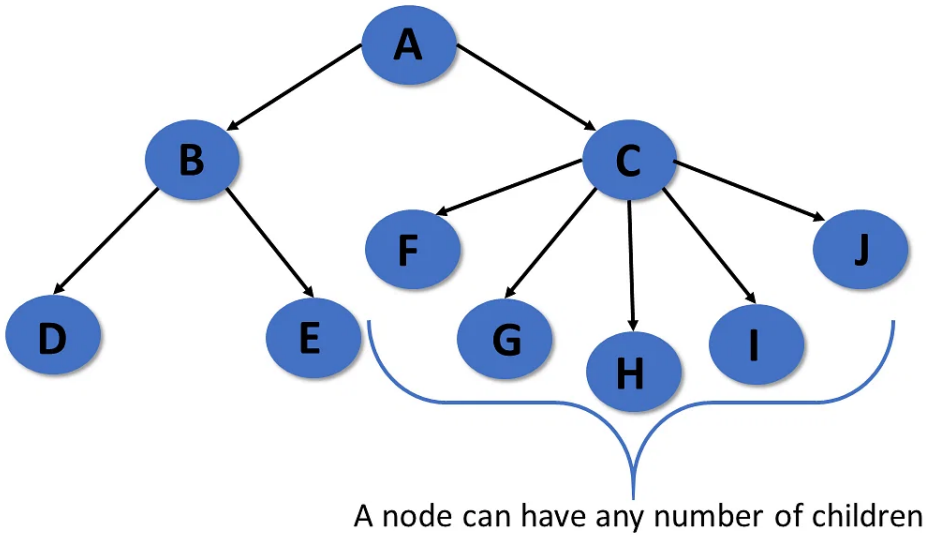
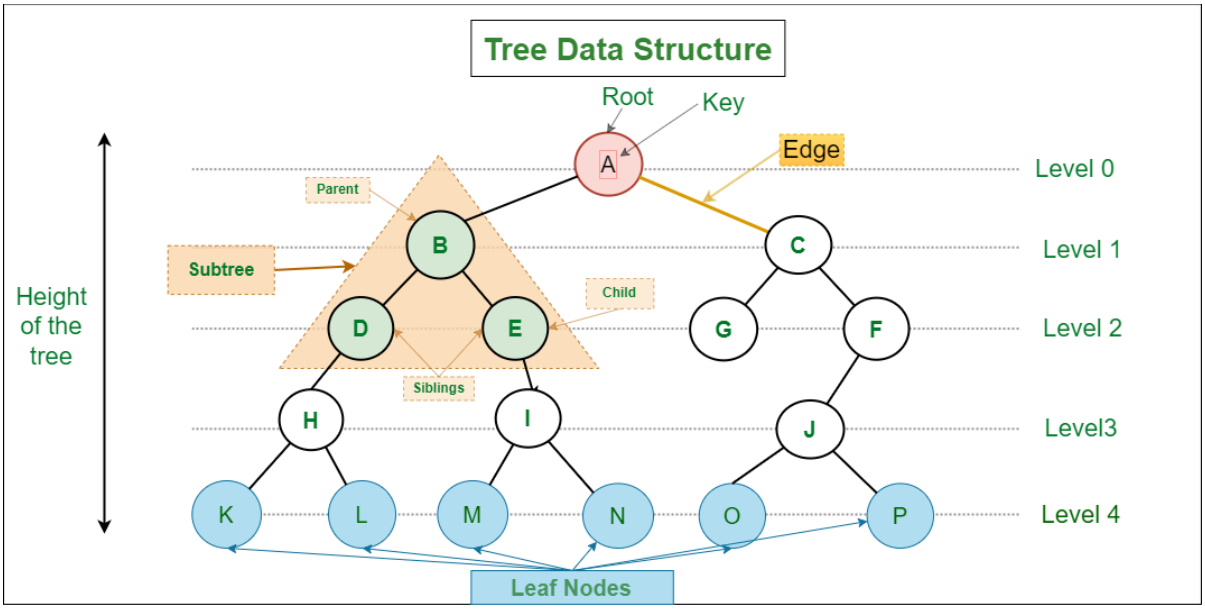
Tree Data Structure

What is a Tree data structure?

1. A Tree is a non-linear hierarchical data structure like a file system in computers. A tree is a collection of nodes and each node is connected to each other through edges, where each node in a tree can point to any number of nodes.



2. The **topmost** node of the tree is called the **root node** , and the nodes below it are called the **child nodes**. Each node can have multiple child nodes, and these child nodes can also have their own child nodes.



Types of Tree Data Structure:

The following are the different types of trees data structures:

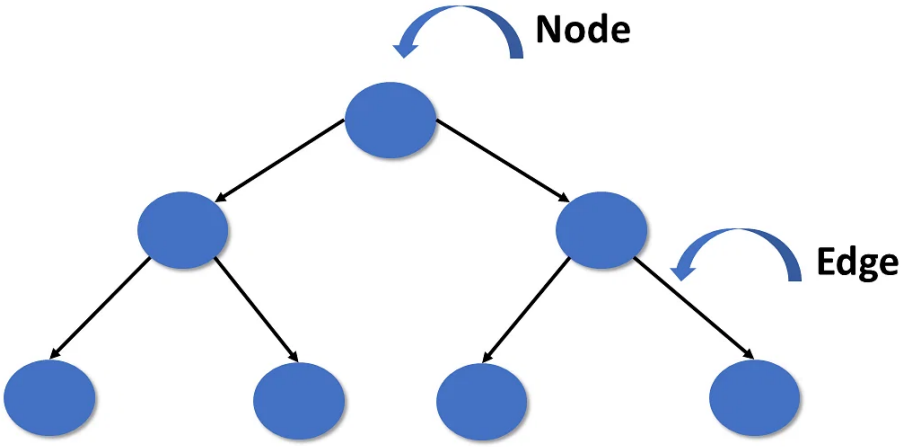
- Binary Tree
- Binary Search Tree (BST)
- AVL Tree
- B-Tree

Why Tree is considered a non-linear data structure?

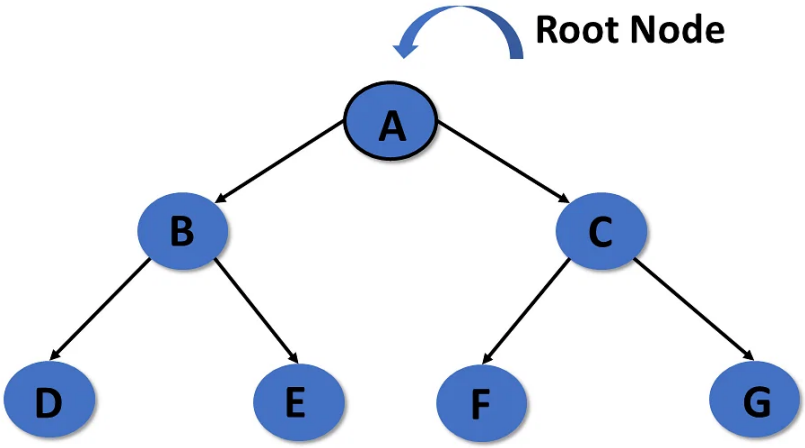
The data in a tree are not stored in a sequential manner i.e., they are not stored linearly. Instead, they are arranged on multiple levels or we can say it is a hierarchical structure. For this reason, the tree is considered to be a non-linear data structure.

Basic Terminologies In Tree Data Structure:

1. **Node**
    a. A `tree` is a collection of entities called `nodes`. Nodes are connected by `edges`. Each `node` contains a `value` or `data`, and it may or may not have a `child node` .

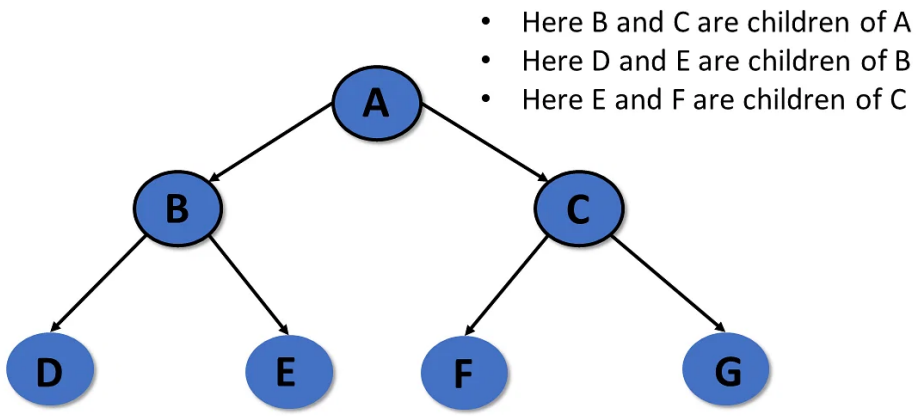


2. **Root Node**
    a. The first node or the topmost node of the the tree is called the root node.



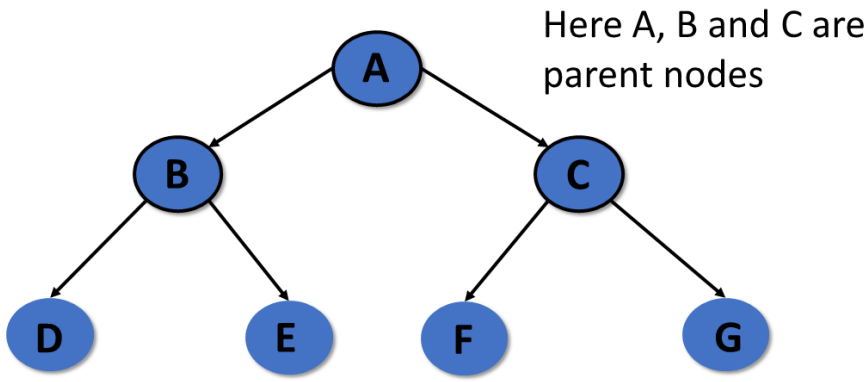
3. **Child Node or Children**

a. Child is a node that has a parent node.



4. **Parent Node or Parent**

a. Parent is a node that has an edge to a child node.

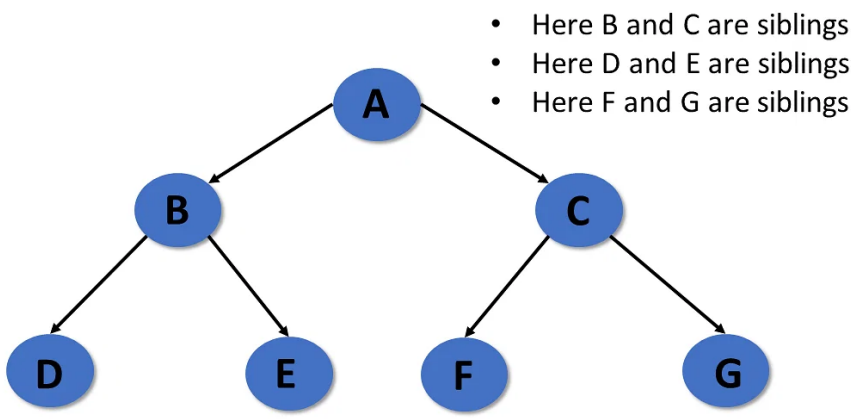


5. **Siblings (means Real Brother and Sister)**

a. Children of the same parent node are called siblings.

b. In simple word, if two or more child nodes having same parent node, then these child nodes is considered as siblings.

c. In Simple Words, Siblings are Children of same parent.



6. **Ancestor (means पूर्वज)**

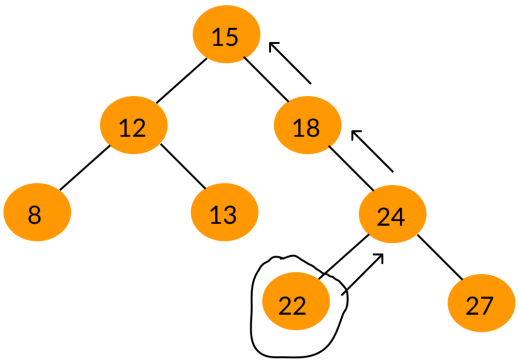
a. Suppose we have to find the ancestors of node 22 in the below given tree.

b. Now, to find the ancestors of node 22 what you have to do is,

i. Pick the node whose ancestor you want to find. Now, move toward the root node, and each node that comes into the path from that node to the root node, that nodes are the ancestor of that particular node.

c. So, Ancestors of 22 are 24, 18, and 15.

Ancestors of a node in a binary tree

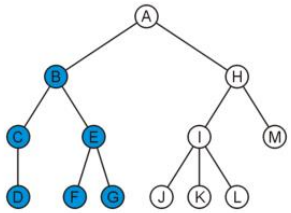


Ancestors of 22 are 24, 18 and 15

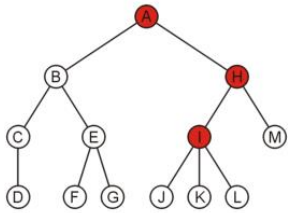
7. **Descendant (means वंशज)**
    a. Suppose we have to find the descendants of node B in the below given tree.
    b. Now, to find the descendants of node B what you have to do is,
        i. Pick the node whose descendant you want to find. Now, move toward the leaf node, and each node that comes into the path from that node to the leaf node, that nodes are the descendants of that particular node.

Terminology

The descendants of node B are B, C, D, E, F, and G:



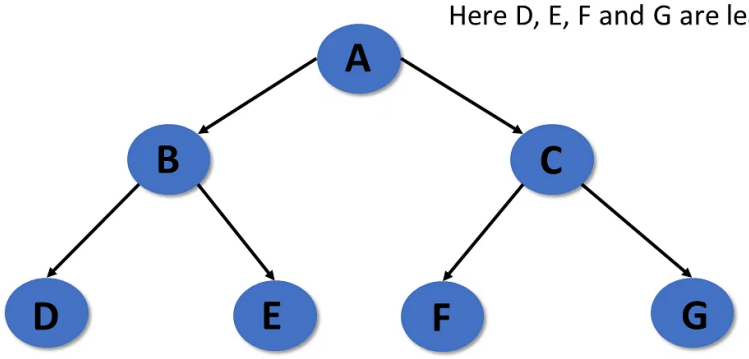
The ancestors of node I are I, H, and A:



8. **Leaf Node**

   a. A leaf is a node that does not have a child node in the tree.

Here D, E, F and G are leaf nodes.



Properties of a Tree:

- **Number of edges:**
  - An edge can be defined as the connection between two nodes. If a tree has N nodes then it will have **(N-1) edges.** There is only one path from each node to any other node of the tree.
- **Depth of a node:**
  - **Depth of a node = Number of edges in the path from the root node of the tree to the node.**
- **Height of a node:**
  - The height of a node can be defined as the length of the longest path from the node to a leaf node of the tree.
  - Height of a Node = number of edges from that node to deepest leaf node.
- **Height of the Tree:**
  - The height of a tree is the length of the longest path from the root of the tree to a leaf node of the tree.
- **Degree of a Node:**
  - The total count of subtrees attached to that node is called the degree of the node. The degree of a leaf node must be **0**. The degree of a tree is the maximum degree of a node among all the nodes in the tree.

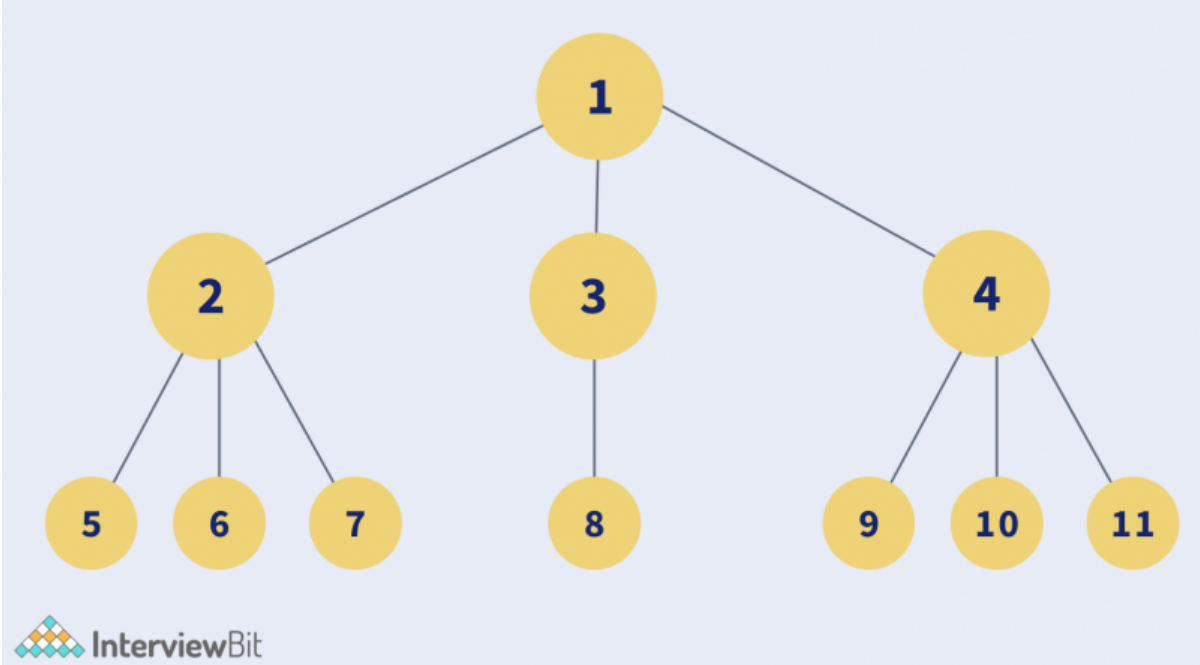
Some more properties are:

- Traversing in a tree is done by depth first search and breadth first search algorithm.
- It has no loop and no circuit
- It has no self-loop
- Its hierarchical model.

What is N-ary Tree in Tree Data Structure ?

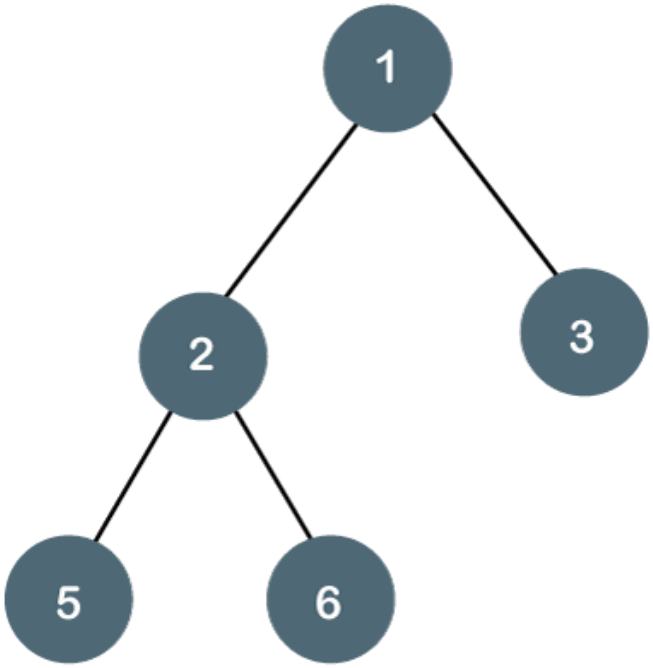
N-ary trees are tree data structures that allow us to have up to **n** children nodes for each of the nodes, differing from the standard binary trees which allow only up to **2** children nodes for each node.

In simple words, In N-ary Tree each node can have “n” number of children (child node).



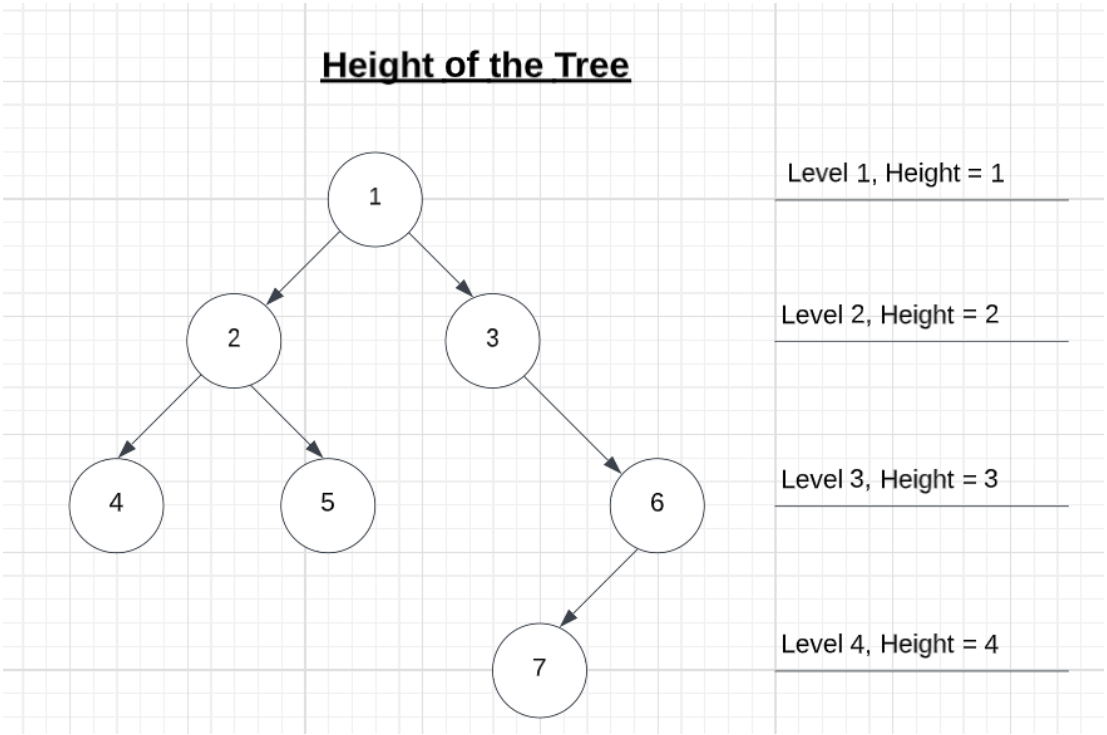
What is Binary trees ?

1. In computer science, a binary tree is a tree data structure in which each node has at the most two children, which are referred to as the “left child” and the “right child.”
2. **Binary Tree:** A tree is called Binary Tress, if each node has 0 child, 1 child or 2 child.
3. **Empty tree is also a valid binary tree.**
4. **In Short:** Binary Tree ⇒ Each Node has ≤ 2 child.
5. **In Binary Tree, each node has 3 attributes**
    a. Data
    b. Pointer to left child.
    c. Pointer to right child.

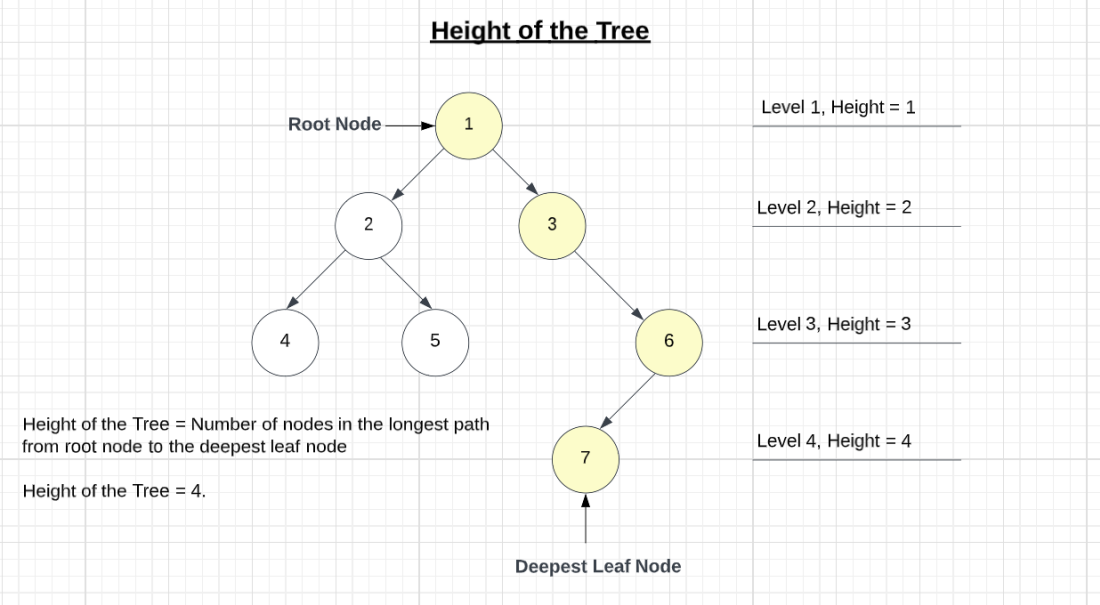


Properties of Tree :

1. **Height of the Tree**
    a. Height of the tree is the maximum levels we have in a tree.



b. or, The height of a tree is the length of the longest path from the root node of the tree to a leaf node of the tree. So, ***the number of nodes in this longest path from root to the deepest leaf node is the height of the tree.***



c. ***Note that the Height and Depth of the tree are same but the Height and Depth of the node may be different.***

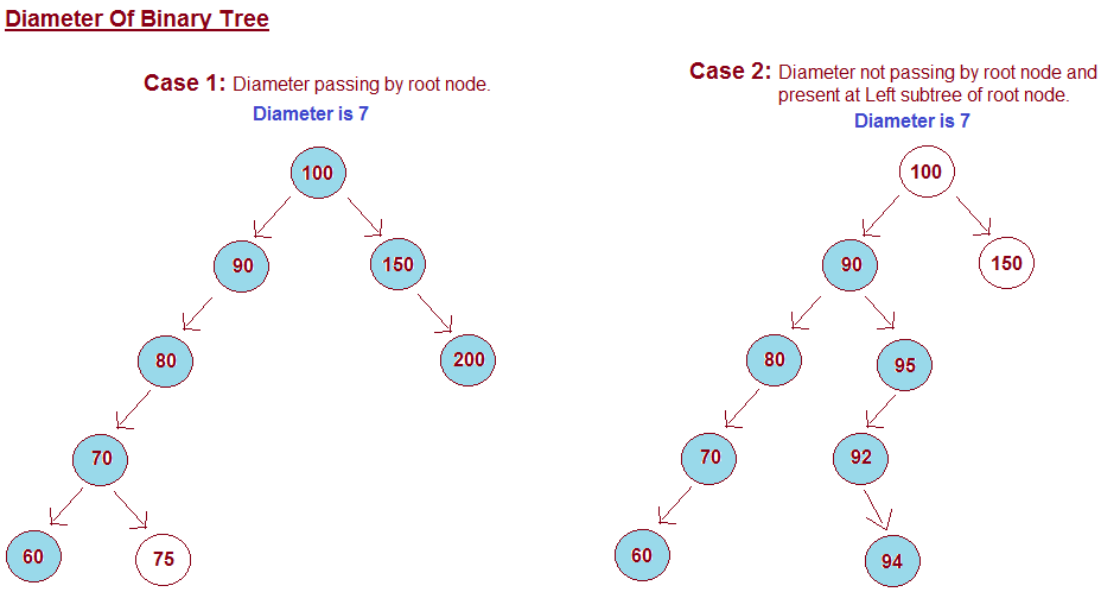
2. **Diameter of a Tree**

a. *Diameter of Tree* = The diameter of a binary tree is the **length** of the longest path between any two nodes in a tree. This path may or may not pass through the root node.

b. *The length of a path between two nodes is represented by the number of edges between them.* ***Length = number of edges in the path.***

c. Or, *The length of a path between two nodes is represented by the number of nodes between them.* ***Length = number of nodes in the path.***

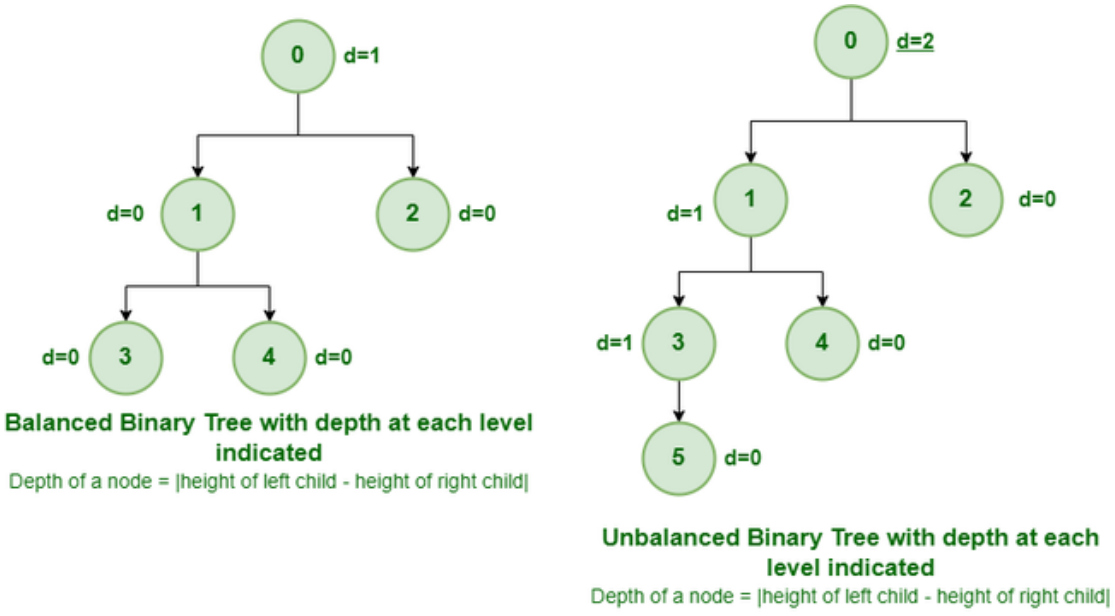
d. Whether to take **length = the number of edges in the path** or **length = the number of nodes in the path**. It is given in the question.



Types of Binary Tree :

1. **Balanced Binary Tree**

   a. *A Binary Tree is Balanced if the difference between the height of its left subtree and right subtree is* ***not more than 1 for all nodes of the tree.***



2. **Skewed Binary Tree**

a. A skewed binary tree is a type of binary tree in which all the nodes have only either one child or no child.

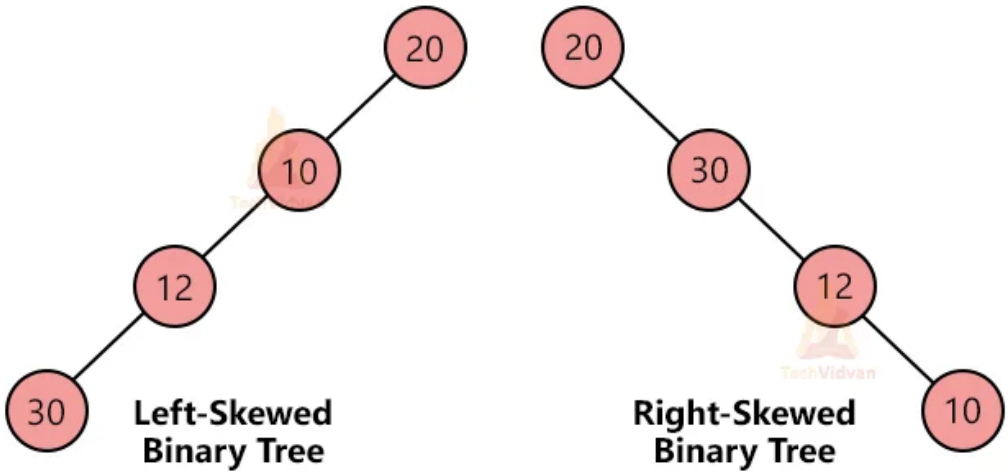
b. Each node has only 1 child node or no child node.

c. **Types of Skewed Binary trees,**

   i. There are 2 types of Skewed Binary Tree,

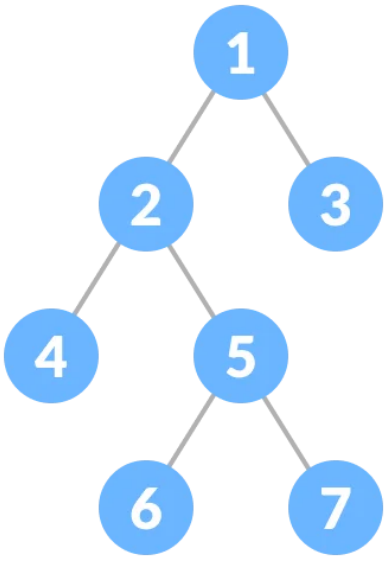
      1. **Left Skewed Binary Tree: Each node only have left child or no child.**

      2. Right **Skewed Binary Tree: Each node only have right child or no child.**

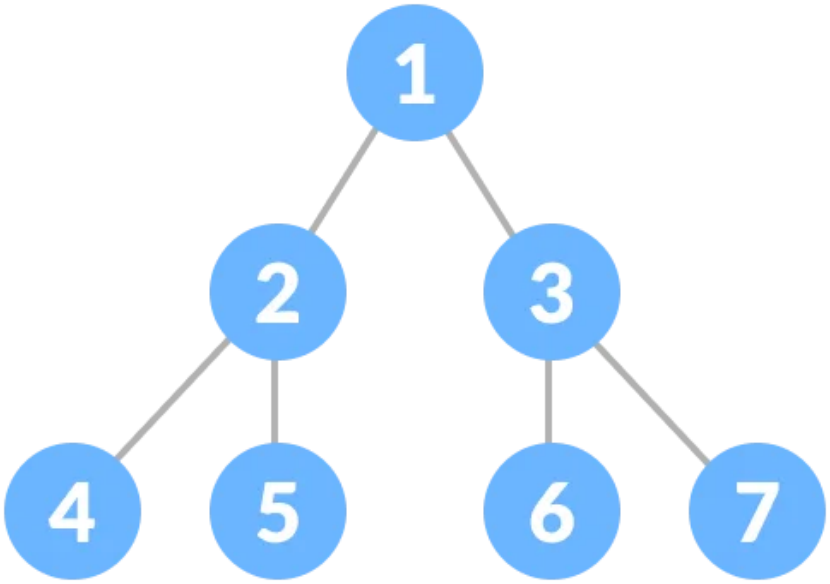


3. **Full Binary Tree**

   a. In Full Binary Tree, each node has either two child or no child.



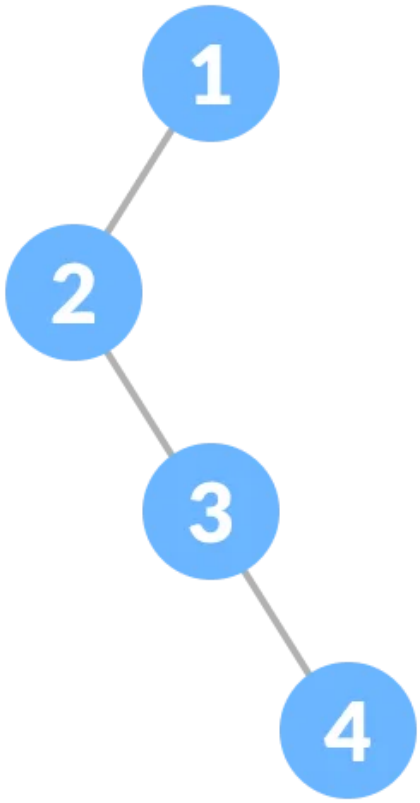
4. **Perfect Binary Tree**
    a. In Perfect Binary Tree, Each node has exactly two child and all the lead nodes are at same level.



5. **Degenerate or Pathological Tree**

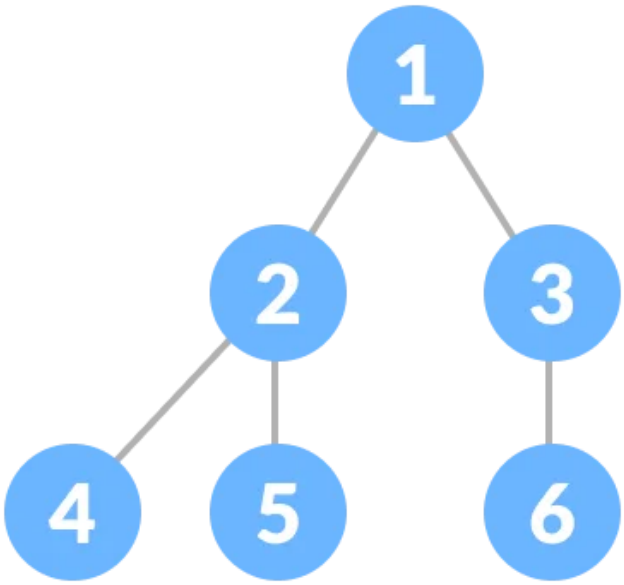
a. A degenerate or pathological tree is the tree having a single child either left or right.

b. Each node has single child node either left or right.



6. **Complete Binary Tree**

   a. A complete binary tree is a binary tree in which all the levels are completely filled except the last level, which is filled from the left.



What is Binary Search Tree ?

The properties that separate a binary search tree from a regular binary tree is:

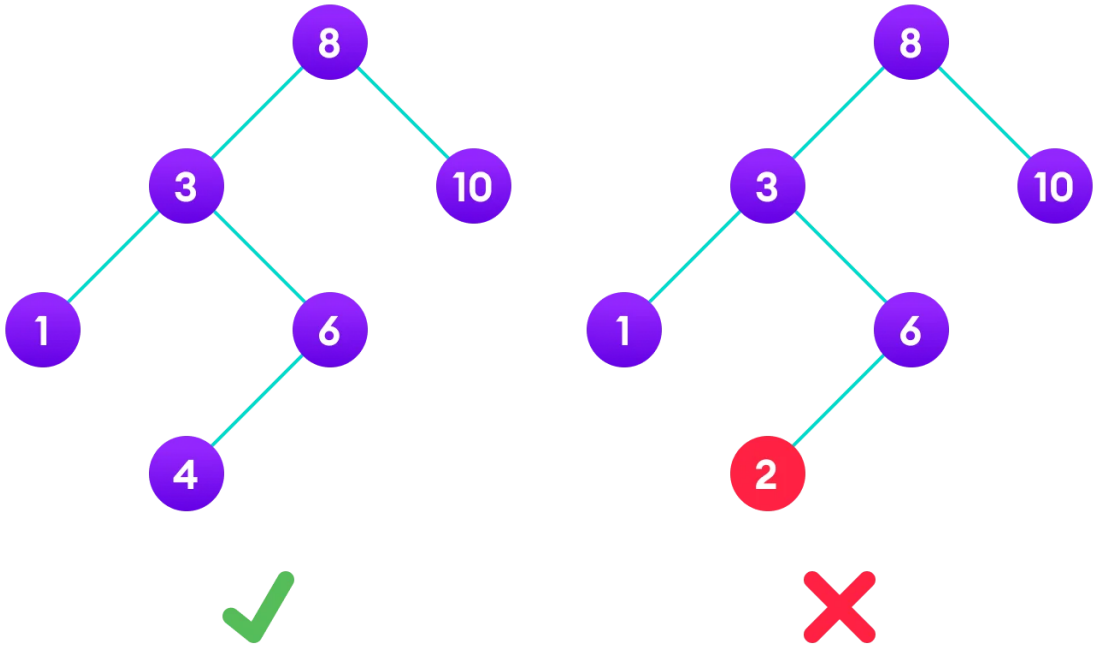
1. For Every Node,
    a. All nodes of left subtree are less than the root node.
    b. All nodes of right subtree are more than the root node.

```
BST = node values at Root -> left <= Root value and node values at Root -> right >= Root value.
So , for every node if this property is statisfied then this tree is considered as Binary Search Tree (BST).

Property:- InOrder Traversal of Binary Search Tree (BST) comes out in sorted order.
```

Advantages of Searching in a Binary Search Tree:

You can easily apply Binary Search on BST to search an element. So, searching in BST having time complexity o(log n) and space complexity is o(h), h is the heigh of the BST.



Note: The In-Order Traversal of Binary Search Tree (BST) comes out in sorted order.