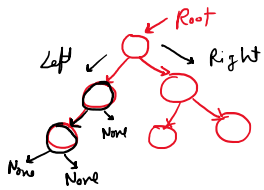
Binary tree

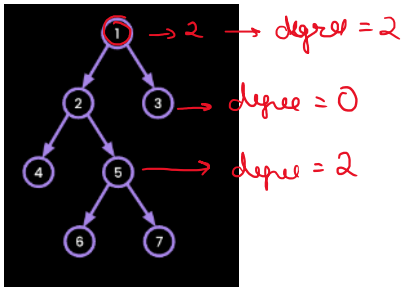


What is a Binary Tree?

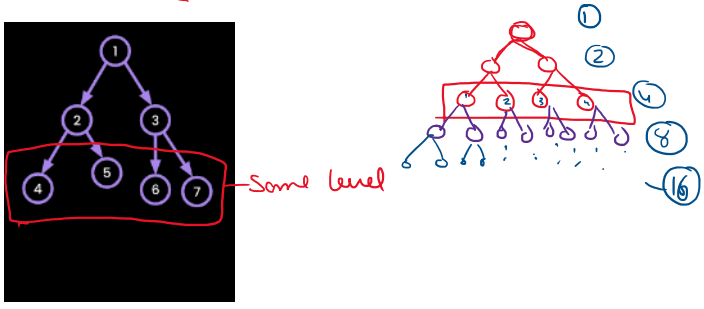
- A binary tree is a type of tree where each node can have at most 2 children.
- A node can either have 0(it is called leaf node), 1 or 2 child nodes.
- Every node in a binary tree has 2 pointers, i.e. left and right.
- These two pointers denote the Left and Right children of the node.
- In case, a node has no children, both the pointers point to null and in case a node has only 1 child, one of the pointers point to null respectively.

Types of Binary Trees

1. Full Binary Tree:
   - Every node other than the leaf nodes has exactly two children.
   - All nodes are either degree-0 (leaf) or degree-2.

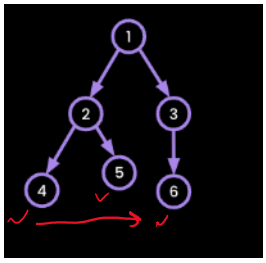


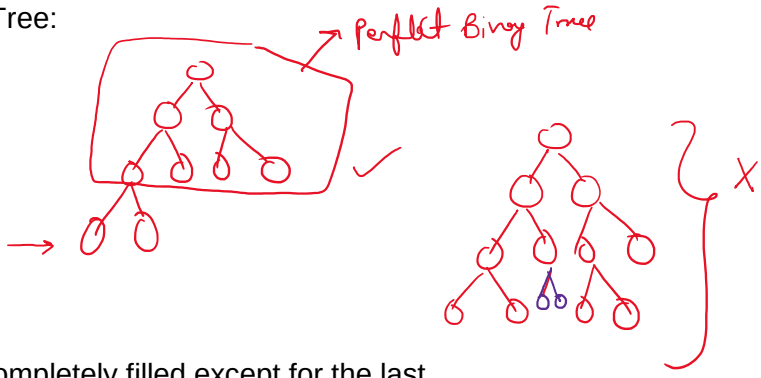
2. Perfect Binary Tree:



- A perfect binary tree is both full and complete.
- All interior nodes have two children.
- All leaves are at the same level.
- The number of nodes doubles at each level going down the tree.

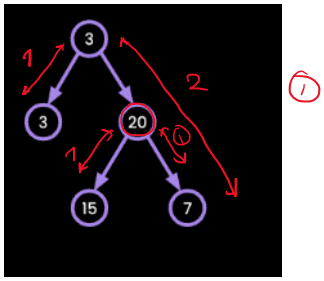
3. Complete Binary Tree:

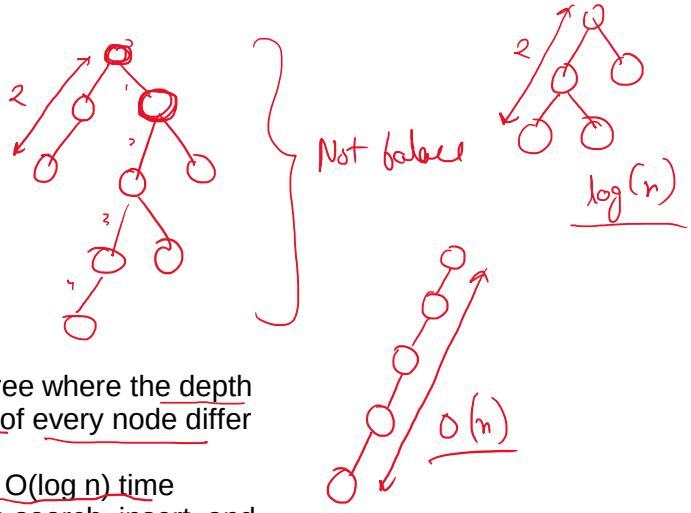




- All levels are completely filled except for the last level which is filled from left to right.
- Essentially, it's a tree that doesn't have any gaps in the sequence of nodes from the left.

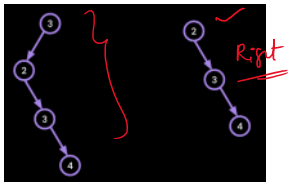
4. Balanced Binary Tree:





- A balanced binary tree is a tree where the depth of the left and right subtrees of every node differ by no more than 1.
- This property helps maintain O(log n) time complexity for operations like search, insert, and delete.

5. Degenerate and Skewed Binary Tree:



Degenerate Tree: Every parent node has only one associated child node. It essentially becomes a linked list.

Skewed Tree: A skewed binary tree is a special case of a degenerate tree where all nodes have only one child, either left or right.