

30/11/2023

# BINARY TREE CLASS - 1

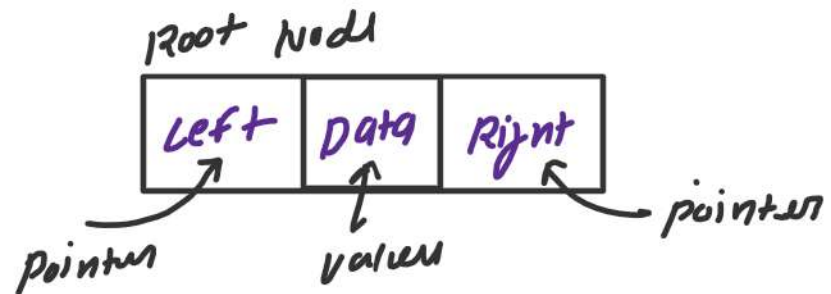
## HOMEWORK

---

## Types of Binary Tree

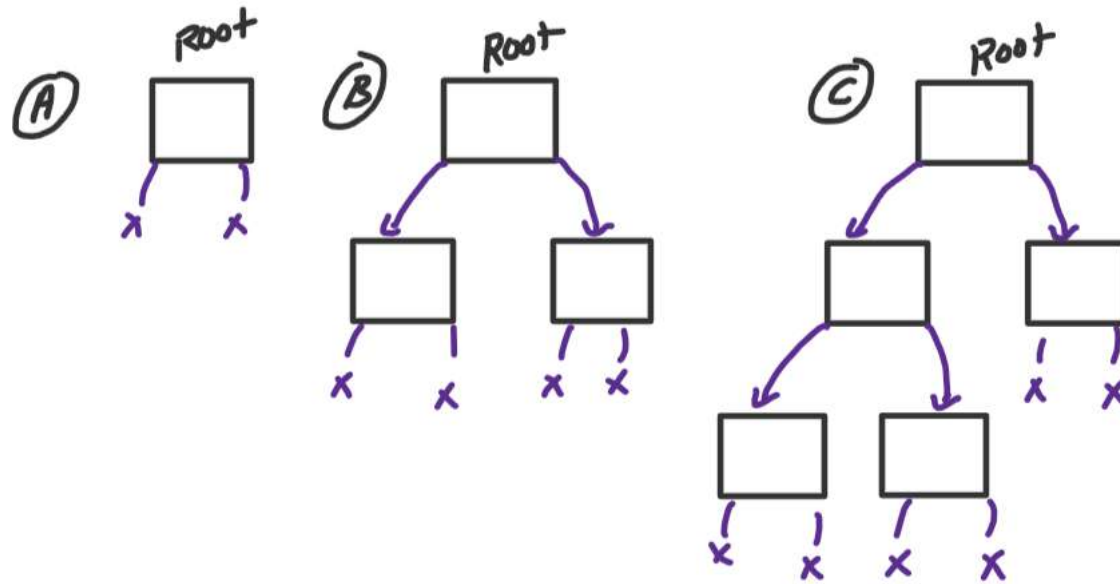
Types of Binary Tree based on the number of children:

1. Full Binary Tree
2. Perfect Binary Tree
3. Complete Binary Tree
4. Degenerate or Pathological Tree
5. Skewed Binary Tree
6. Balanced Binary Tree



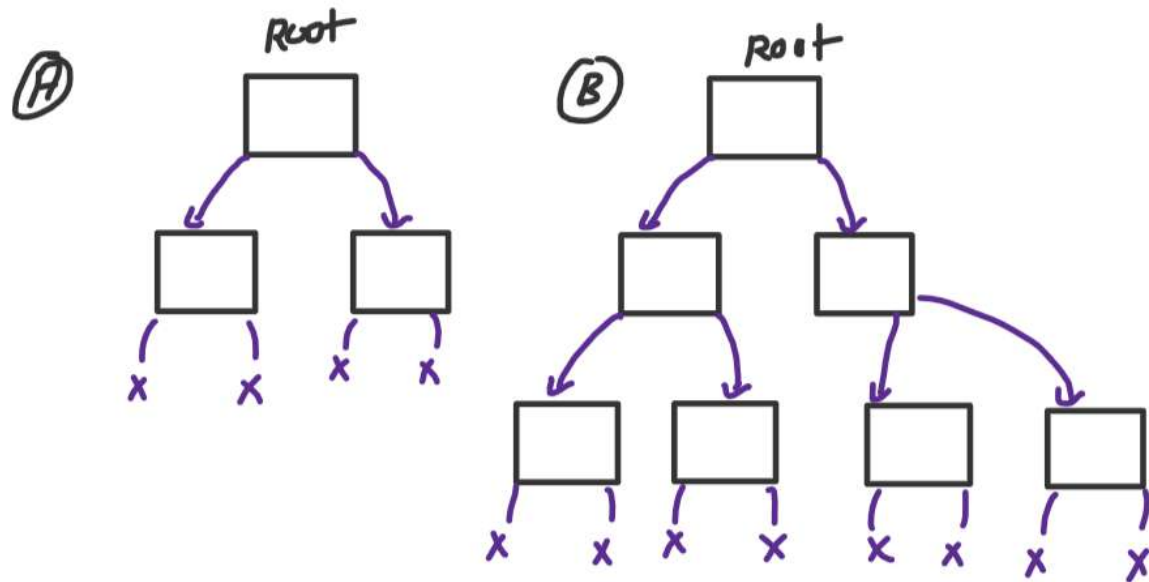
### 1. Full Binary Tree

A full Binary tree is a special type of binary tree in which every parent node/internal node has either two or no children.



## 2. Perfect Binary Tree

A perfect binary tree is a type of binary tree in which every internal node has exactly two child nodes and all the leaf nodes are at the same level.



### 3. Complete Binary Tree

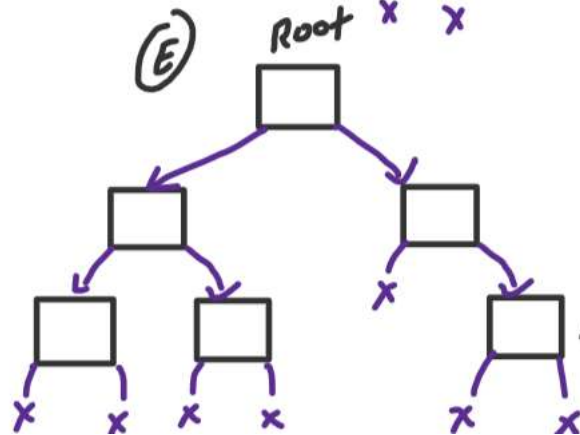
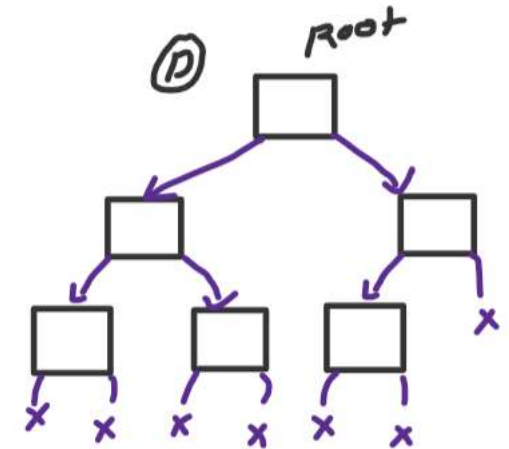
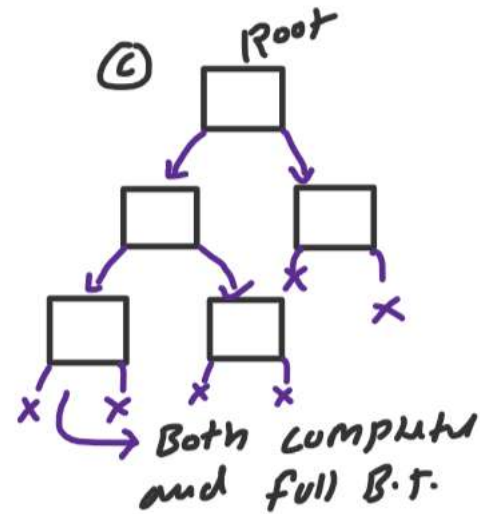
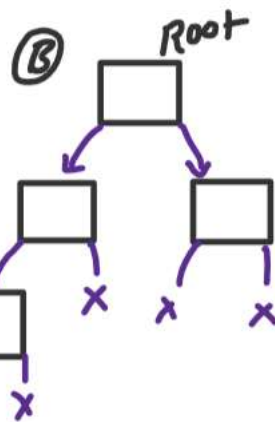
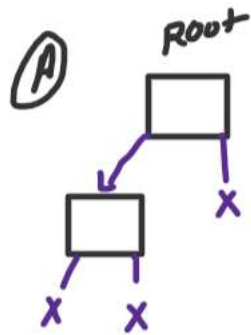
A complete binary tree is just like a full binary tree, but with two major differences

I. Every level must be completely filled

II. All the leaf elements must lean towards the left.

III. The last leaf element might not have a right sibling

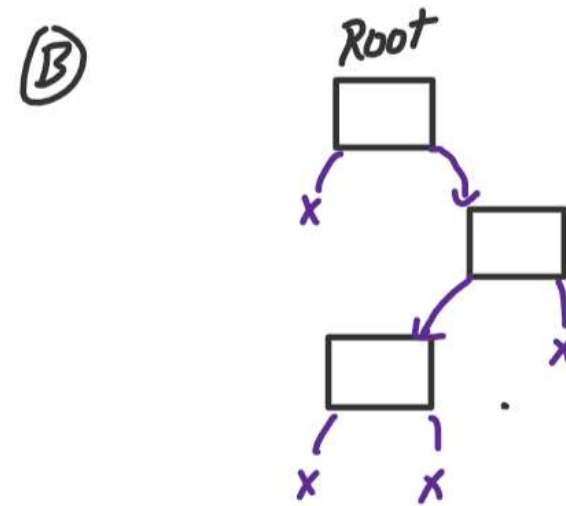
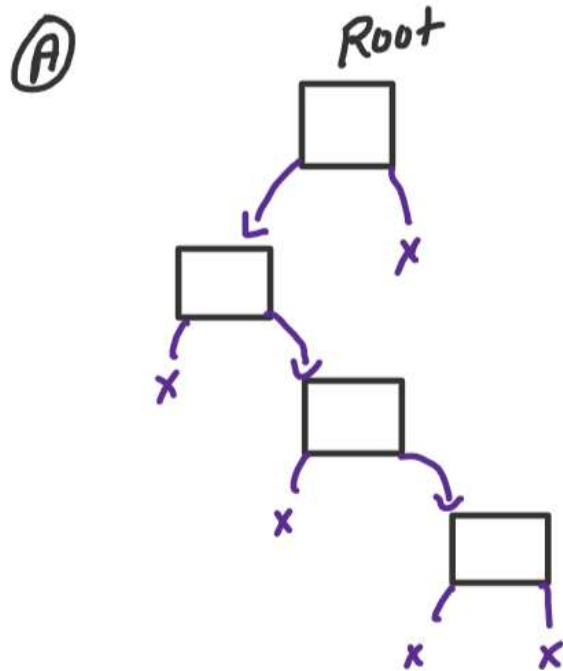
i.e. a complete binary tree doesn't have to be a full binary tree.



Neither complete nor full B.T.

#### 4. Degenerate or Pathological Tree

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

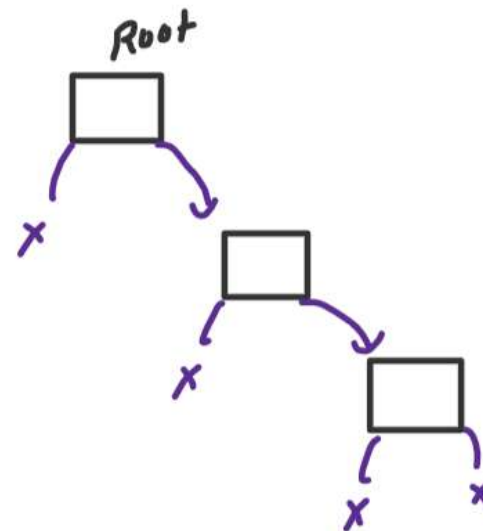
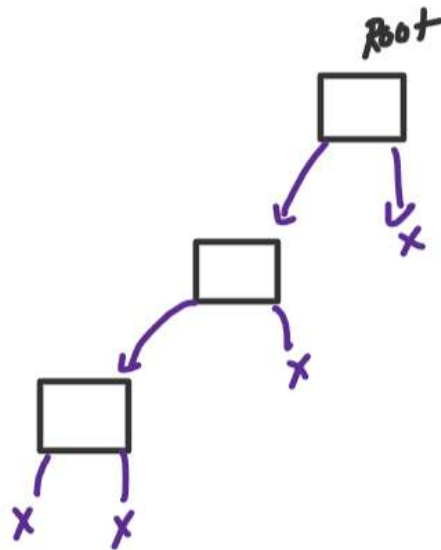


### 5. Skewed Binary Tree

A skewed binary tree is a pathological/degenerate tree in which the tree is either dominated by the left nodes or the right nodes. Thus,

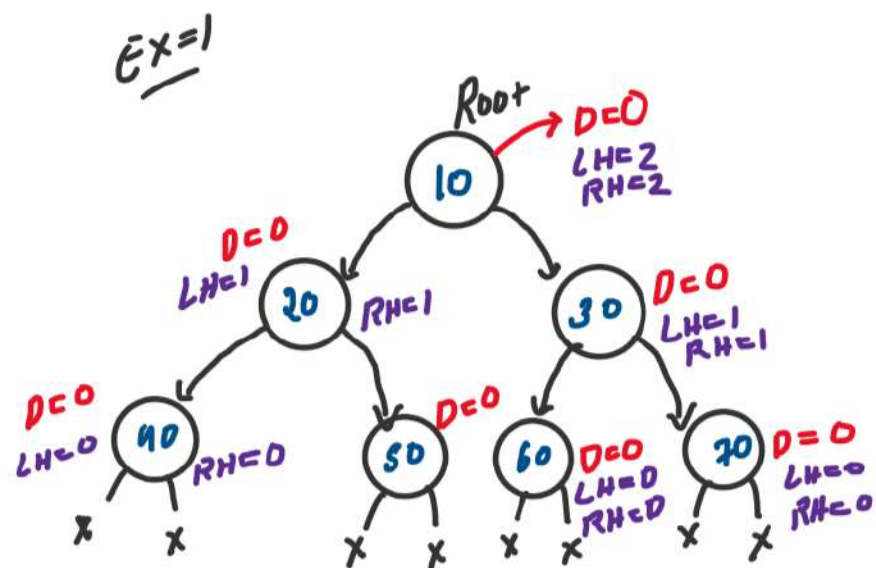
There are two types of skewed binary tree:

**Left-skewed binary tree** and **right-skewed binary tree**



## 6. Balanced Binary Tree

It is a type of binary tree in which the difference between the height of the Left and the right subtree for each node is either 0 or 1.



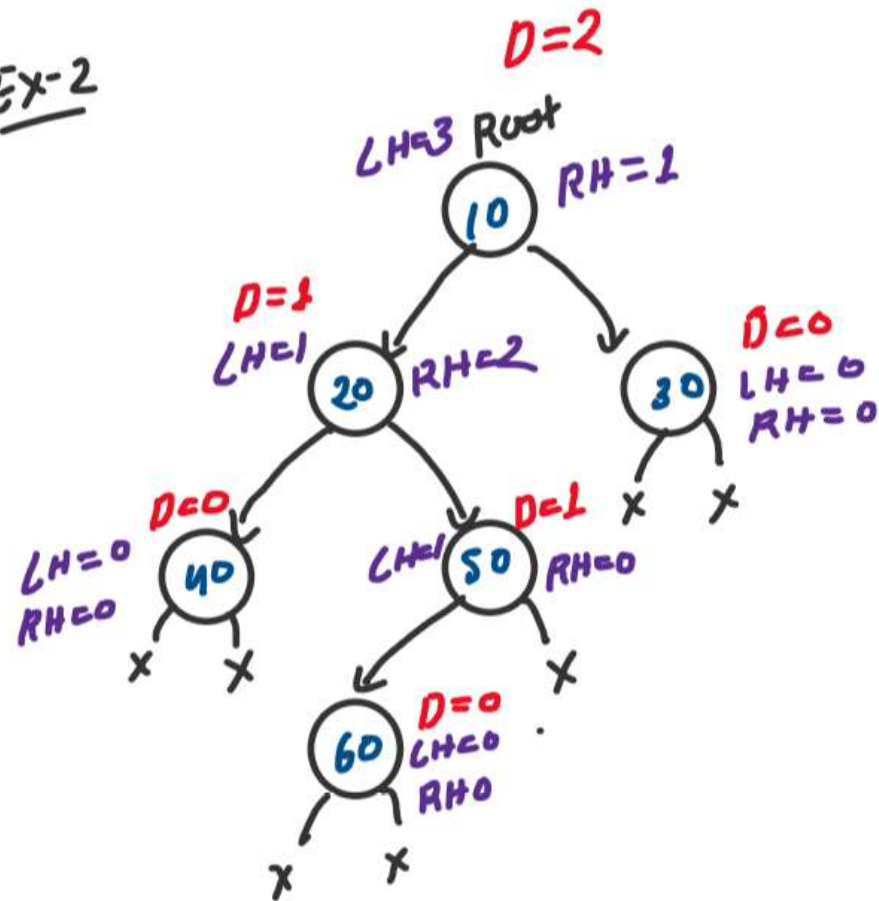
What is Balanced Binary Tree?

Left and Right sub tree ki height ka diff.<sup>n</sup> at most 1 hona chahiye.

$$LH - RH \in [-1, 1]$$

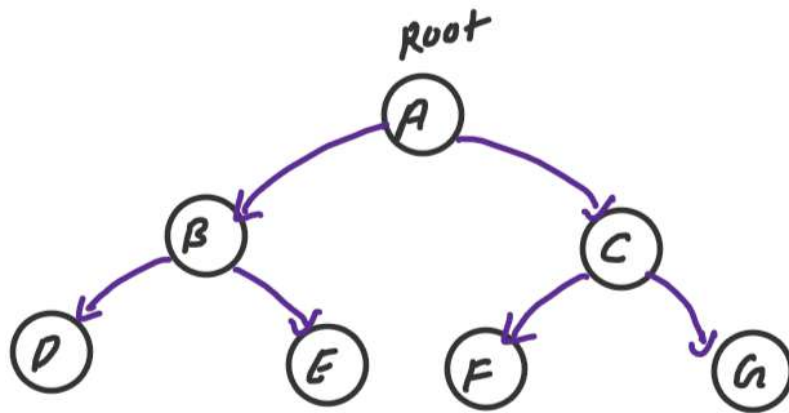


Ex-2



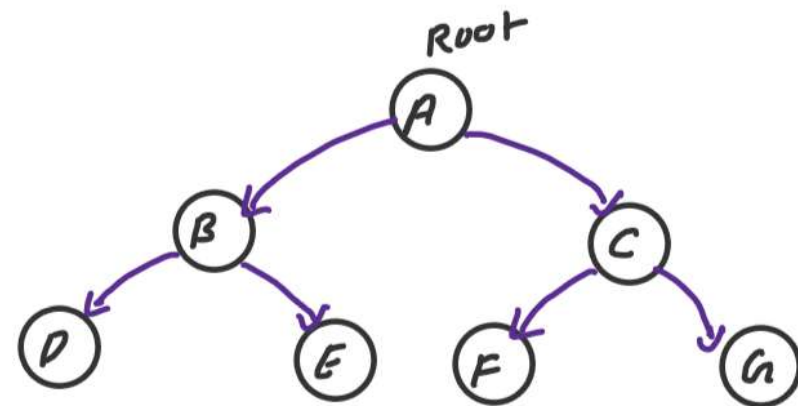
This is not a Balanced  
Binary TREE

## BFS & DFS Algorithms of Binary Tree



**BFS - Breath First Search**  
Level-Order Traversal

→ A B C D E F G



**DFS - Depth First Search**

Pre-Order Traversal (NLR)

A B D E C F G

In-Order Traversal (LNR)

D B E A F C G

Post-Order Traversal (LRN)

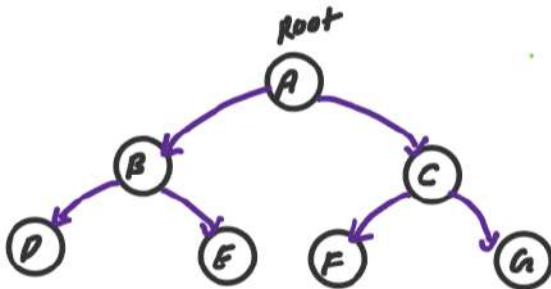
D E B F G C A

**DFS Traversals:**

1. Pre-order traversal (NLR)
2. In order traversal (LNR)
3. Post order traversal (LRN)

**BFS Traversals:**

4. Level order traversal

**Why do we care?**

There are many tree questions that can be solved using any of the above four traversals.

**Using DFS Traversal:**

- a.) Finding Size of Tree.
- b.) Finding Height of Tree.
- c.) Finding Max or Min element in a Tree.
- d.) Diameter of Binary Tree.
- e.) Print nodes at K distance.
- f.) Checking if a binary tree is subtree of another binary tree.
- g.) Ancestors of a given node.

**Using BFS Traversal:**

- a.) Maximum Width of Binary Tree.
- b.) Left View of Tree.
- c.) Connect Nodes at same Level.