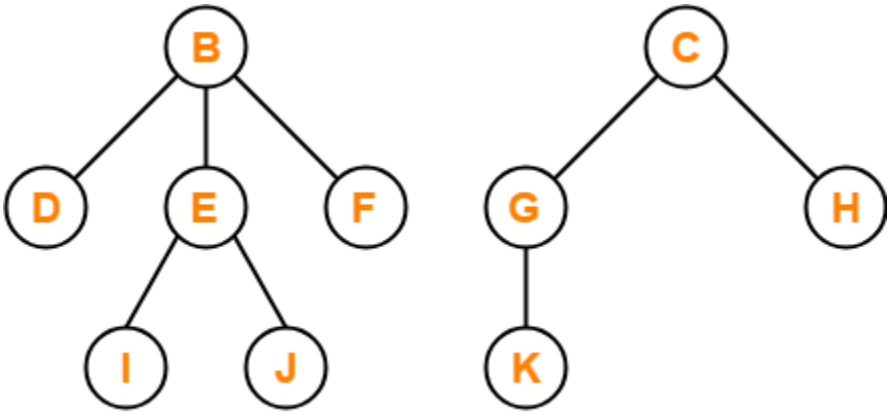
The terms "level" and "depth" are used interchangeably.



- Depth of node A = 0
- Depth of node B = 1
- Depth of node C = 1
- Depth of node 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

13. **Forest:** A forest is a set of disjoint trees.



Forest

Advantages of Tree

- Tree reflects structural relationships in the data.
- It is used to represent hierarchies.
- It provides an efficient insertion and searching operations.
- Trees are flexible. It allows to move subtrees around with minimum effort.