



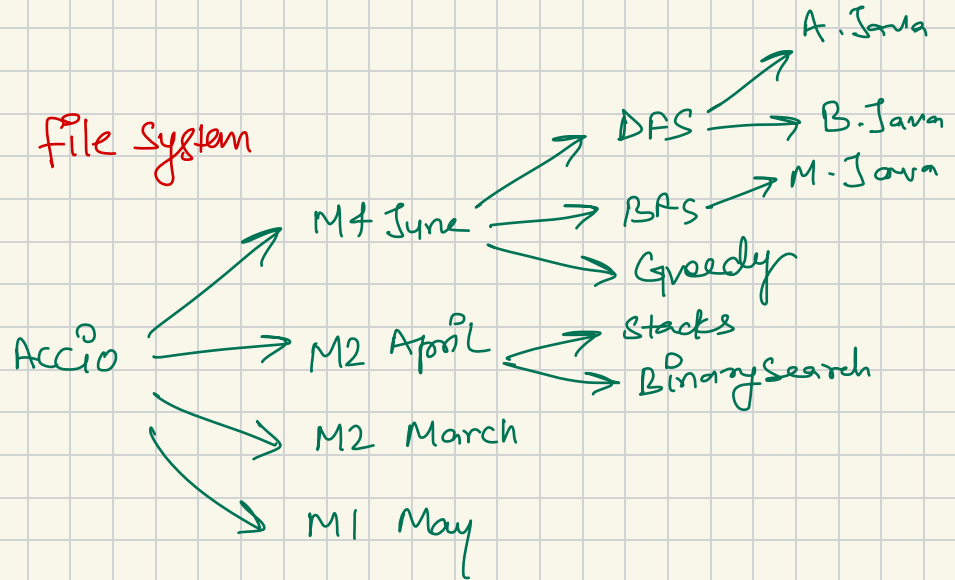
Binary Trees

↳ Non Linear Data Structure

Data

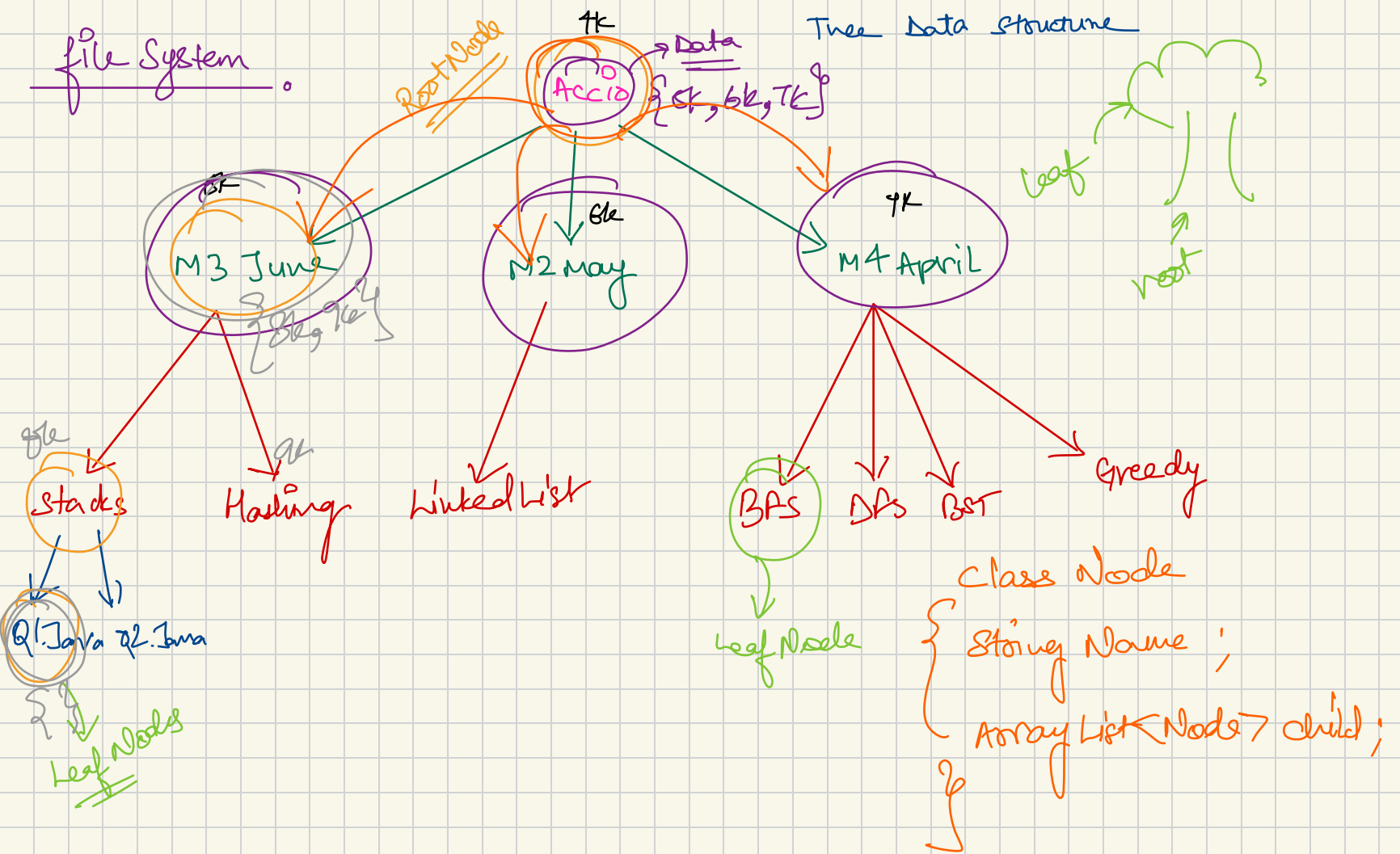
- family tree
- file system
- org. chart

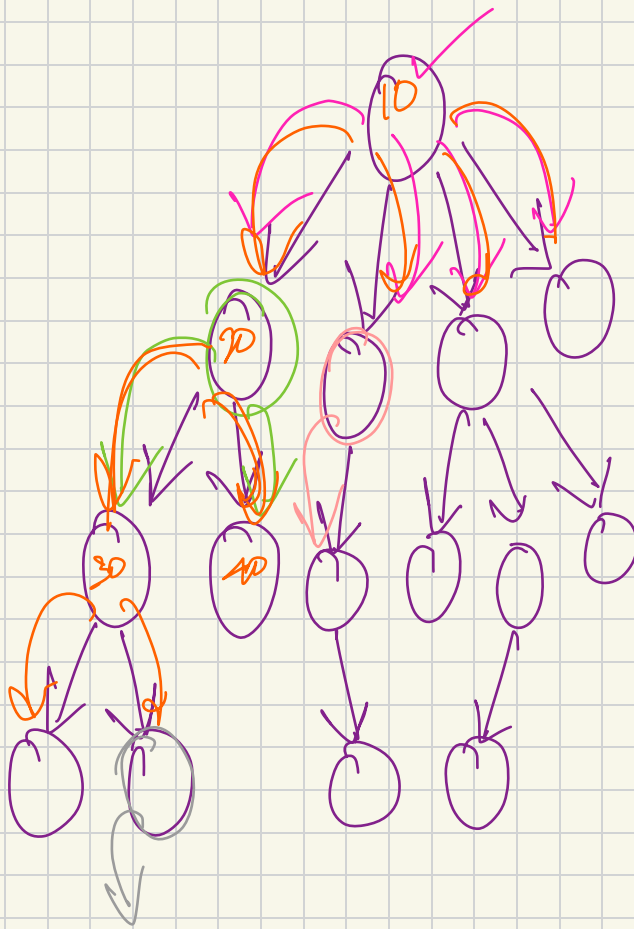
file System



file System .

Tree Data Structure



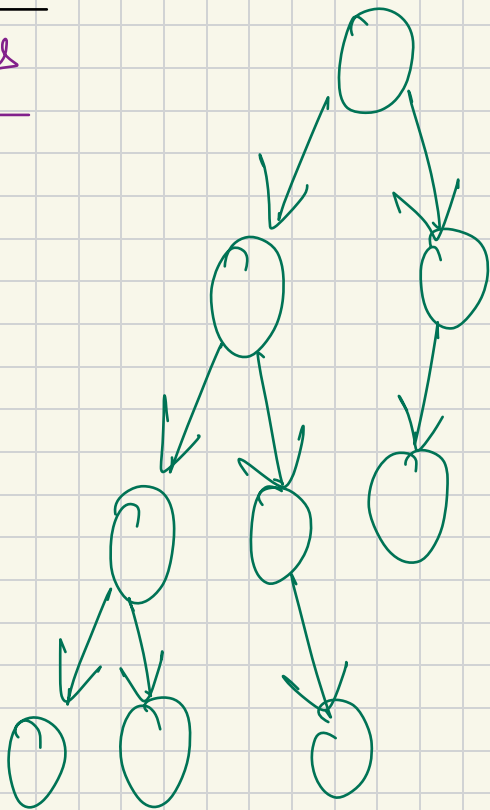


Generic Tree

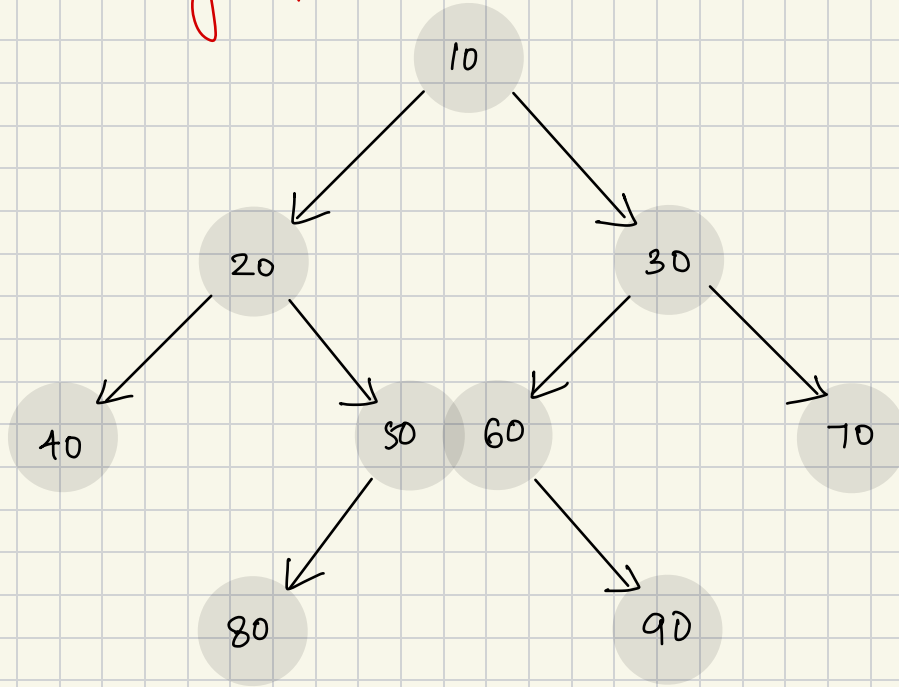
No. of child
 $= \{0 \text{ to } N\}$

- 0, 1 or 2 children

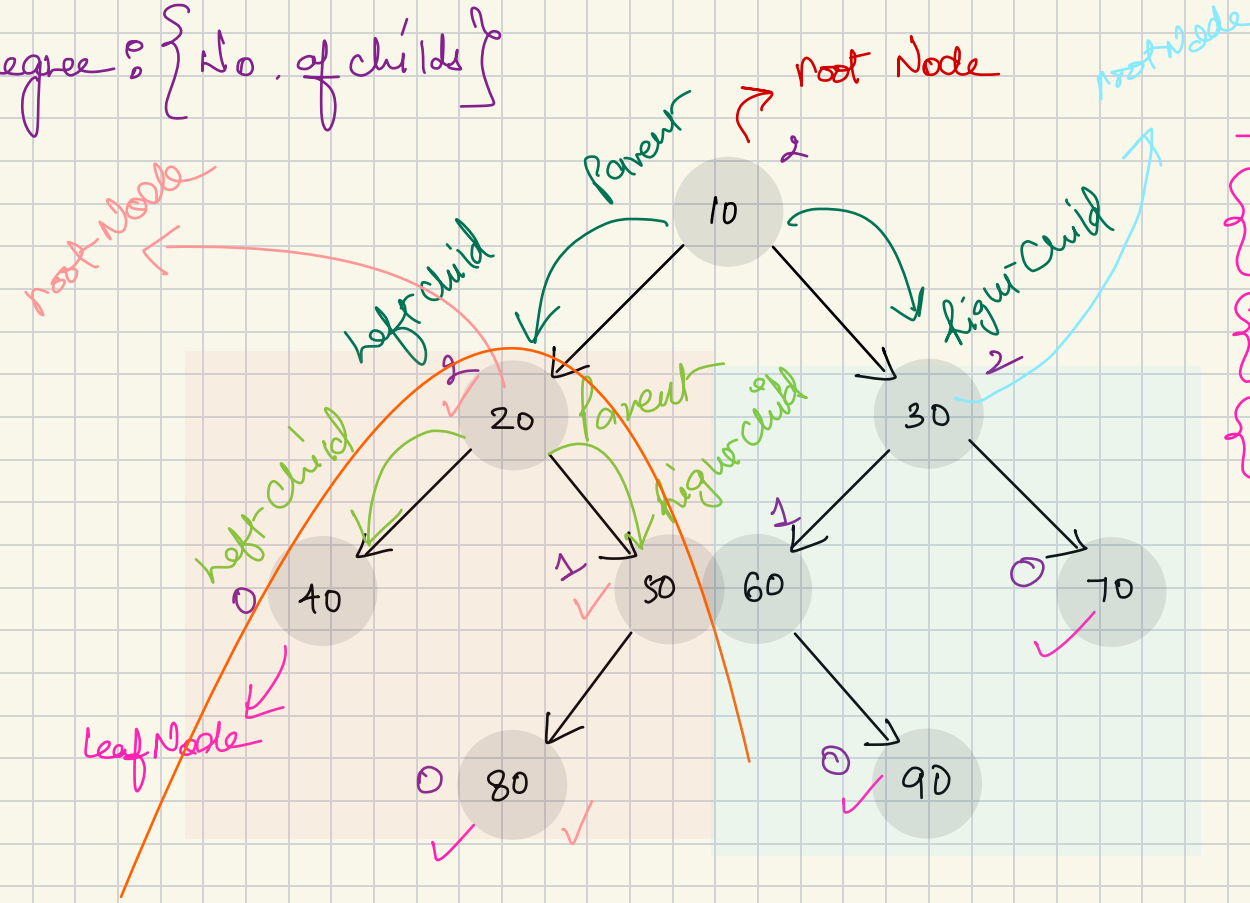
Binary Tree



For Teaching Purpose



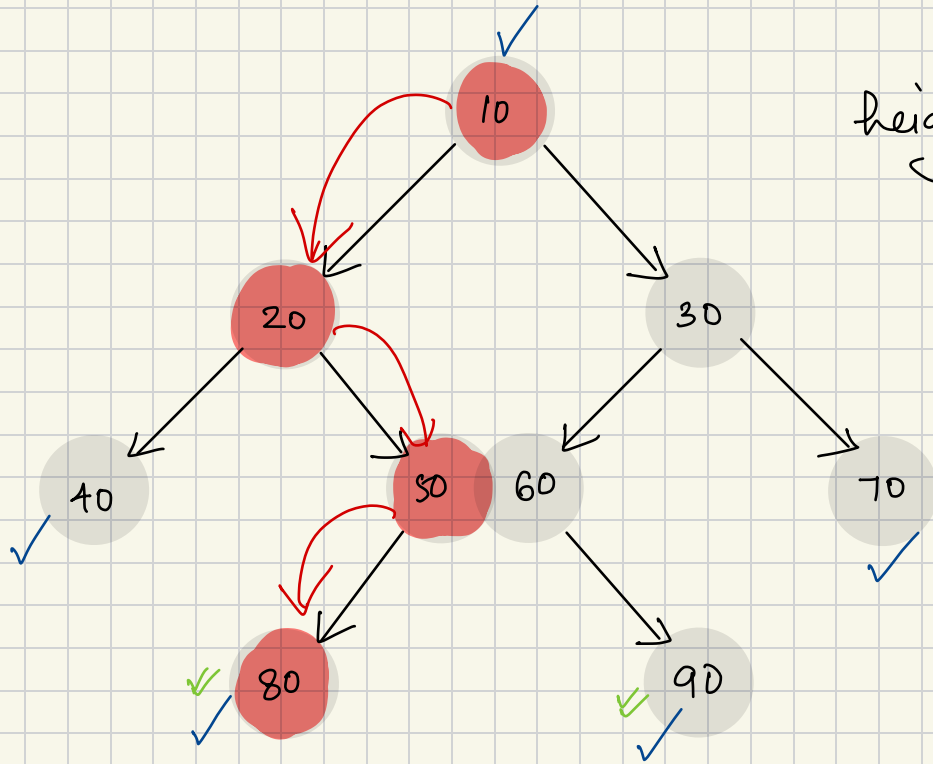
degree: { No. of childs }



Siblings
{ 20, 30 }
{ 40, 50 }
{ 60, 70 }

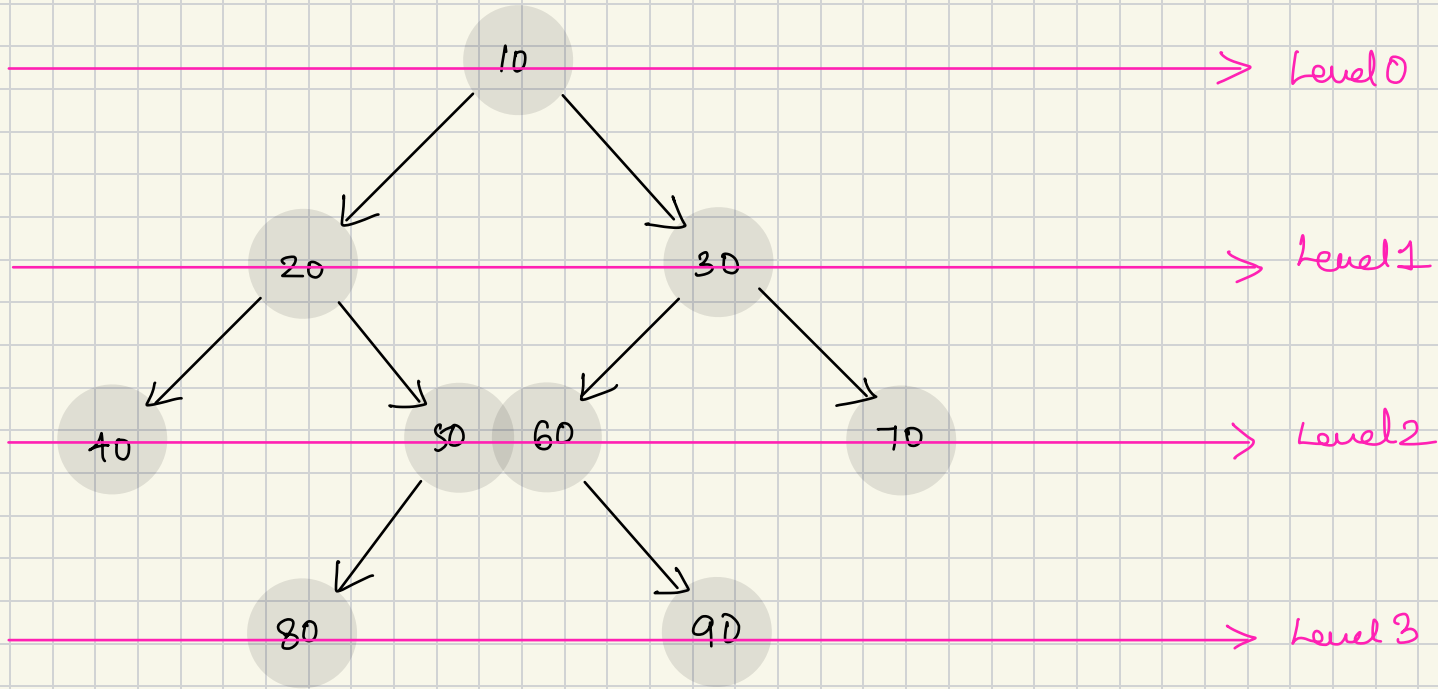
Height of a Binary Tree

o dist. b/w root Node and deepest leaf Node

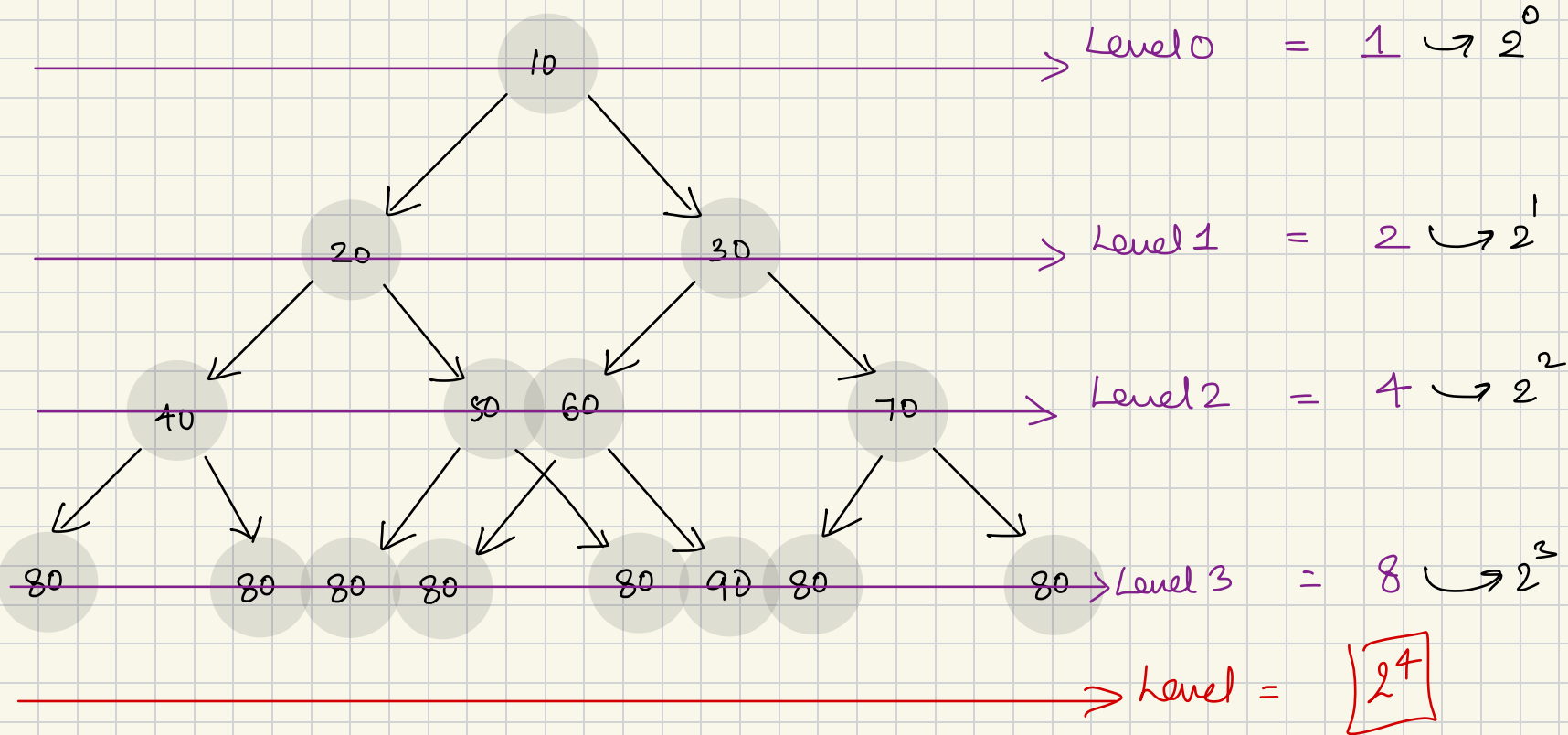


height = 4 { in terms of Node }

Levels in a Binary Tree

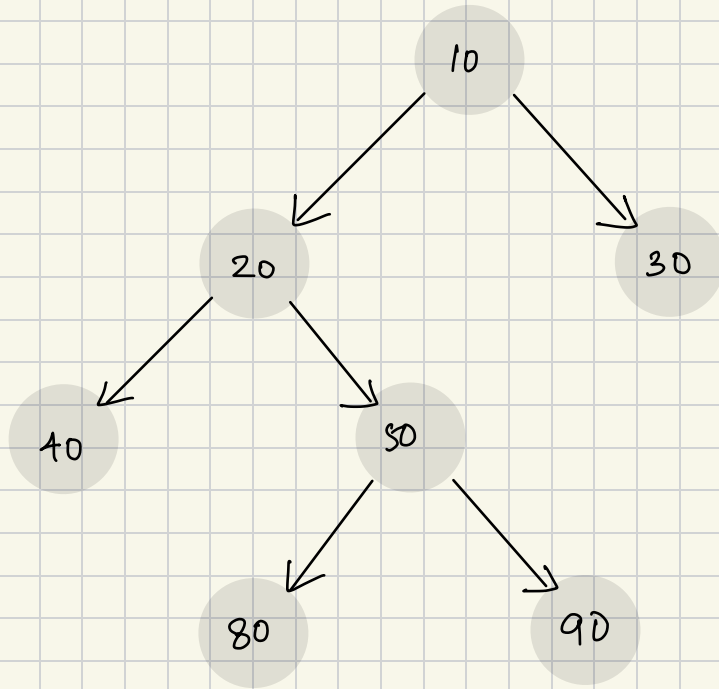


Perfect Binary Tree } No. of Nodes in each level (L) = 2^L }



Full Binary Tree

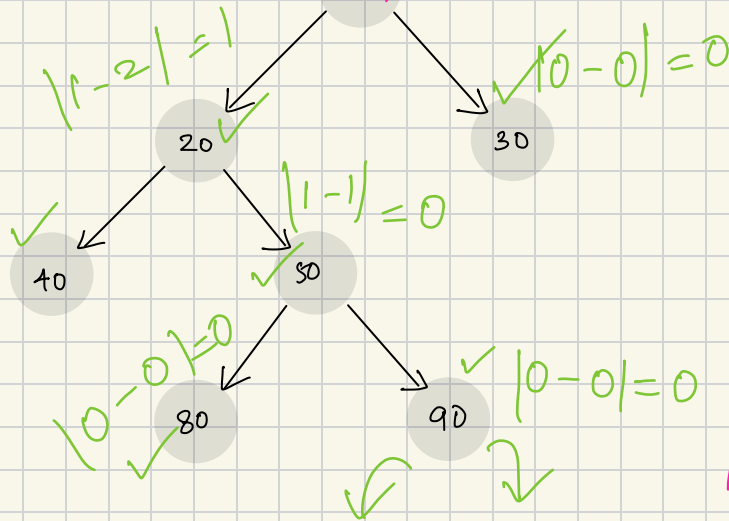
↳ Each Node can either have 0 or 2 child



Balanced Binary Tree

o where each node is balanced.

$$|3-1| = 2 \neq 1$$



Balanced Node

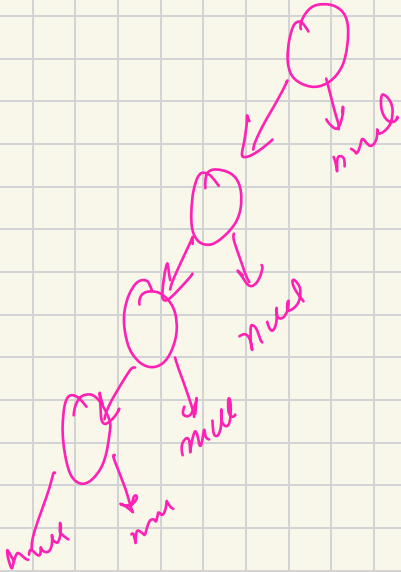
$$|h_{LST} - h_{RST}| \leq 1$$

NOT Balanced

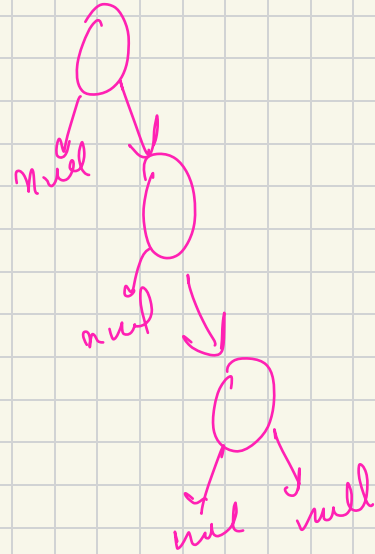
Skew tree

1. Left skew

left child or no child



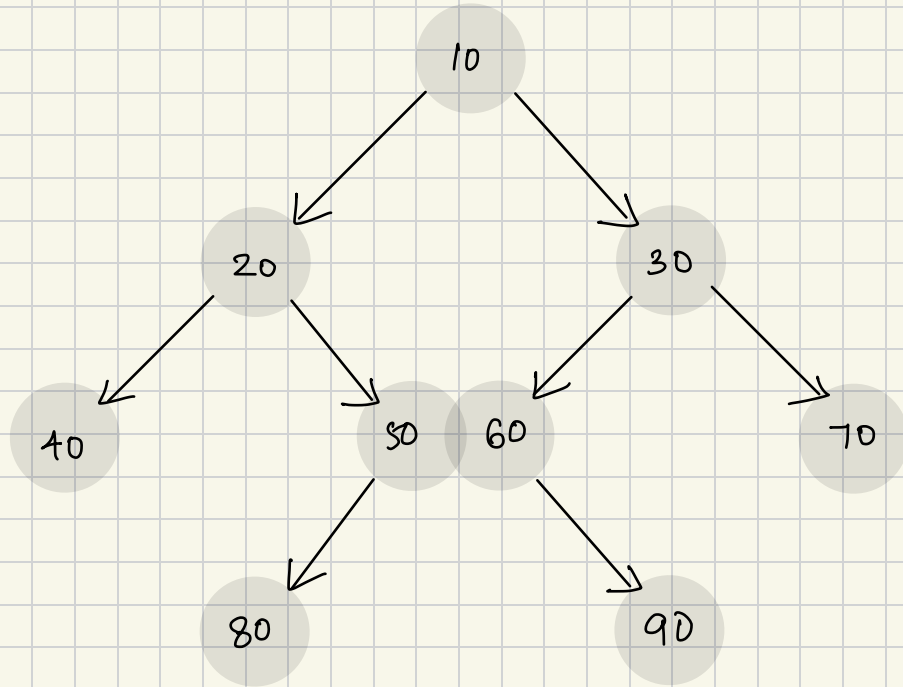
2. Right skew



Traversal over a tree

• Pre Order Traversal

OP { 10, 20, 40, 50, 80, 30, 60, 90, 70 }

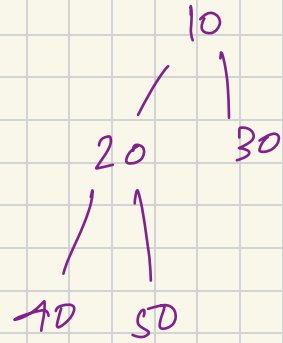


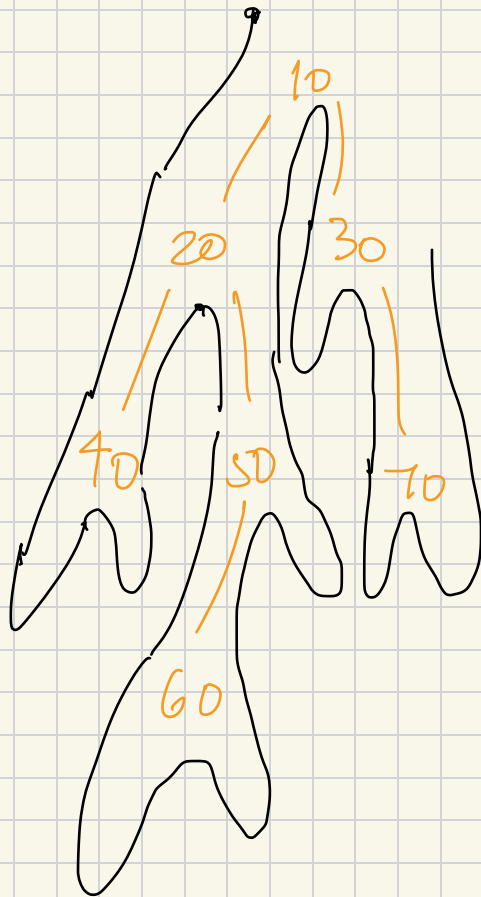
- Print root Node
- Print LST {preorder}
- Print RST {preorder}

fact:
prints preorder
of the BT
starting from
root

```
void fun (Node root)
{
    ① if (root == null) return;
    ② print (root.data);
    ③ fun (root.left);
    ④ fun (root.right);
}
```

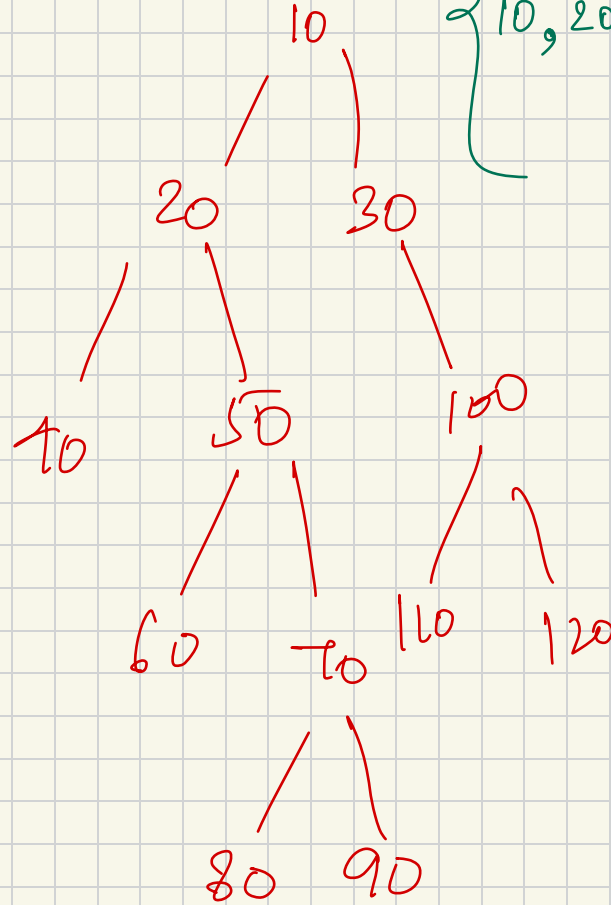
10, 20, 40, 50, 30





Euler Path

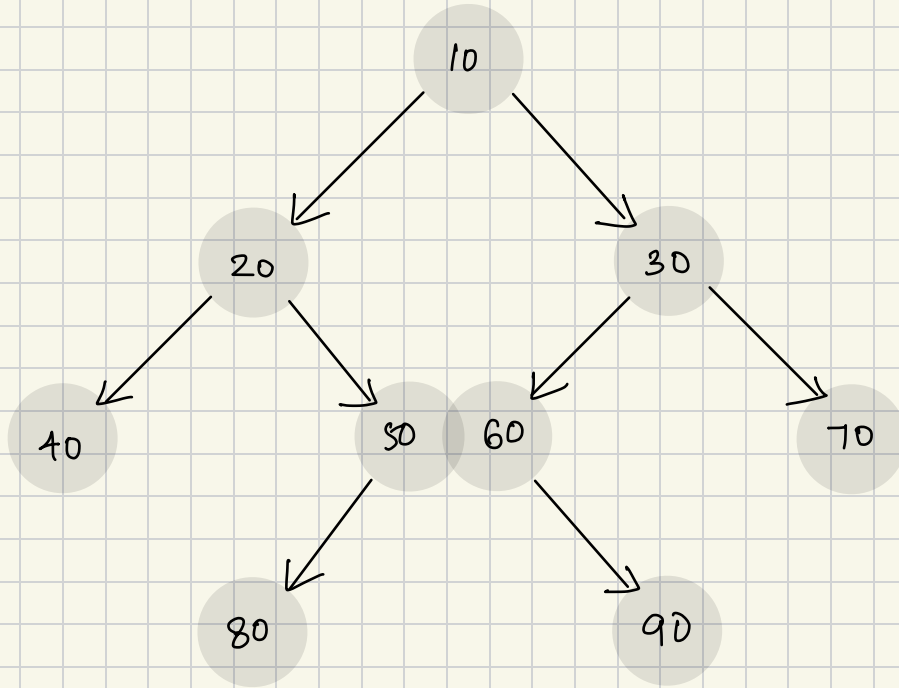
$\{10, 20, 40, 50, 60, 30, 70\}$



$\{10, 20, 40, 50, 60, 70, 80, 90, 100, 110, 120\}$

In Order Traversal

40, 20, 80, 50, 10, 60, 90, 30, 70



- Inorder BST
- print root Node
- Inorder BST



10, 20, 60, 50, 70, 10, 30, 90, 80, 100