Date: 25-04-2022 **Assignment 6**

**1) Describe what kind of data-structure is tree**

A tree data structure is a **non-linear data structure** because it does not store in a sequential manner. It is a hierarchical structure as elements in a Tree are arranged in multiple levels. In the Tree data structure, the topmost node is known as a root node. Each node contains some data, and data can be of any type.

**2) List and explain binary tree, strictly binary tree, Binary search tree, red , black tree , Left & Right skewed tree , Balanced tree etc**

**1.** **Binary Tree :** Binary name itself suggests two numbers, i.e., 0 and 1. In a binary tree, each node in a tree can have utmost two child nodes. Here, utmost means whether the node has 0 nodes, 1 node or 2 nodes.

**2.** **Strictly Binary Tree** : A Binary tree is said to be Full Binary Tree, if all its internal nodes has 0 or 2 [children](https://codepumpkin.com/tree-data-structure-terminologies-set-1/#children). In other words, if all the nodes other than leaf nodes has 0 or 2 children, then that it is Full Binary Tree.

In other words, all of the nodes in a **Full or strictly binary tree**are of degree zero or two, never degree one.

**3. Complete Binary Tree :**  A Binary tree is said to be complete Binary Tree if all levels are completely filled except possibly the last level and the last level has all keys as left as possible. Don't confuse it with Perfect Binary Tree. Lets define Complete Binary Tree with respect to Perfect Binary Tree.

A Perfect Binary Tree whose rightmost leaves (perhaps all) on the last level have been removed is called Complete Binary Tree.

**4. Skewed Binary Tree :** A binary tree, which is dominated solely by left child nodes or right child nodes, is called a skewed binary tree, more specifically left skewed binary tree, or right skewed binary tree.

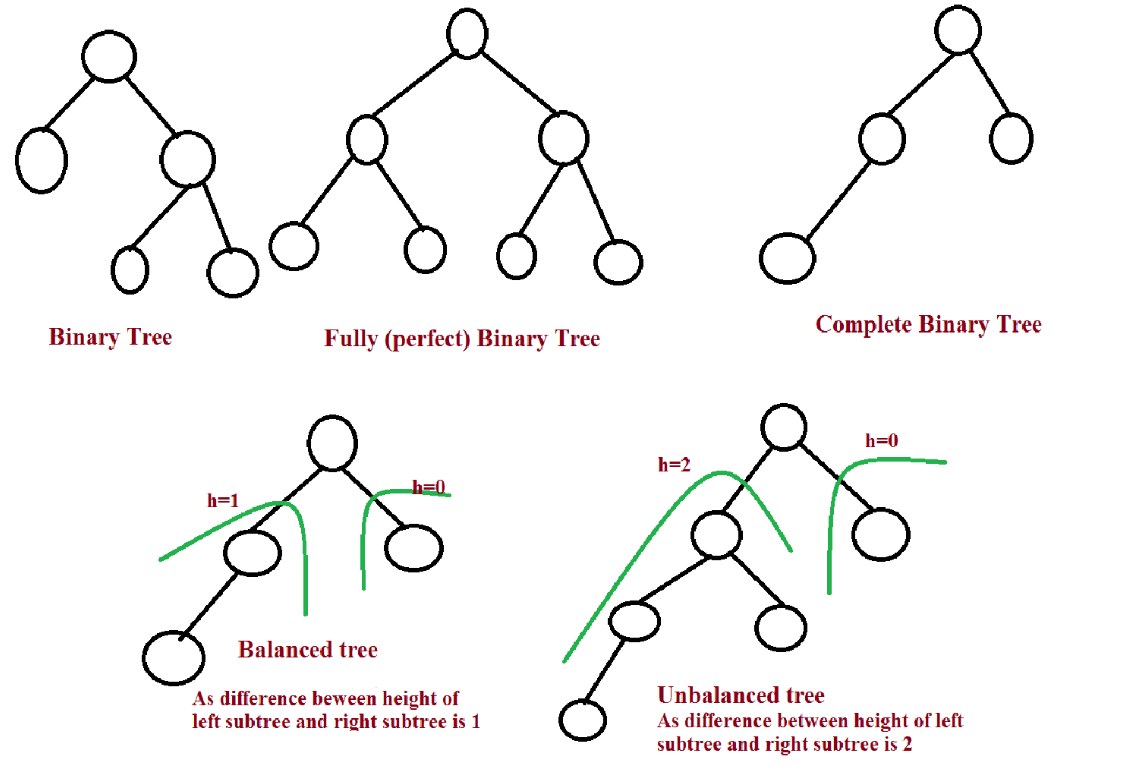
**5.** **Binary Search Tree** : Binary search tree is a non-linear data structure in which one node is connected to **n** number of nodes. It is a node-based data structure. A node can be represented in a binary search tree with three fields, i.e., data part, left-child, and right-child. A node can be connected to the utmost two child nodes in a binary search tree, so the node contains two pointers (left child and right child pointer).  
Every node in the left subtree must contain a value less than the value of the root node, and the value of each node in the right subtree must be bigger than the value of the root node.

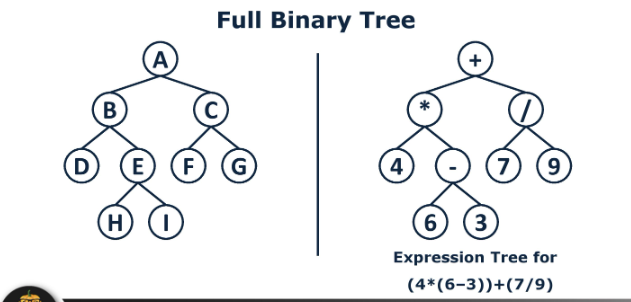
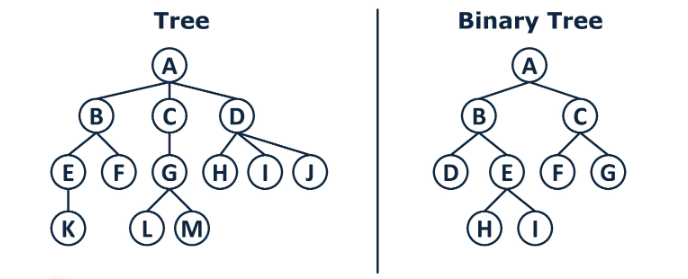
**6.** **Red-Black Tree** : **The red-Black tree** is the binary search tree. **The red-black tree** is a self-balancing binary search tree. In the Red-black tree, a maximum of 2 rotations are required to balance the tree. It contains one extra bit that represents either the red or black color of a node to ensure the balancing of the tree.

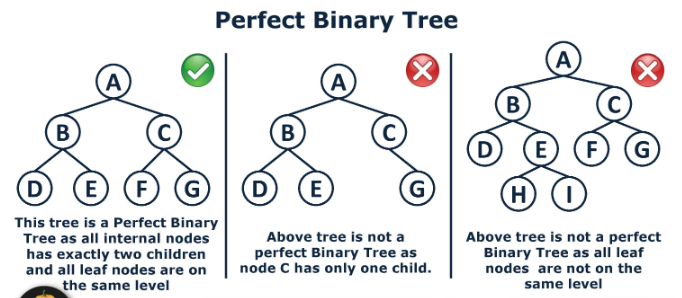
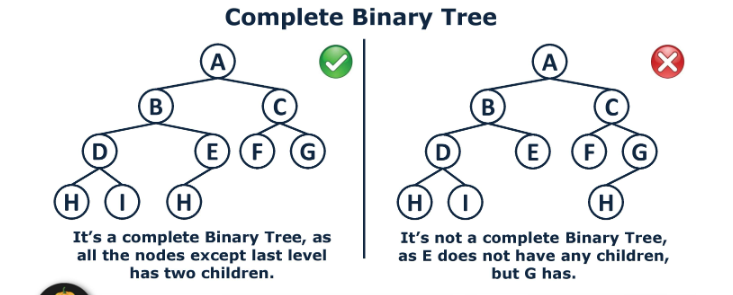
**7.** **Left & Right Skewed Tree** : A binary tree, which is dominated solely by left child nodes or right child nodes, is called a skewed binary tree, more specifically left skewed binary tree, or right skewed binary tree.

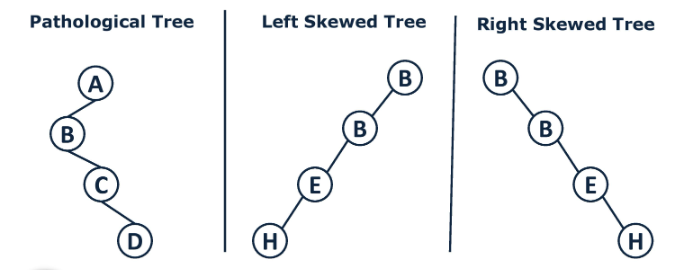
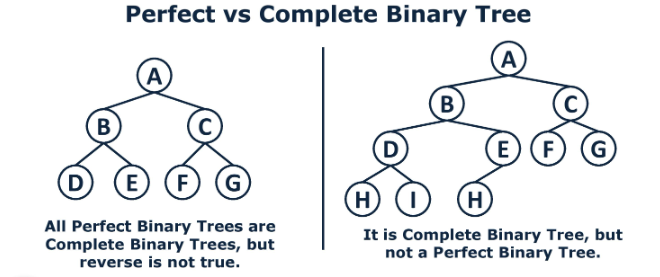
**8.** **Balanced Tree** : Binary tree is called Balanced Binary Tree, if difference of left and right subtree height is maximum one for all the nodes.

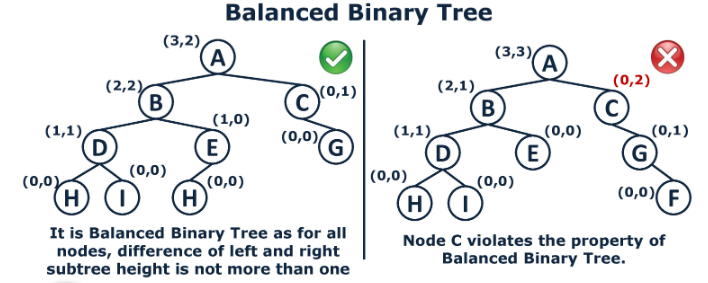
**9.** **AVL Tree** : AVL tree is also a height balancing binary search tree.\



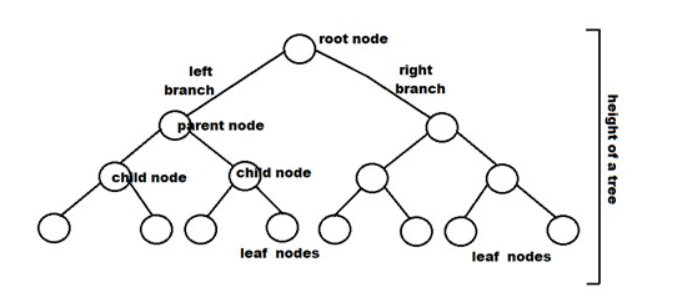






**3) Define and understand various terminologies wrt tree i.e height, depth, rootnode, leaf node, level etc**



* **Root:** The root node is the topmost node in the tree hierarchy. In other words, the root node is the one that doesn't have any parent. In the above structure, node numbered 1 is **the root node of the tree.** If a node is directly linked to some other node, it would be called a parent-child relationship.
* **Child node:** If the node is a descendant of any node, then the node is known as a child node.
* **Parent:** If the node contains any sub-node, then that node is said to be the parent of that sub-node.
* **Sibling:** The nodes that have the same parent are known as siblings.
* **Leaf Node:-** The node of the tree, which doesn't have any child node, is called a leaf node. A leaf node is the bottom-most node of the tree. There can be any number of leaf nodes present in a general tree. Leaf nodes can also be called external nodes.
* **Internal nodes:** A node has atleast one child node known as an ***internal***
* **Ancestor node:-** An ancestor of a node is any predecessor node on a path from the root to that node. The root node doesn't have any ancestors. In the tree shown in the above image, nodes 1, 2, and 5 are the ancestors of node 10.
* **Descendant:** The immediate successor of the given node is known as a descendant of a node. In the above figure, 10 is the descendant of node 5.

**4) Draw a binary search tree for given example 12,35, 67, 89,40,56? List some realtime applications of Tree-data structure?**

Real time Application:

1) Insert new file, delete file, search folder

2] different flowcharts

3] Company Organisation Structures: A report to B, B and C report to D...

**5) Perform different traversals for given Binary tree**

**6) Why it is said that searching a node in a binary search tree is efficient than that of a simple binary tree?**

Binary Search Tree allows for fast retrieval of elements stored in the tree as **each node key is thoroughly compared with the root node, which discards half of the tree**. **A Binary search tree is a tree that follows some order to arrange the elements, whereas the binary tree does not follow any order**.

**7) There Are 8, 15, 13, 14 Nodes Were There In 4 Different Trees. Which Of Them Could Have Formed A Full Binary Tree?**

**8) Which one is faster? A binary search of a ordered set of elements in an array or a sequential search of elements?**

Binary search is of order log n whereas sequential search is of order n,   
So, Binary search is faster..

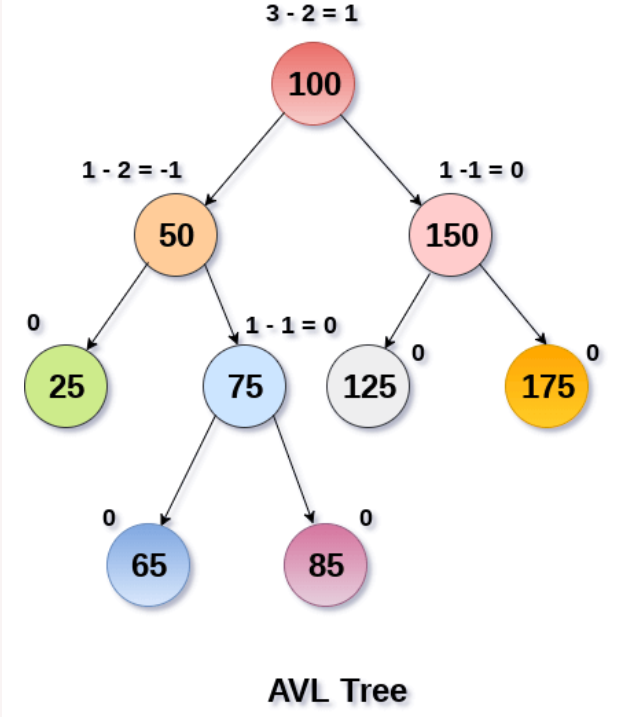
**9) Explain AVL tree?**

AVL Tree can be defined as height balanced binary search tree in which each node is associated with a balance factor which is calculated by subtracting the height of its right sub-tree from that of its left sub-tree.

Tree is said to be balanced if balance factor of each node is in between -1 to 1, otherwise, the tree will be unbalanced and need to be balanced.

**Balance Factor (k) = height (left(k)) - height (right(k))**

1] If balance factor of any node is 1, it means that the left sub-tree is one level higher than the right sub-tree. 2] If balance factor of any node is 0, it means that the left sub-tree and right sub-tree contain equal height. 3] If balance factor of any node is -1, it means that the left sub-tree is one level lower than the right sub-tree.



\*\*Programs

1)WAP to implement a Binary tree and perform following operations a)Create a binary tree b)Insert a node c)Delete anode d)Perform Preorder, Inorder, Postorder traversals on tree

2)WAP to implement a Binary Search tree and perform following operations a)Create a binary tree b)Insert a node c)Delete anode d)Perform Preorder, Inorder, Postorder traversals on tree