# Types Of Binary Trees

| | |
|---|---|
| Created | @September 19, 2022 8:10 PM |
| Class | |
| Type | |
| Materials | |
| Reviewed | ☐ |

We have many different types of binary trees, few of them we are going to discuss

## Skewed Binary tree

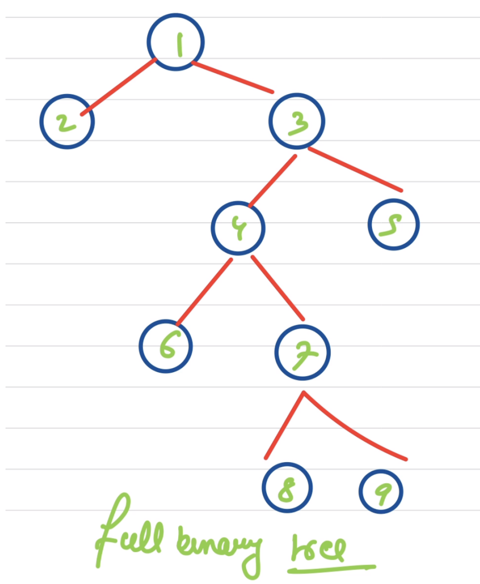
It is a special type of binary tree, where tree as a majority grows in one direction and gives an overall structural look like Linked List. Skewed binary trees are bent in one direction and having either left dominated nodes (i.e. only consecutive left child) or right dominated nodes(i.e. only consecutive right child).


## Degenrate Tree:

It is a special type of tree where every node has only one child except leaf. They might not be bended on any direction.

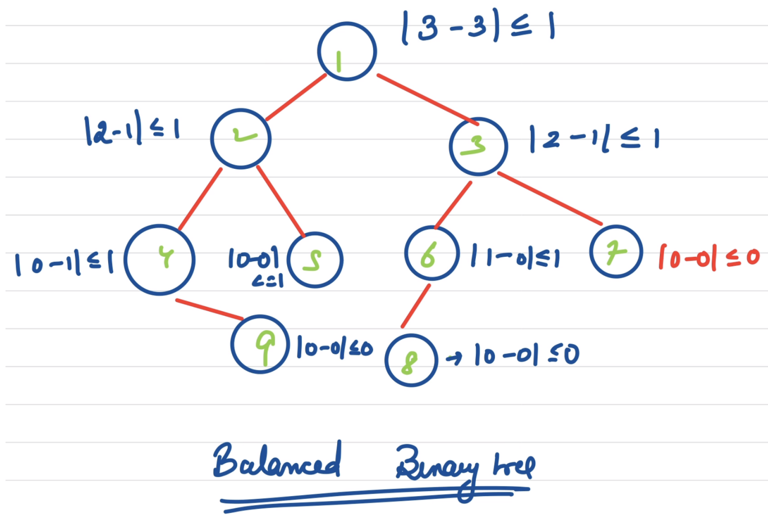
# Full binary tree

In a full binary tree, every node has either 2 children or NO children. Full binary tree has some application such as `Segment trees`. Every Segment tree is a full binary tree.




# Balanced Binary tree

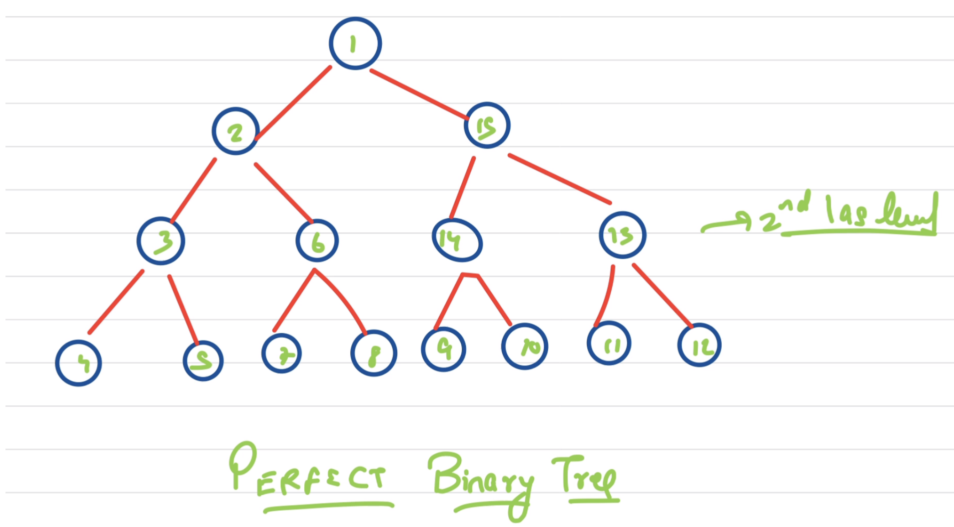
In a balanced binary tree the absolute difference between the height of left subtree and right subtree is at max 1, i.e. `|height_of_lst - height of_rst| ≤ 1` and this should be recursively true for all the subtrees of the given tree. Due to this, overall height of the tree becomes `O(logn)`. One of the applications of Balanced Binary tree is `AVL tree.`




# Binary search tree

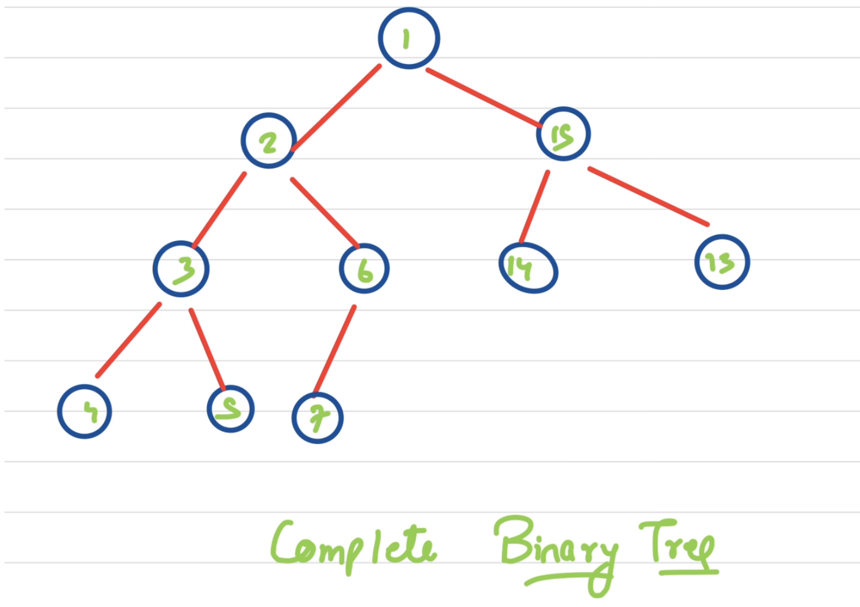
# Perfect Binary tree

In a perfect binary tree, all the nodes till the 2nd last level have 2 children. Height of the perfect binary tree is always `O(logn)`



# Complete Binary Tree

In a complete binary tree all the level except the last level are full, and the last level is filled from left to right without leaving any null pointers. Height of complete binary tree: `O(logn)`



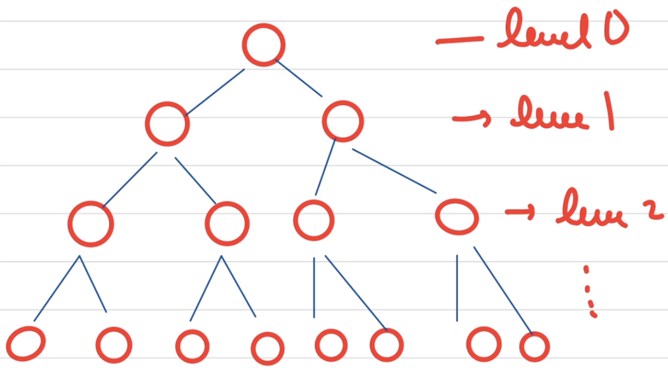

# Question:

How many leaf nodes are there in a perfect binary tree with total N nodes.

## Answer

If the perfect binary tree has N nodes, then there are (N+1)/2 nodes as leaf, which is approx half of the total no of nodes.

Total nodes $= n$




Sum of nodes at each level should be equal to total no. of nodes

Total nodes on $k^{th}$ level is $2^k$

this term shows the count of last level nodes.

$$\underbrace{2^0 + 2^1 + 2^2 + 2^3 \cdots\cdots\cdots 2^{h-1}} + 2^h = n$$

$$\Rightarrow \frac{1 \times (2^h - 1)}{2-1} + 2^h \Rightarrow 2^h - 1 + 2^h = n \Rightarrow 2 \times 2^h = n+1$$

$$\Rightarrow \boxed{2^h = (n+1)/2}$$

→ Total leafs in Perfect B.T