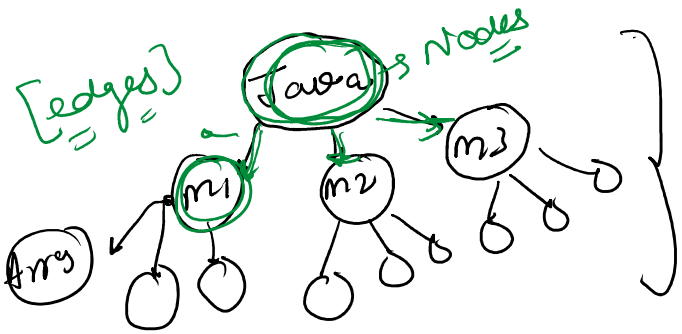
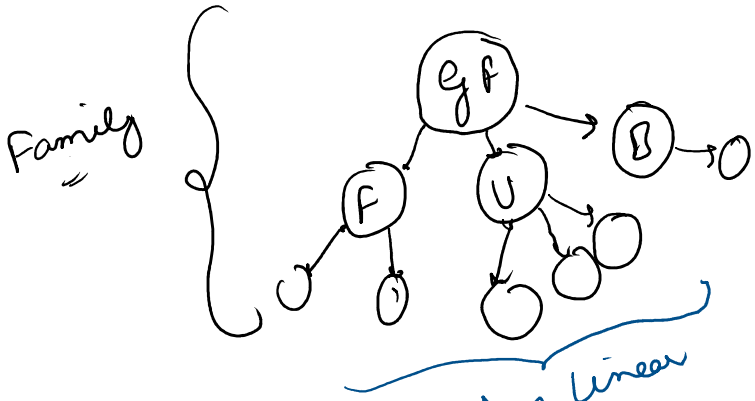
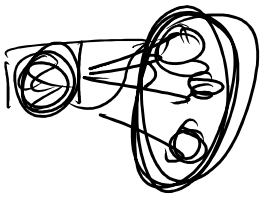
3.16 - Binary Trees - 1

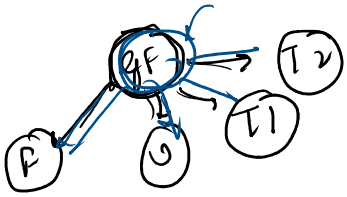
Thursday, August 14, 2025 9:02 PM

Trees





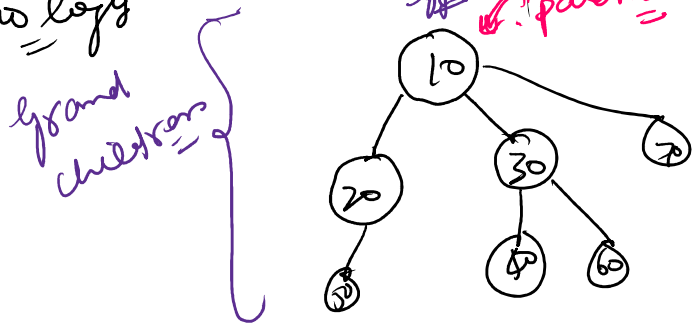





```
class Node {
    int data;
    ArrayList<Node> kids;
}
```

child nodes

Terminology

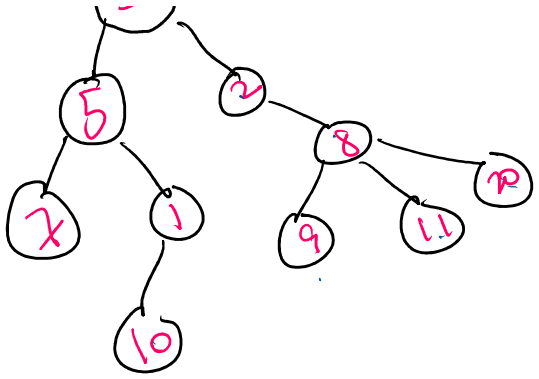


20, 30, 70
child nodes

⇒ Generic / N-ary tree: A tree where a node can have any no / N number of children

⇒

10 Nodes
9 edges



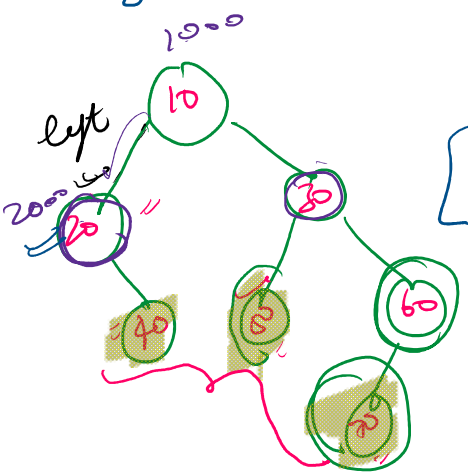
9 edges

→ A tree with (N) nodes has (N-1) edges.

Binary Tree

[atmost 2 children] → 0, 1, 2

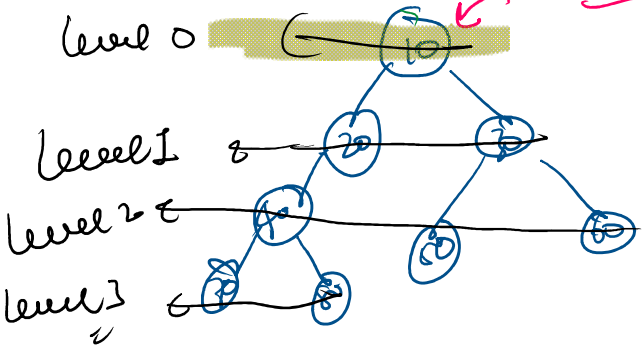
```
class Node {
    int data;
    Node left;
    Node right;
}
```



Binary Tree

[Leaf Nodes → Nodes with 0 children]

Levels in a Tree

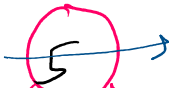


root Node
origin of the tree

level == Generation

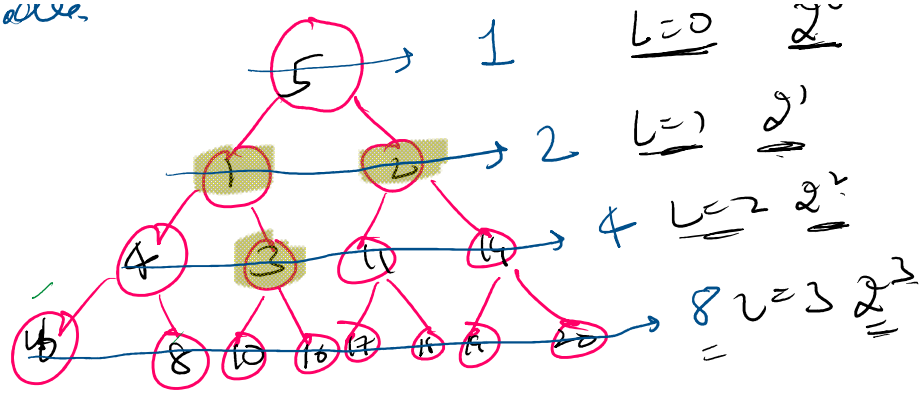
Perfect Binary Tree

→ where every Node has exactly 2 children except leaf Nodes.



→ 1 L=0 2^0

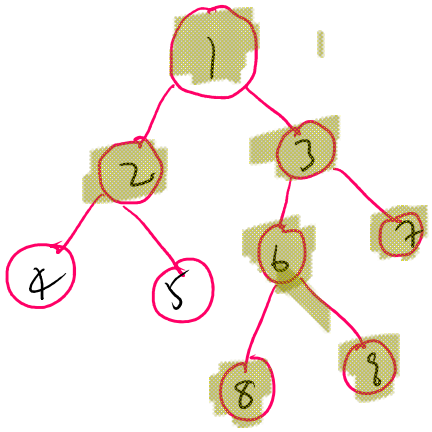
Nodes



$K^{th} \Rightarrow 2^K$

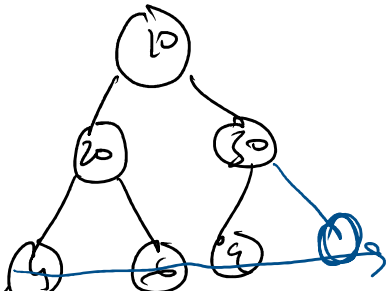
Full Binary Tree

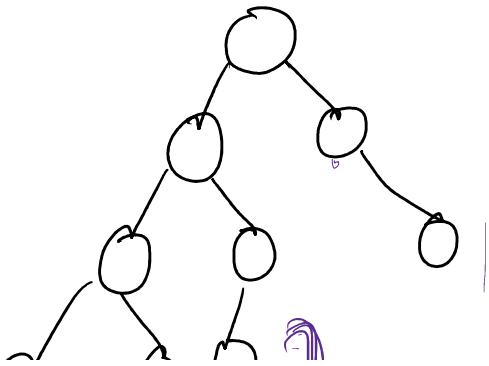
⇒ where every node will have either 0 or 2 child.

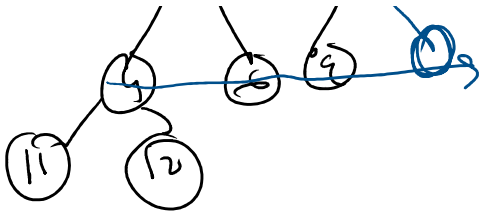


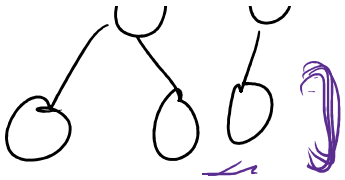
Complete Binary Tree

⇒ where every level will have the max no of nodes except the last level. All the nodes on last level will be as left as possible



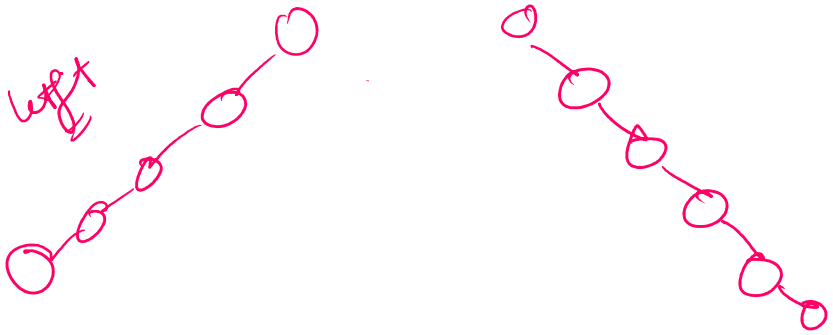






Skewed Trees

→ where every node is in same direction



Height of a Tree

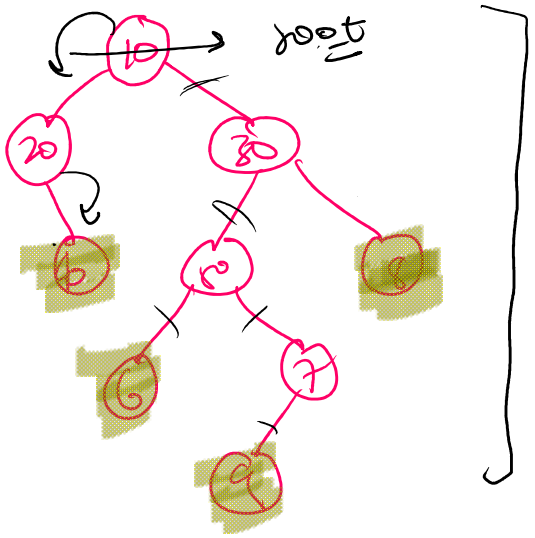
→ Distance b/w the root node and the deepest leaf node in terms of edges.

root → 10 } 0

60 } 3

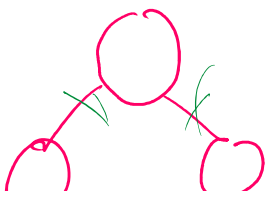
9 } → 4

8 } → 2



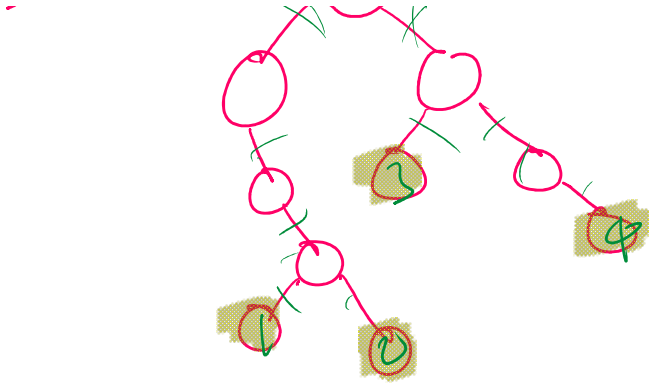
→ 4

→

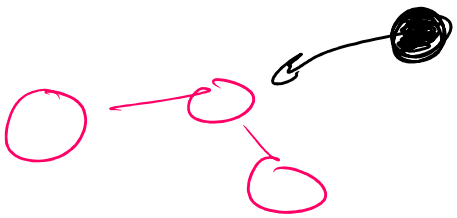


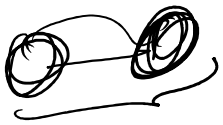
In terms of

OFN Page 4



In tree
↳ is connected network



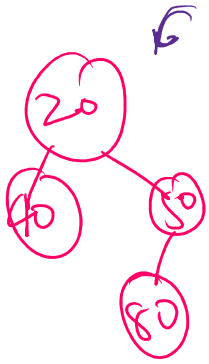


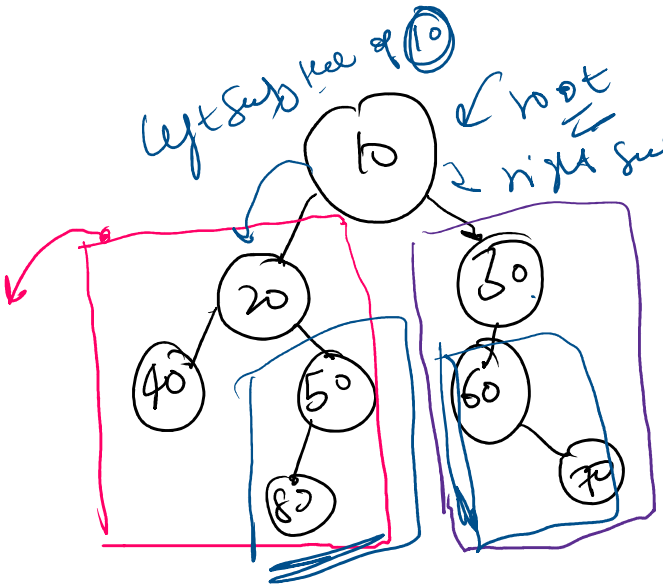
Balanced Tree

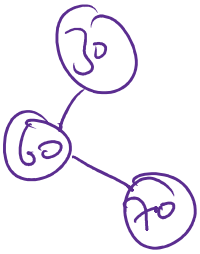
→ where every node is balanced

| Height of L.S.T − Height of R.S.T. | ≤ 1

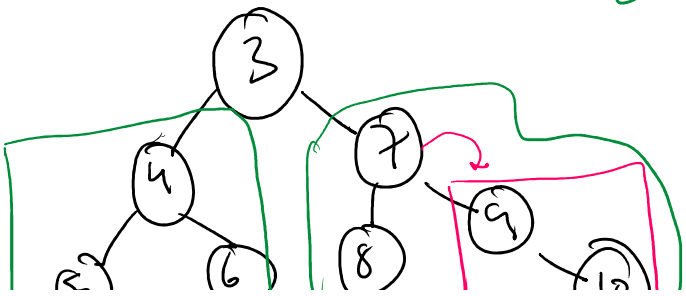
→ SubTree

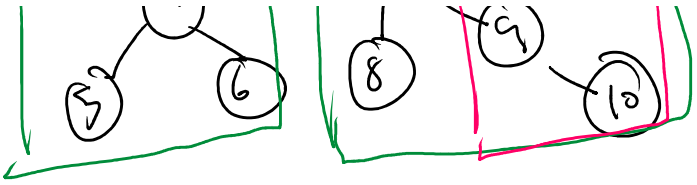


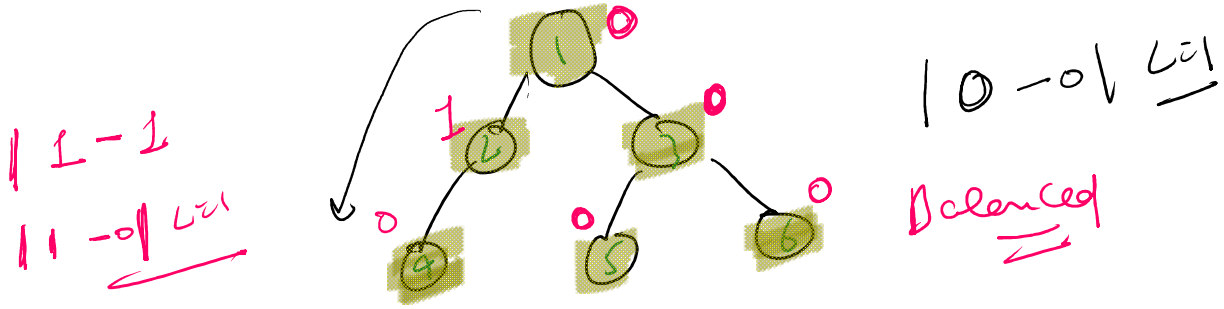


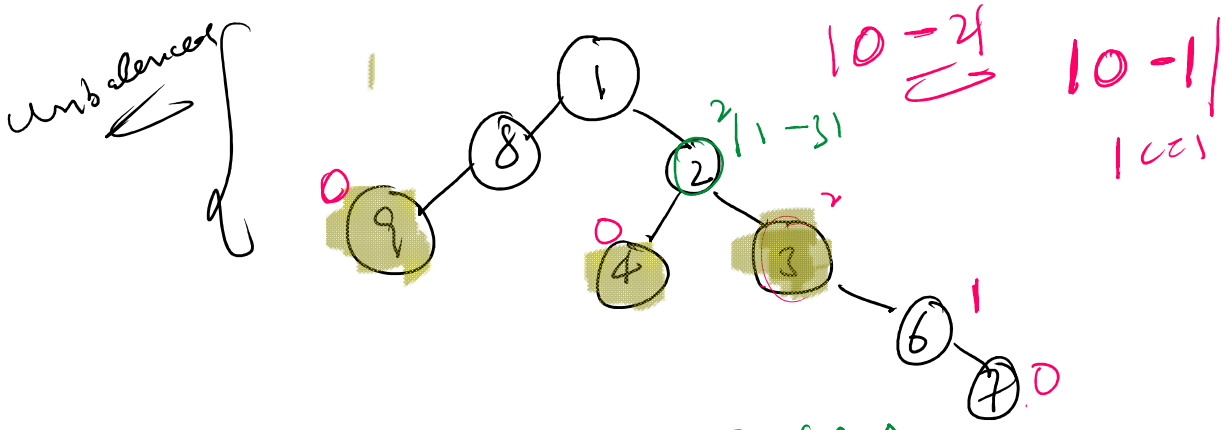


LST(3) → 4









→ Subarray sum divisible By K
→ Integer to Roman
→ Distinct elemt in a window of size k