Ayush Sharma ⇒ Scaler ⇒ Led DSA

⇓

2019 ⇒ CSE, IIT Patna

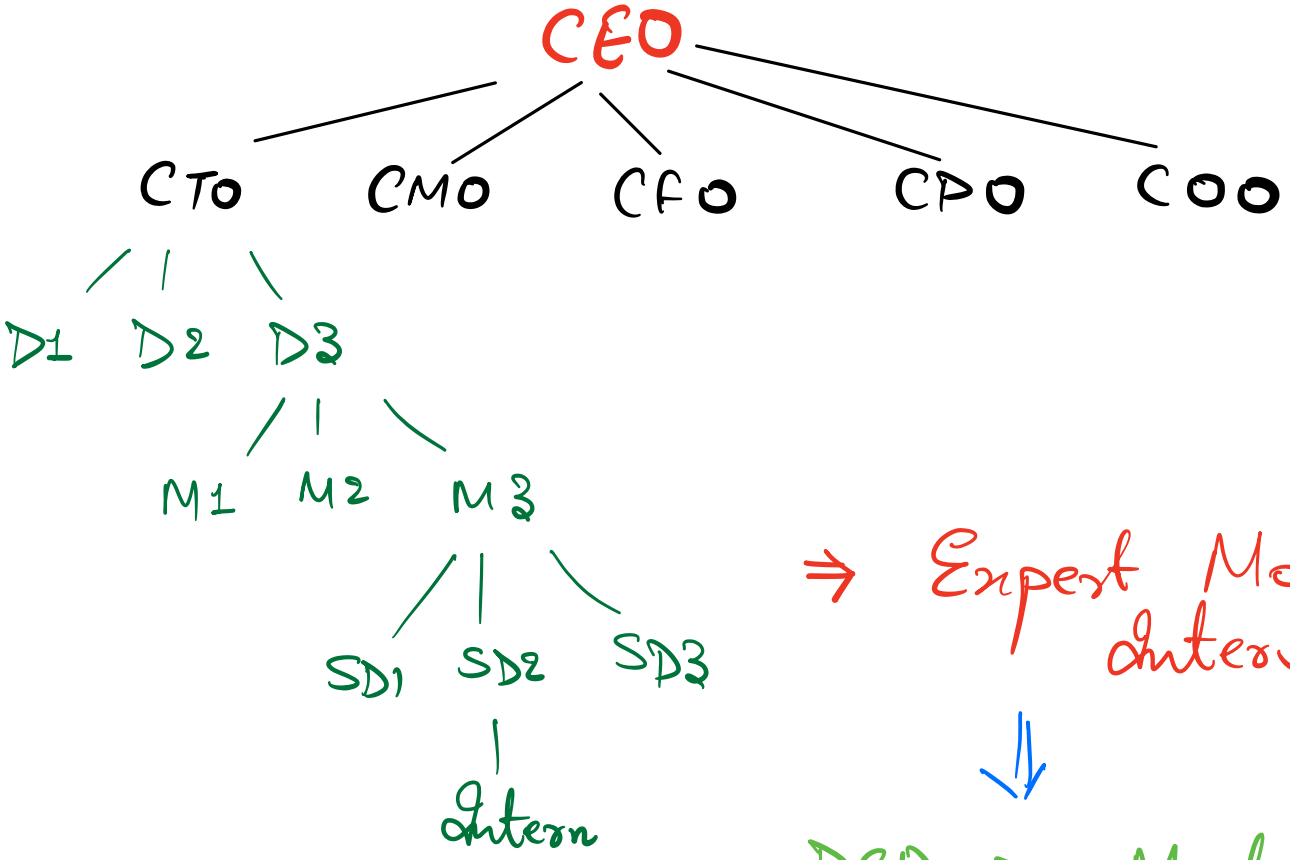
⇓

Strand life Sciences

⇓

Scaler (2021)

Trees ⇒ Hierarchical Data



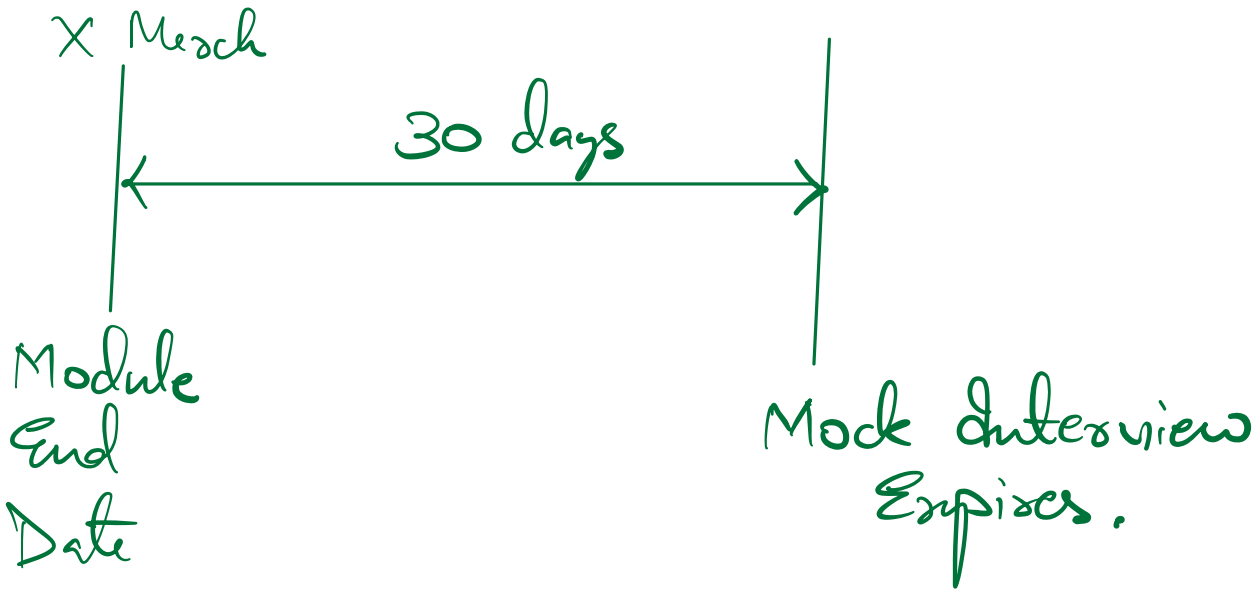
⇒ Expert Mock interviews.

⇓

DSA ⇒ Mock interview

⇓

100% Companies.



20 mentor session

12 (Mock Interviews)

8 (Interactive)

Mandatory (5)

floaters (7)

Reattempt

DSA

language
SQL
LLD
HLD

level ⇒ At par or lower than what we have covered

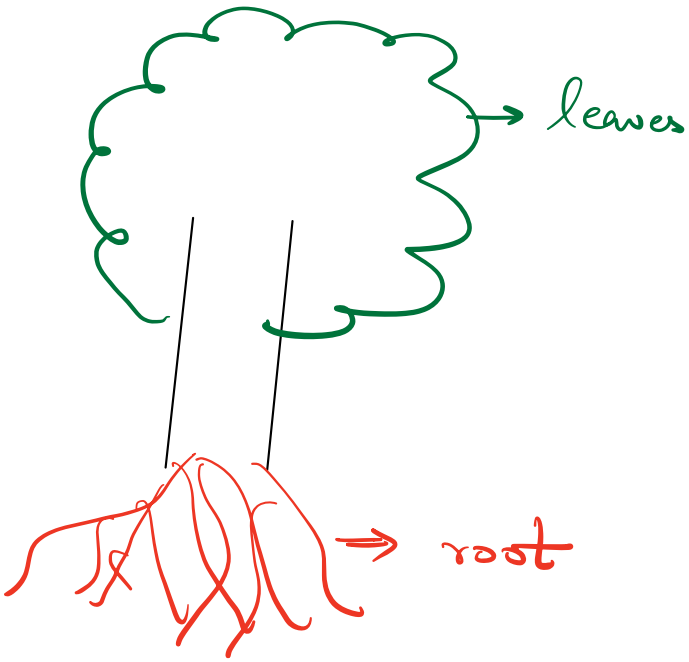
⇓

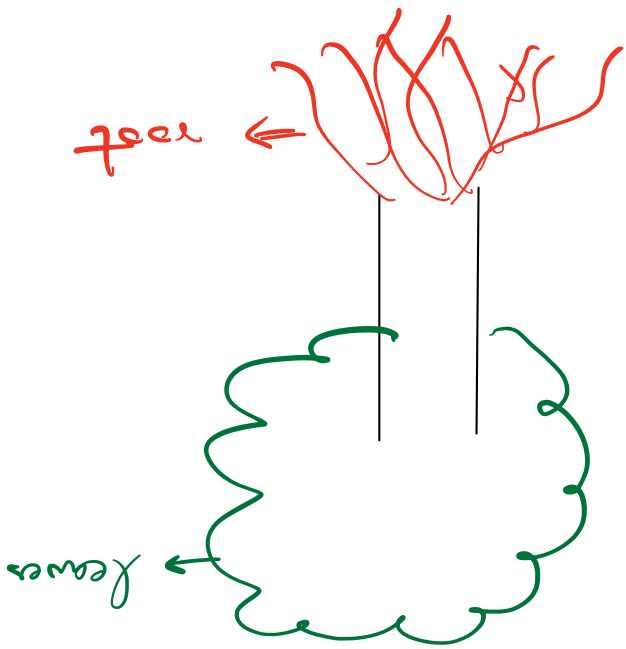
PSP ≥ 70%.

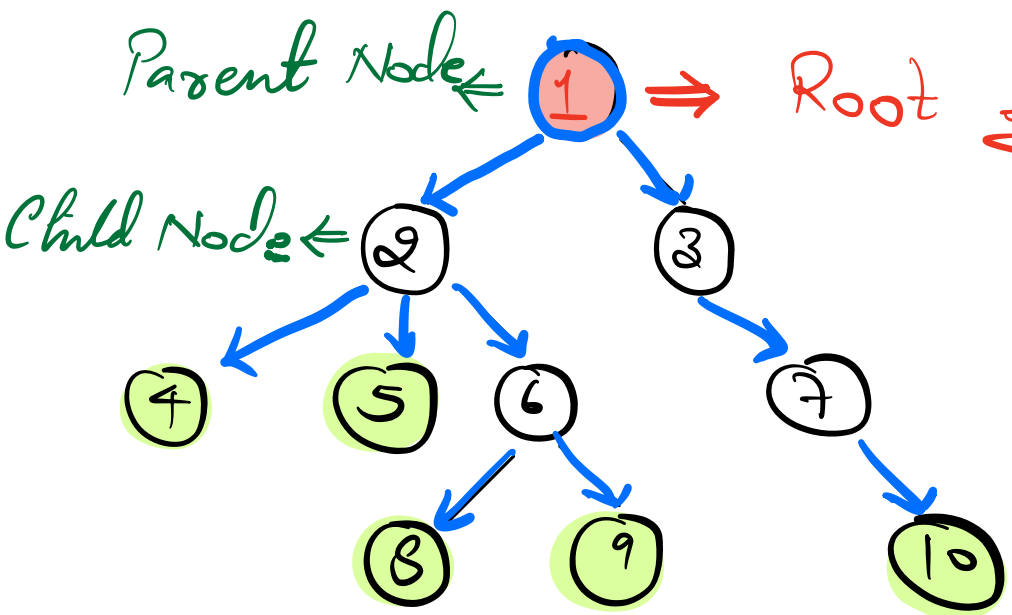
||

90% chance of clearing the MSE

Trees







An element in a Tree ⇒ Node.

- Child nodes of Node 2

  ④, ⑤, ⑥

- Parent Node of ⑦

  ③

Leaf Nodes

↳ Nodes with no children

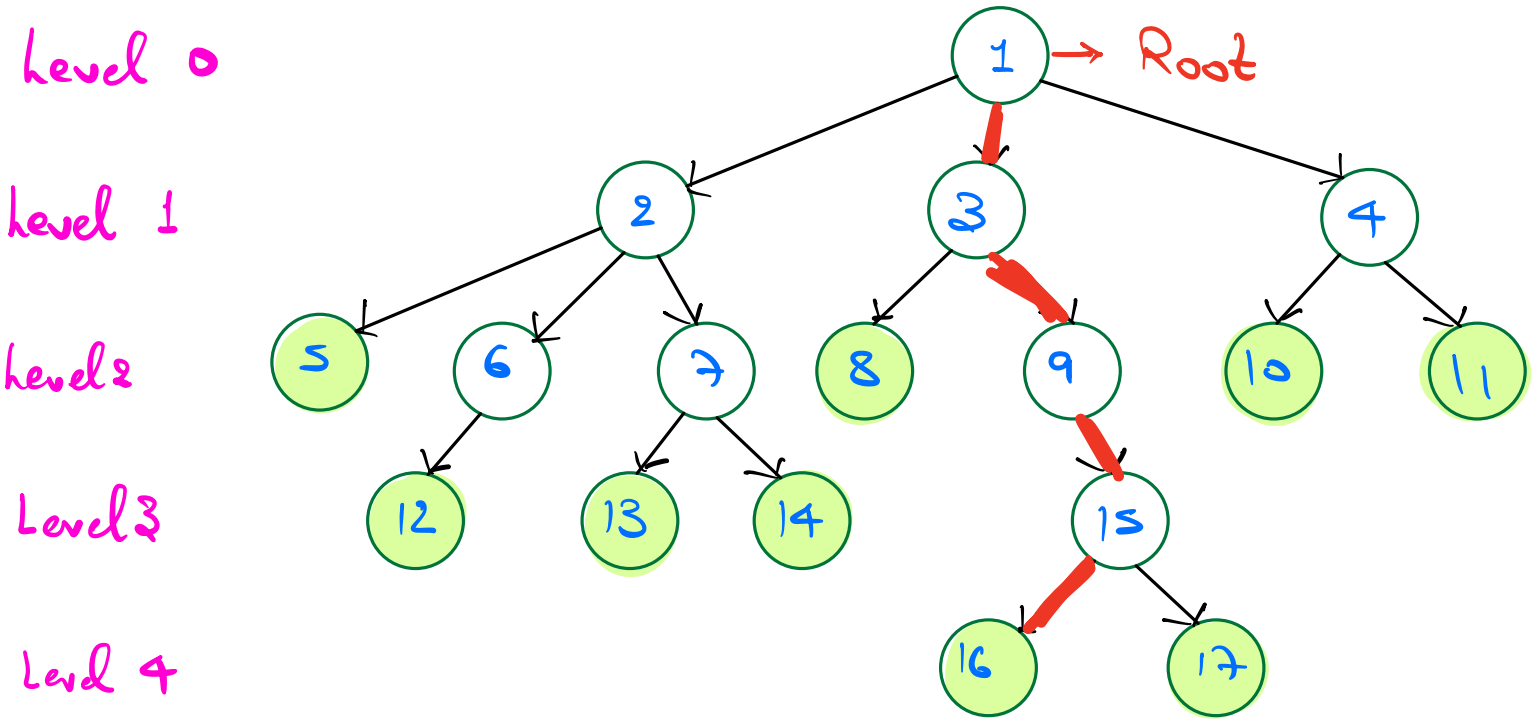
4, 5, 8, 9, 10.

Siblings

↳ Nodes with same parent

(2, 3)

Level, Height & Depth



Depth of a Node

⇒ level at which a node is present
(Distance of the node from root)

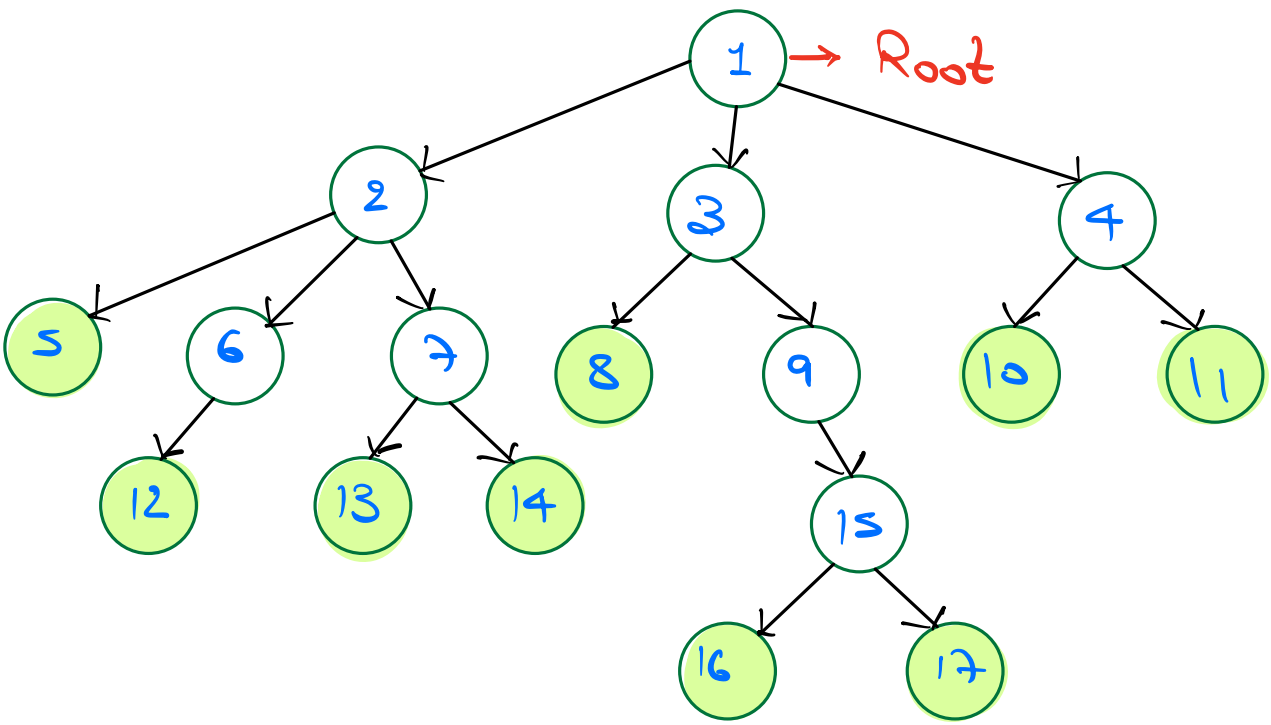
Depth of (14) = 3

Height of a Node

length of the longest path from a node to the farthest leaf.

Height of Node (2) ⇒ 2

Height of root node = 4 = Height of the Tree.



Height of leaf Node = 0.

For a tree with N nodes.
find the total no. of subtrees possible ??

N unique root nodes possible $\Rightarrow$ N unique subtrees possible

---

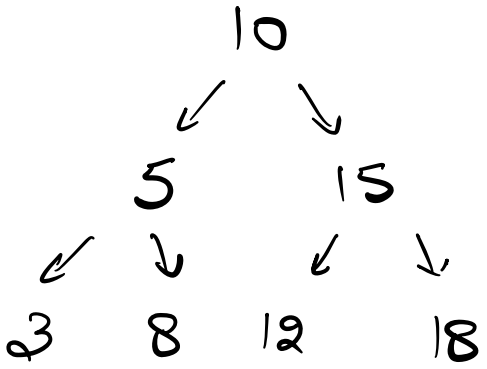
Nomenclature of a Tree

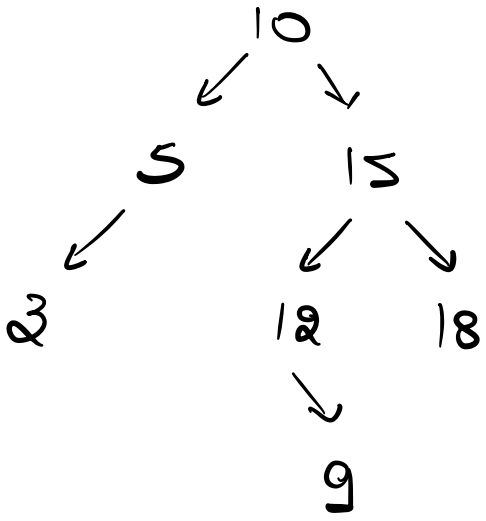
$\Rightarrow$ A tree is named on the basis of max. no. children any node can have.

Max no. of children	Name
2	Binary tree
3	Ternary tree
⋮	
N	N-ary tree

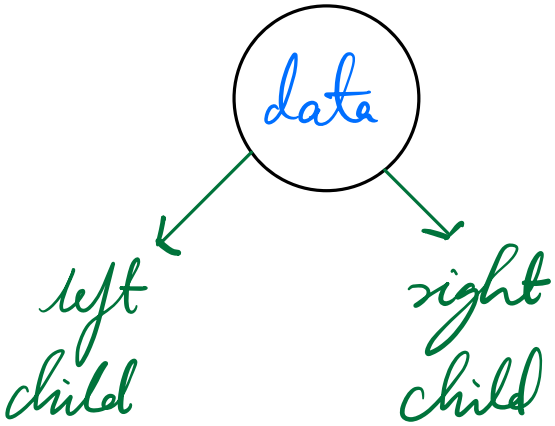
Binary Tree

⇒ A tree where any node can have atmax 2 children (0, 1 or 2)



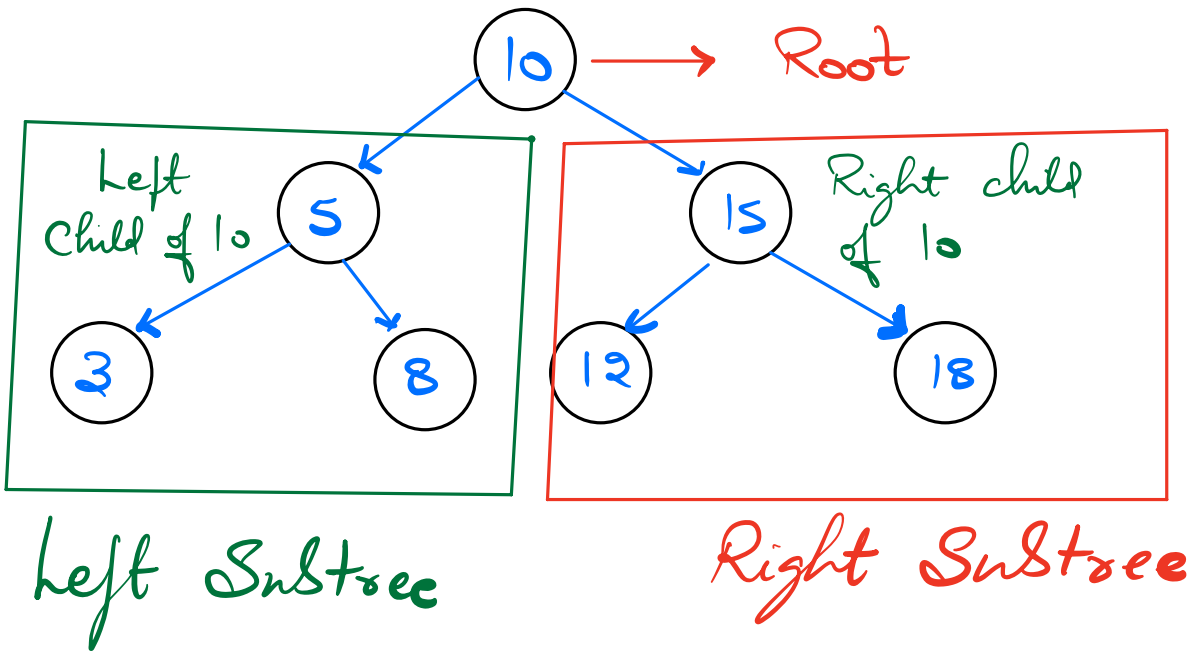


Structure of a Node in B.T.



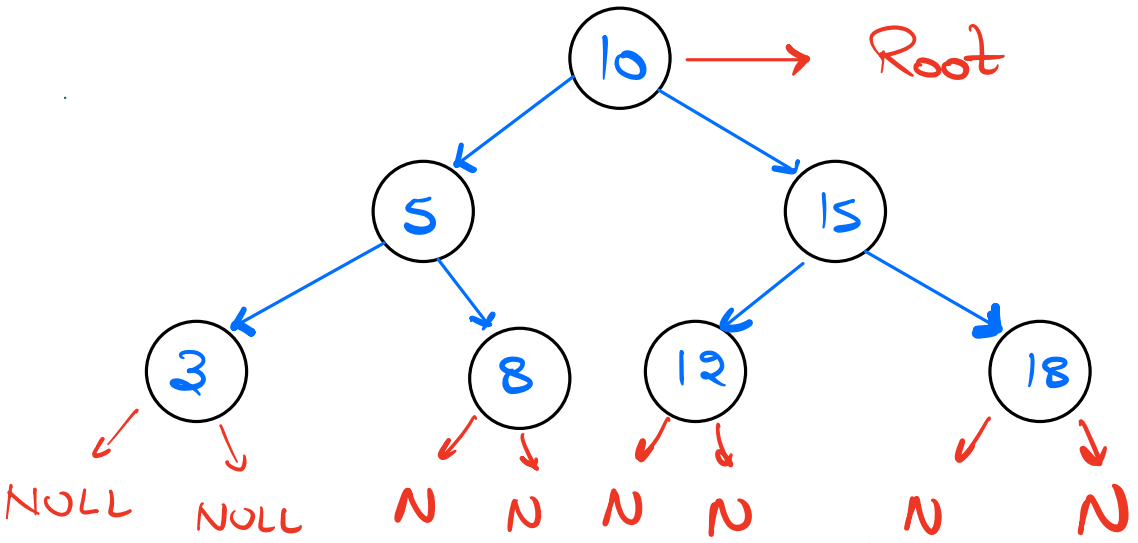

```
class Node {
    int data
    Node left;
    Node right;

    Node (int x) {
        data = x;
        left = right = NULL
    }
}
```

Traversals of Tree



Subtree rooted at 10 ⇒ Entire Tree

Left subtree of 10 ⇒ Subtree rooted at left child of 10



1) Pre Order Traversal ⇒ Node, Left, Right

2) In Order Traversal ⇒ Left, Node, Right

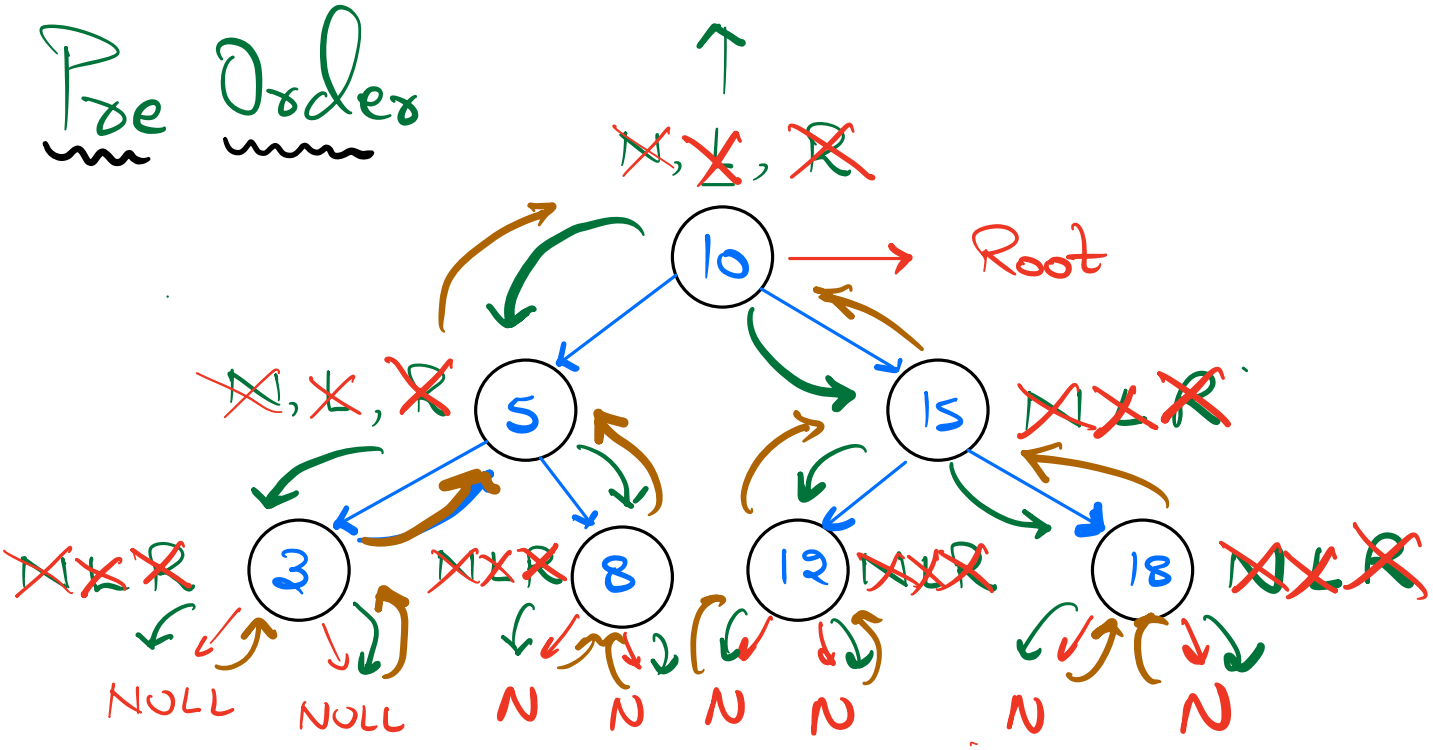
3) Post Order Traversal ⇒ Left, Right, Node

4) Level Order Traversal

5) Vertical Order Traversal

} Next Class

Pre Order



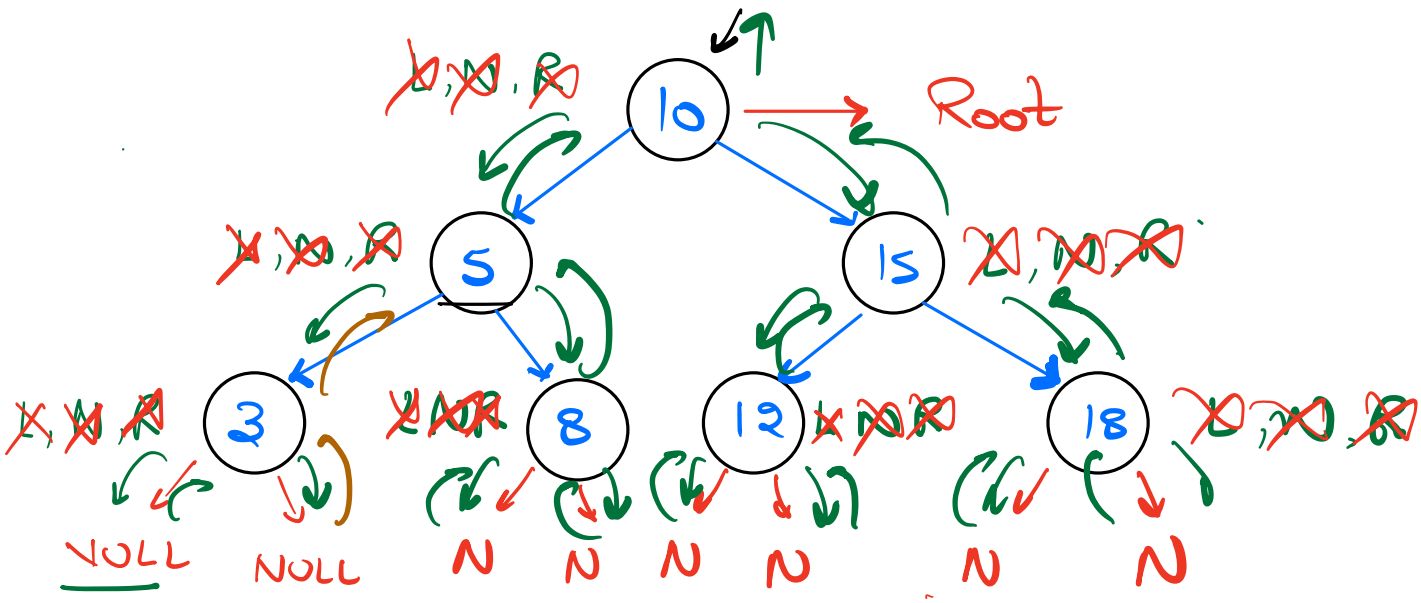
Ans: 10, 5, 3, 8, 15, 12, 18

Code

```
void preOrder ( Node root) {
    if (root == NULL) { return ; }

    print ( root.data);
    preOrder (root.left);
    preOrder (root.right);
}
```

Node
Left
Right

In-Order Traversal



3, 5, 8, 10, 12, 15, 18

```
void inOrder (Node root) {
    if (root == NULL) { return ; }
    inOrder (root.left);
    print (root.data);
    inOrder (root.right);
}
```

Left
Node
Right

Post Order

H.W.

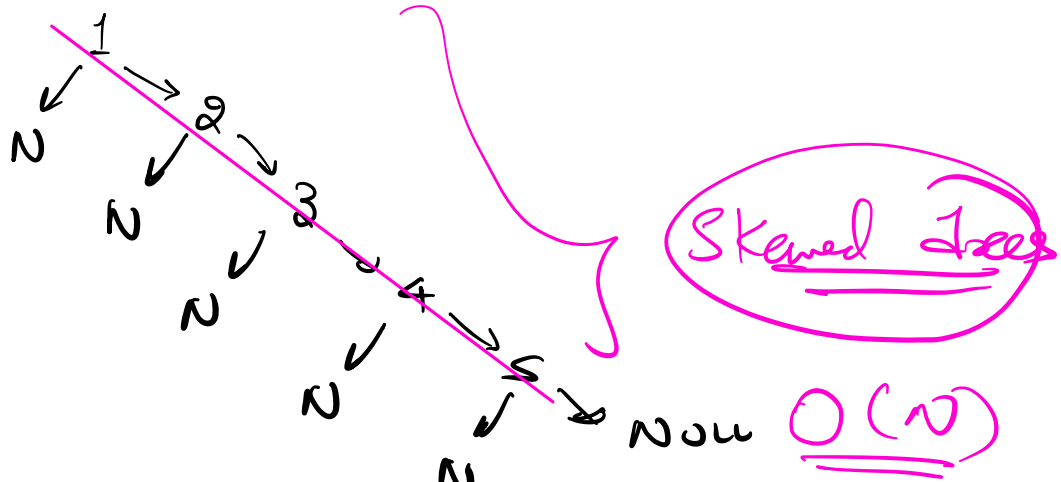
```
void postOrder (Node root) {
    if (root == NULL) { return ; }
    postOrder (root.left);
    postOrder (root.right);
    print (root.data);
}
```

Left
Right
Node

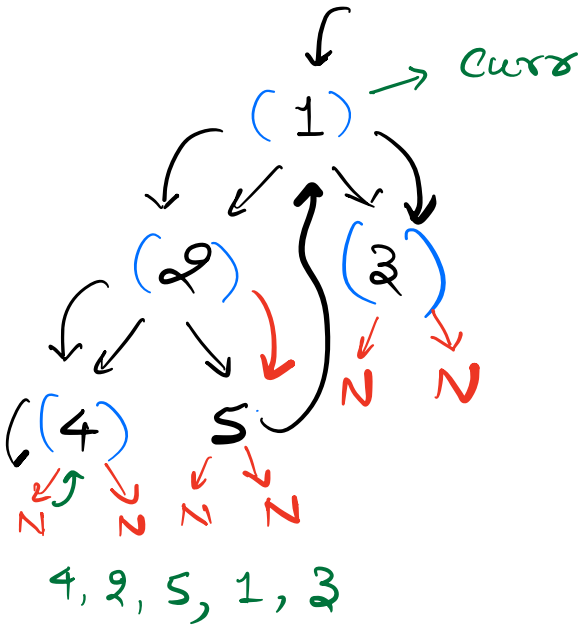
T.C. of Recursion = (# Recursive Calls) × (Time per Recursive Call)

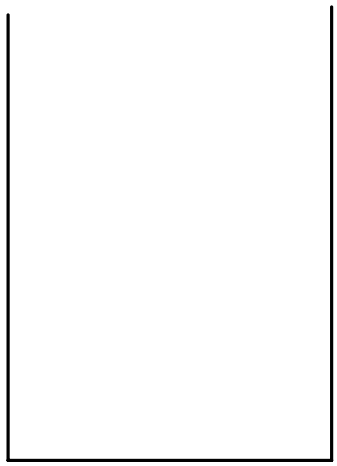
T.C. = $O(N) \times O(1)$

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



Q Can we perform an iterative inOrder traversal of a Binary Tree



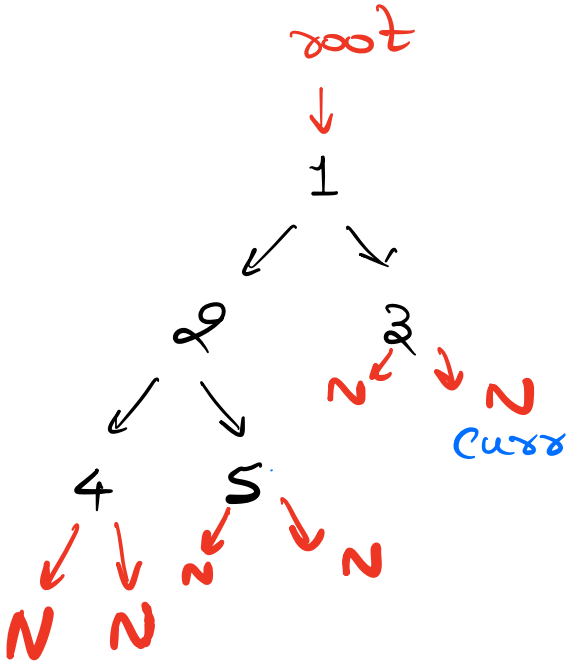


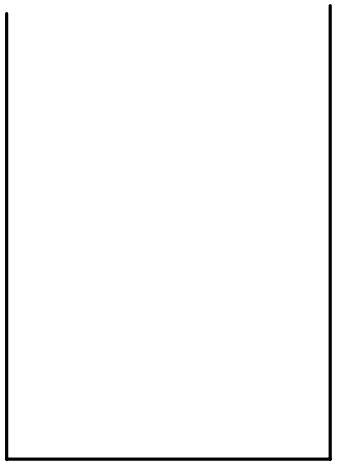
4 2 5 1 3

Code

```
Stack<Node> st
Node curr = root;

while (curr != NULL || !st.isEmpty()) {

    if (curr != NULL) {
        st.push(curr);
```

```
        curr = curr.left;
    }
    else {
        curr = st.pop()
        print (curr.data);
        curr = curr.right;
    }
}
```





4, 2, 5, 1, 3

Wednesday ⇒ Contest

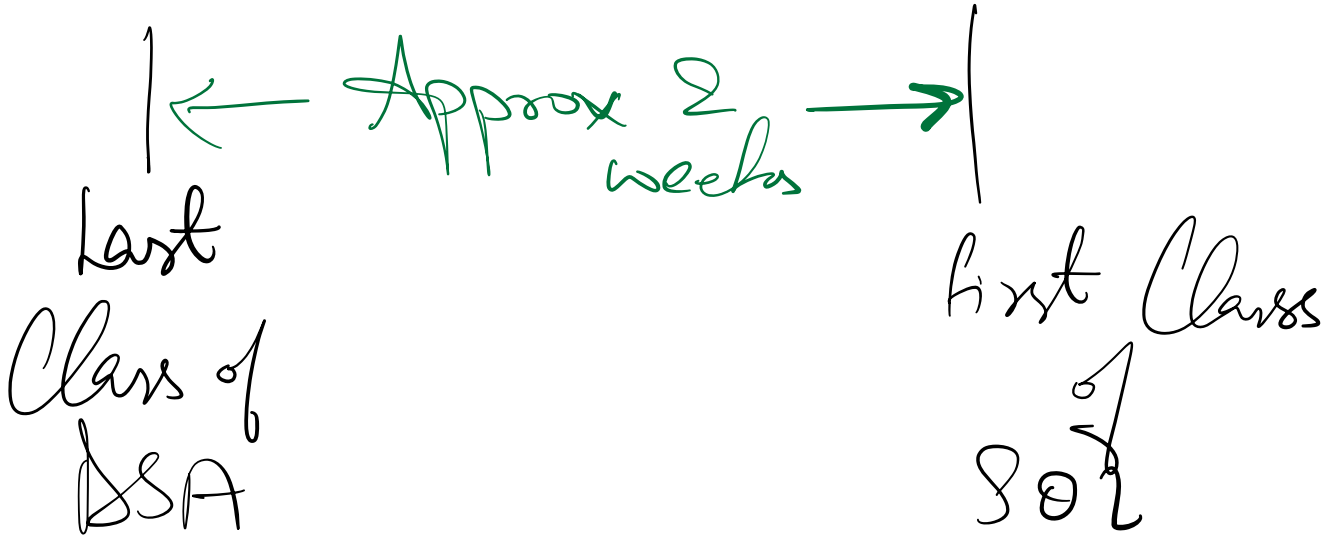
⇓

Linked List, Stack & Queue

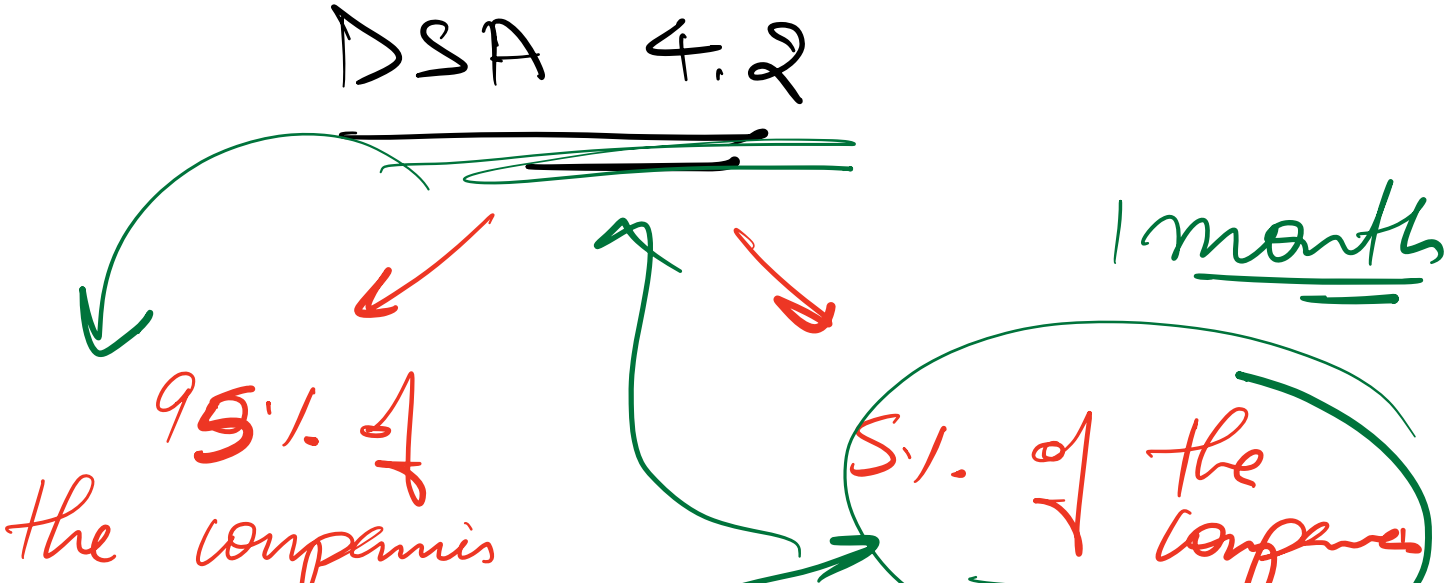
Friday (26th January) ⇒ Off

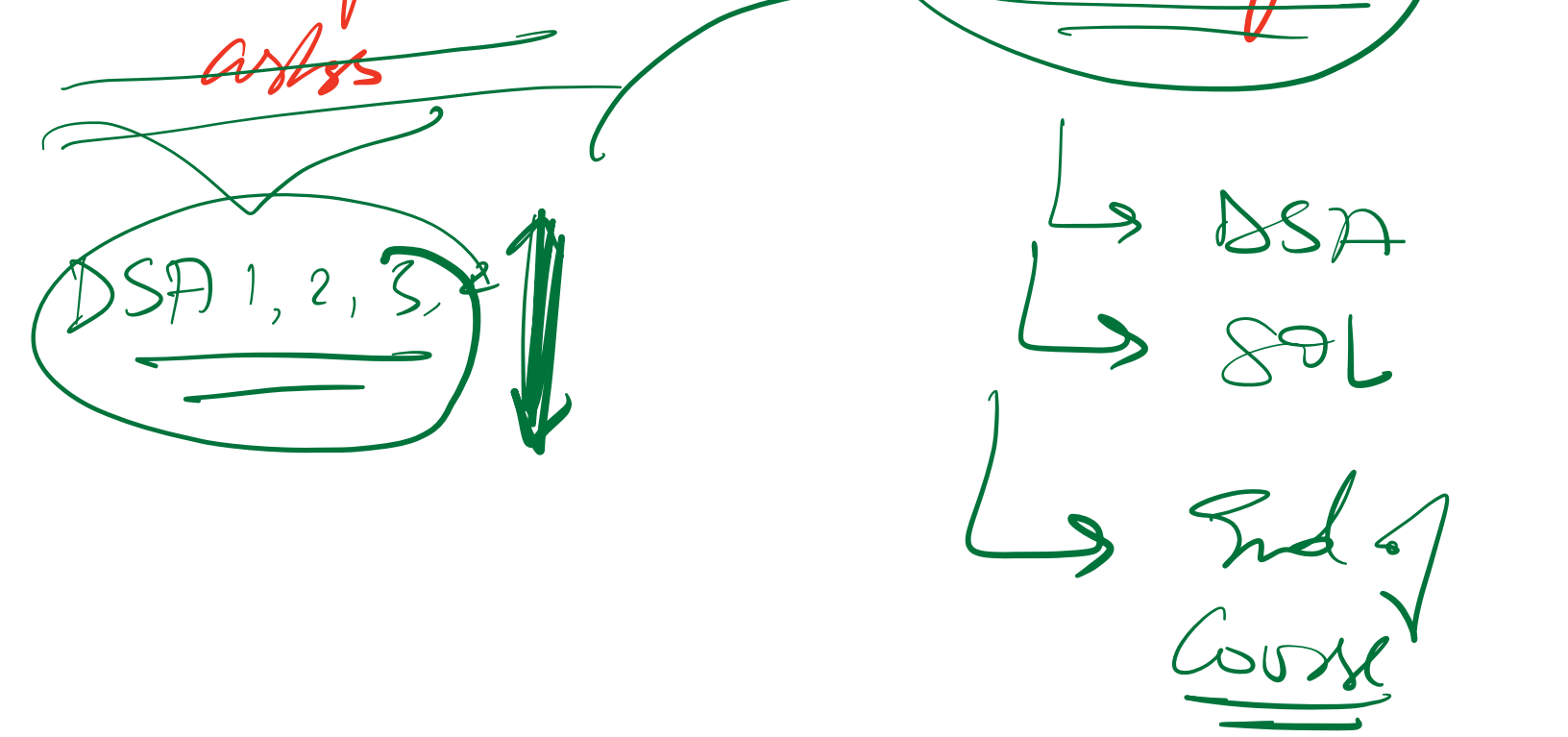
⇓ Next Class

Monday ⇒ 29th January



DSA 4.2




DSA 1, 2, 3,
DSA
SQL
End Course