(String, Hashing & Recursion)

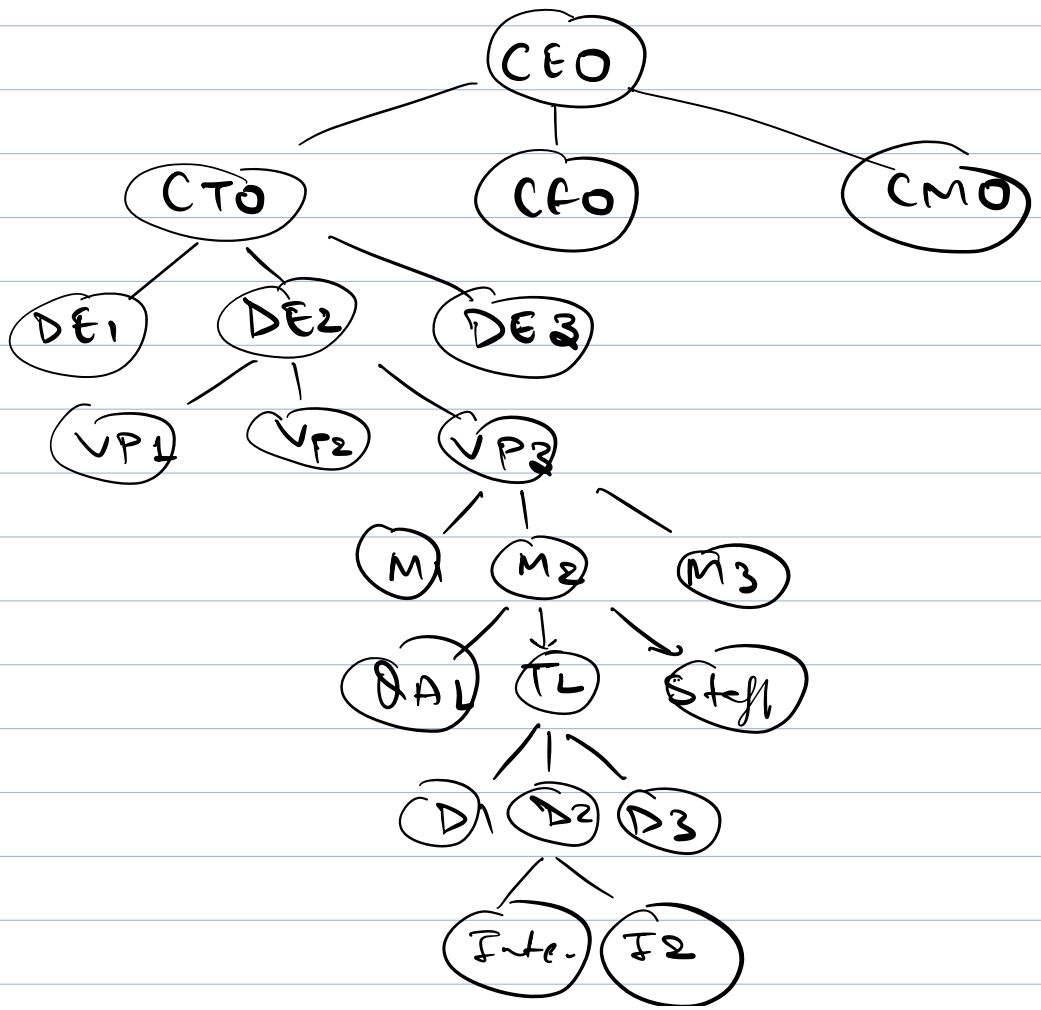
Friday ⇒ { Contest 3 (9:00PM to 10:30PM)
Discussion (10:30 PM to 11:30 PM)

Sat & Sunday (Reattempt)

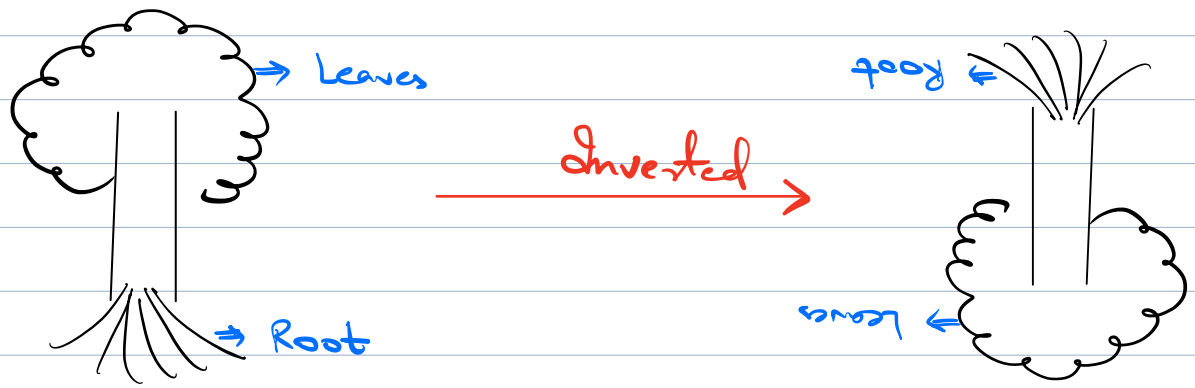
Thursday ⇒ 9:00PM (Optional PS session)

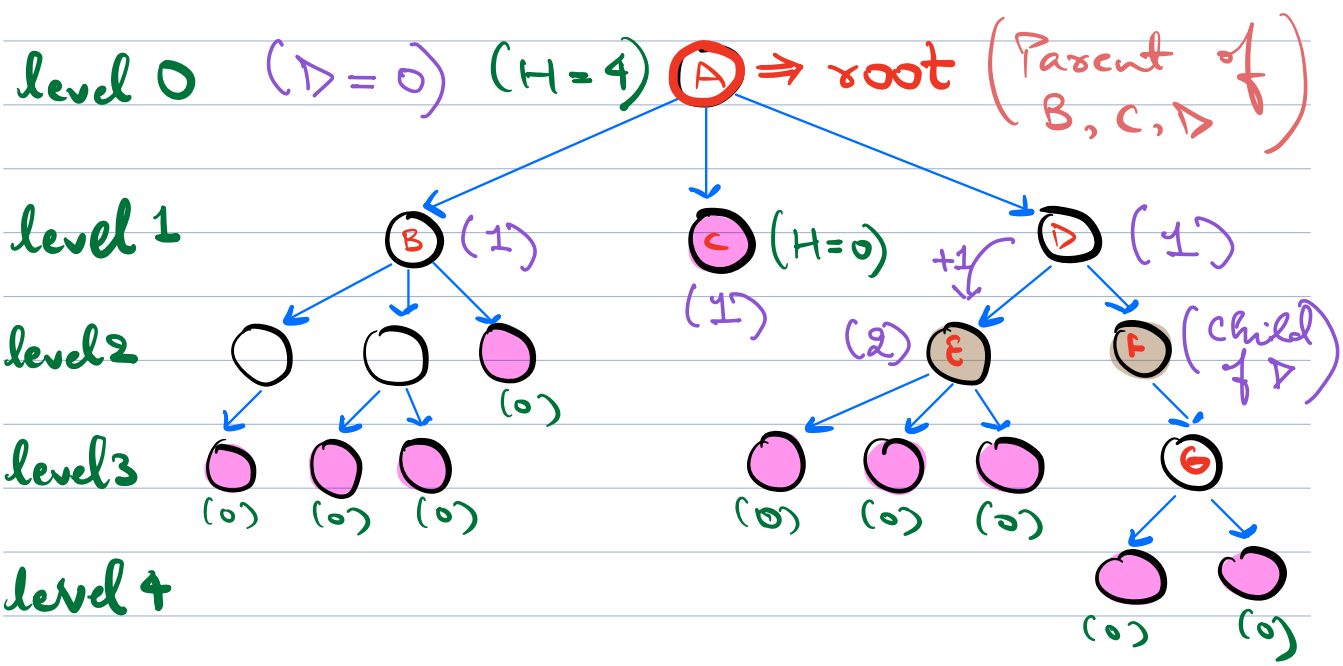
Advanced DSA ⇒ 22nd May

---



# Tree




Max no. of parent a node can have = 1

Max no. of childra a node can have in a tree = N - 1

with N nodes

Leaf Node : Node with no children.
(10 leaf nodes)

Sibling : Nodes with same parent

Ancestor / Descendent : F, D, A ⇒ Ancestors of G

Height (Node) : Distance to the farthest descendent leaf node

Distance : No. of edges b/w 2 nodes.

Height (leaf) = 0

Height (tree) = Height (root node)

Depth (Node) : Distance from root Node.

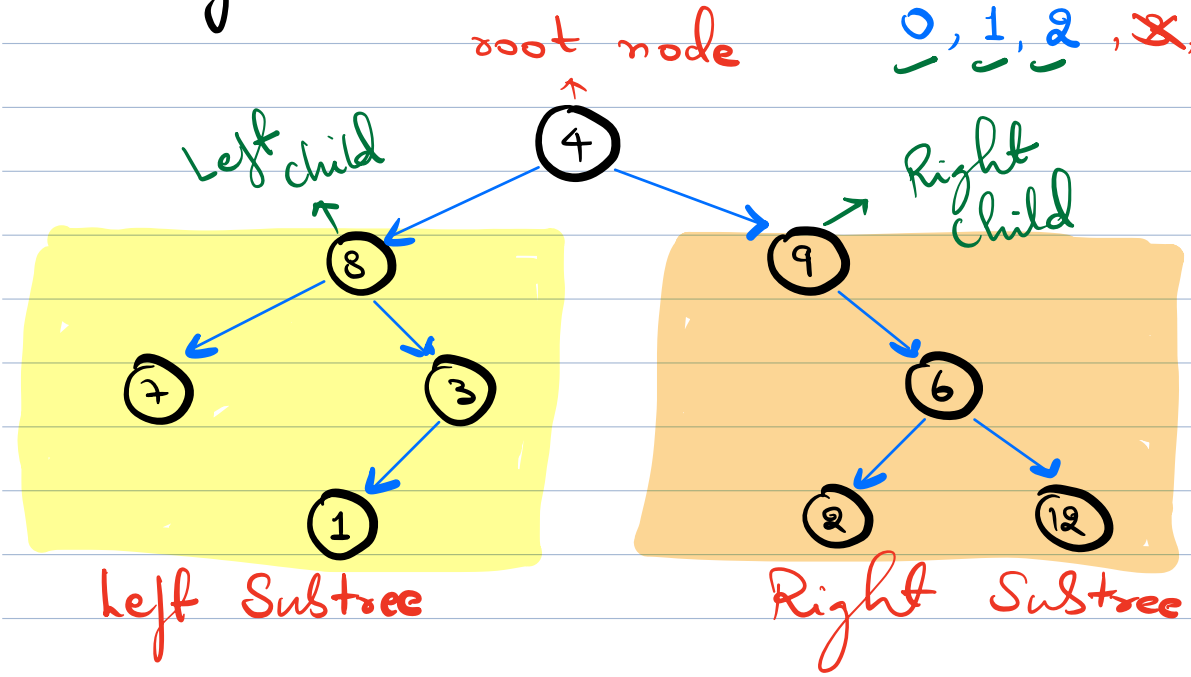
Depth (root) = 0

if Depth of a node = d
then depth of its children = d + 1

# Binary Tree

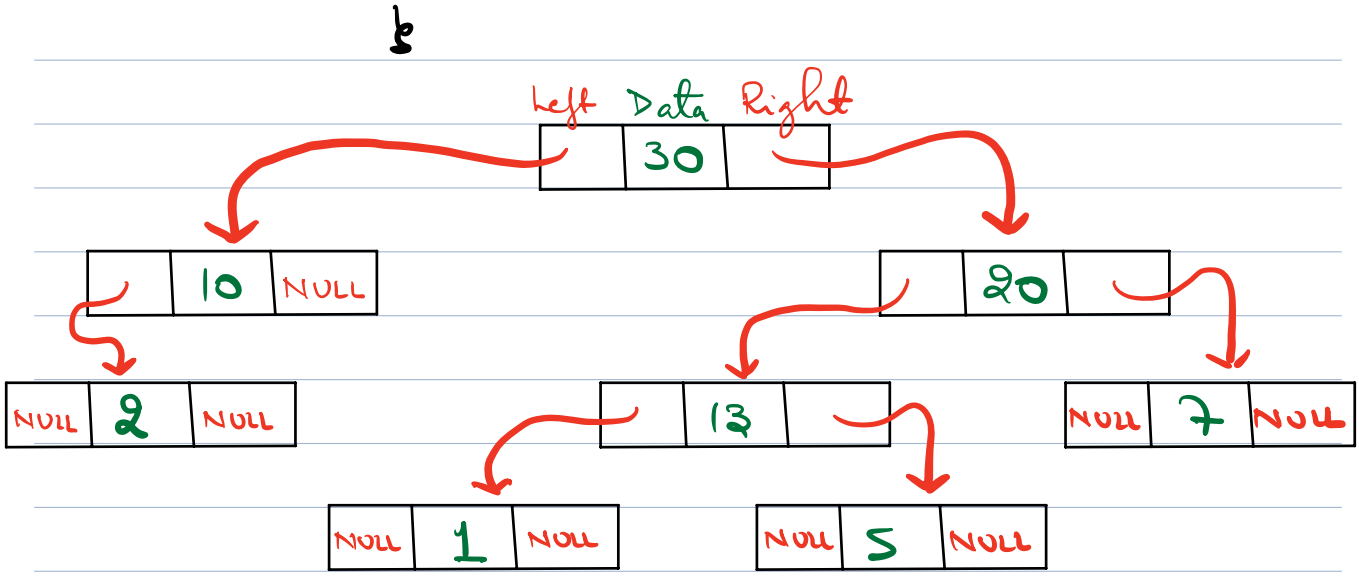
→ Every node can have atmax 2 children

0, 1, 2, ~~3~~, ~~4~~ ....




→ the given tree is rooted on the Node 4

```
class Node {
    int data;
    Node left;
    Node right;
    Node (data) {
        this.data = data;
        left = right = NULL
    }
```




```
if (node.left == NULL && node.right == NULL) {
        Leaf Node
}
```

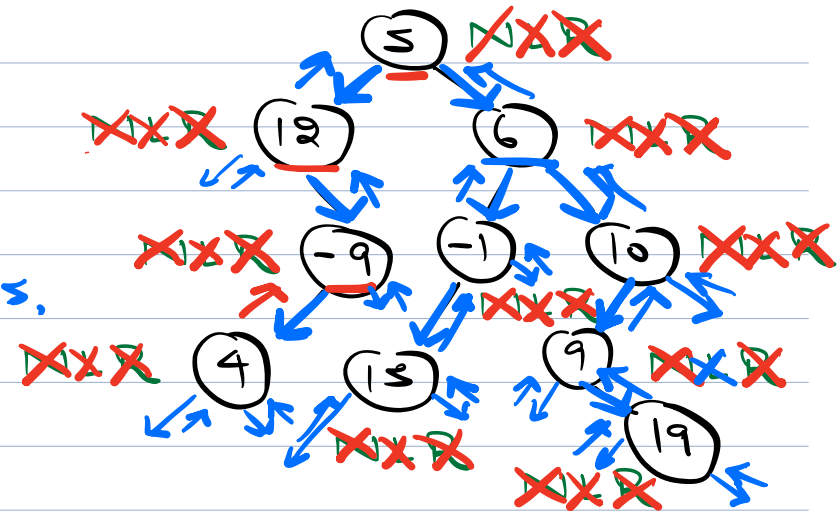
---

# Tree Traversal

- Pre Order
- In Order
- Post Order

Level Vertical (Adv. DSA)

1) Pre Order (Node, left, right)

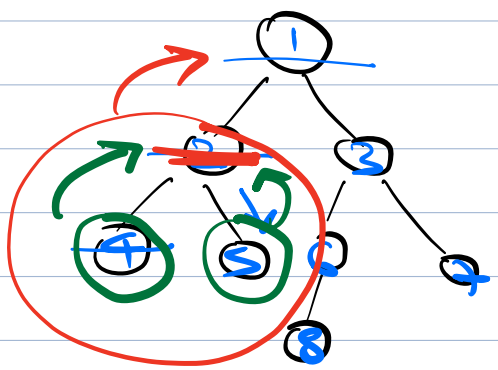


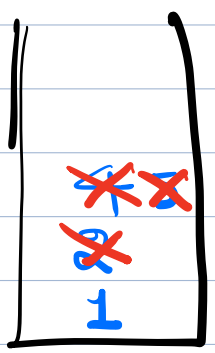

5, 12, -9, 4, 6, -1, 15, 10, 9, 19

# Code

```
void preOrder (Node root) {
    if (root == NULL) {
        return;
    }                                  > Base Case
1   print (root.data);
2   preOrder (root.left);
3   preOrder (root.right);
}
```

T.C. = O(N)
S.C. = O(Height)



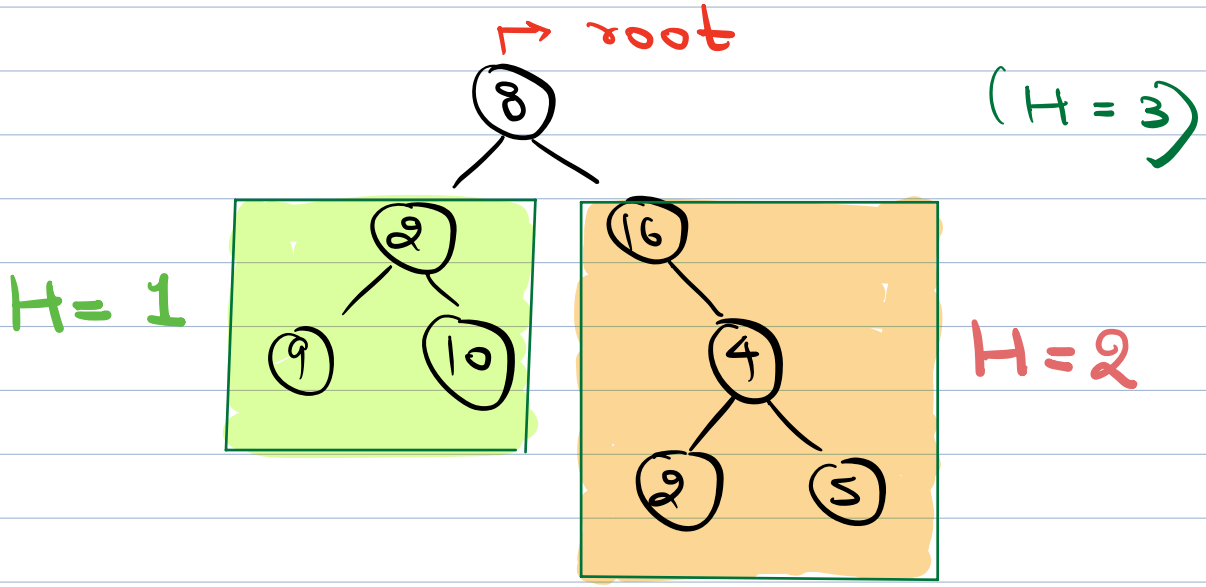
PRE: N, L, R ⇒ 1, 2, 3

IN: L, N, R ⇒ 2, 1, 3

POST: L, R, N ⇒ 2, 3, 1

---

Given the root Node of a BT, return the Height of tree.

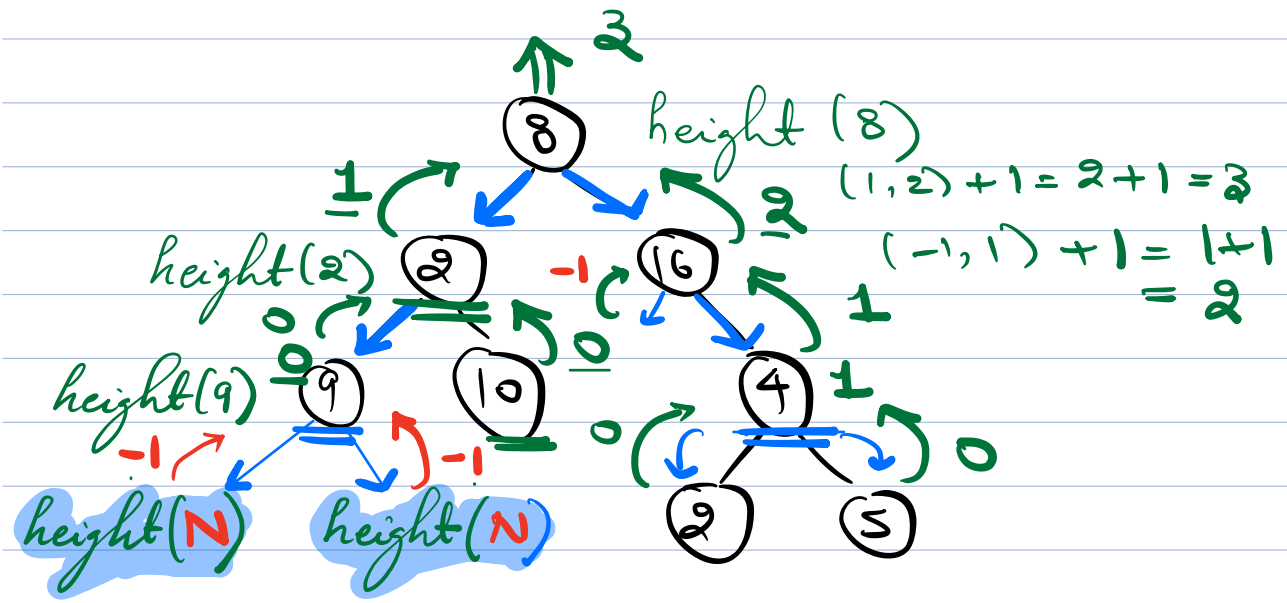



$$Height(root) = \max\left(Height_{LST}, Height_{RST}\right) + 1$$

Code

```
int height (Node root) {
    if (root == NULL) {
        return -1;
    }
    int l = height (root.left);
    int r = height (root.right);

    return max (l, r) + 1;
}
```




$$\max(x, x) + 1 = x + 1 = 0$$

$$-1 + 1 = 0$$

$$x = -1$$