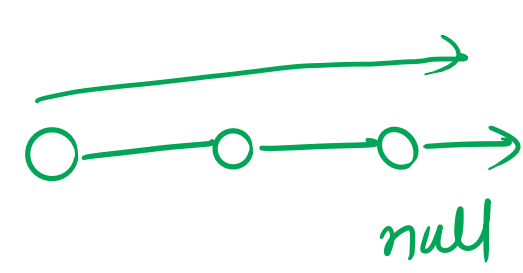
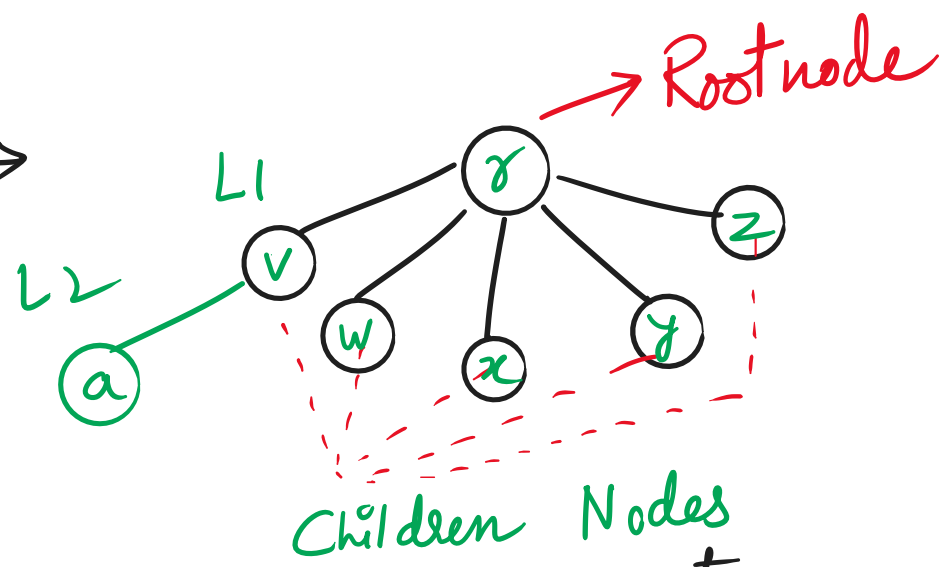


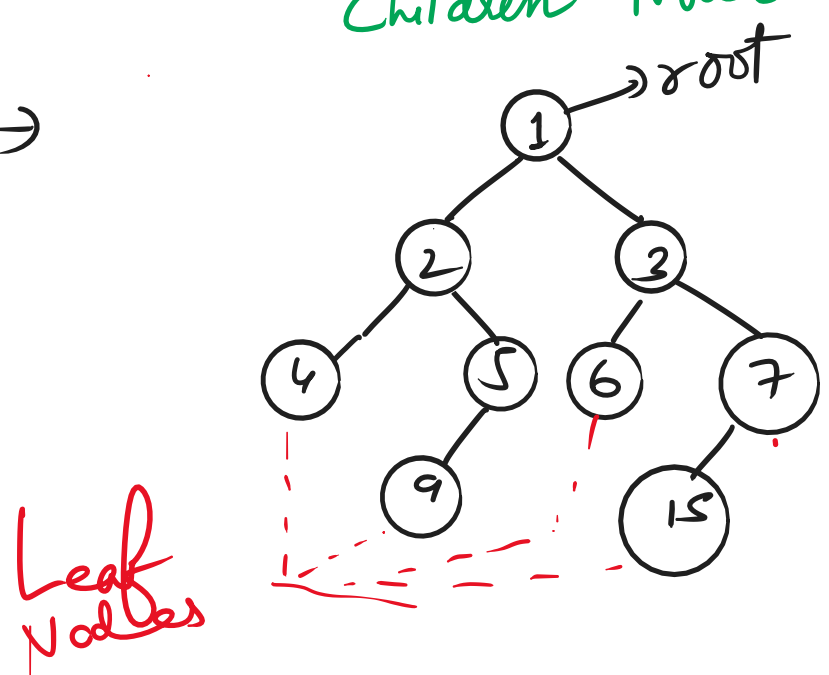
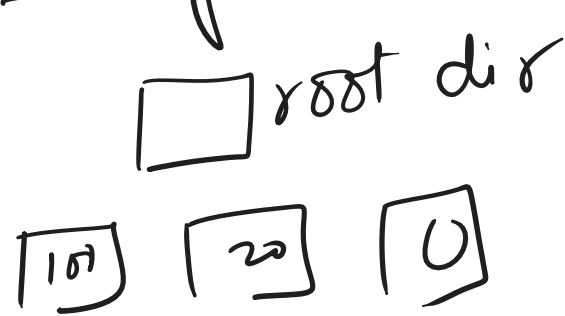
Introduction to non-linear Data Structures : →

Trees :→ It is a non-linear data structure consisting of nodes. The starting node of the tree is called the "root" node.

Normal Tree →



Binary Trees :→



Left Node

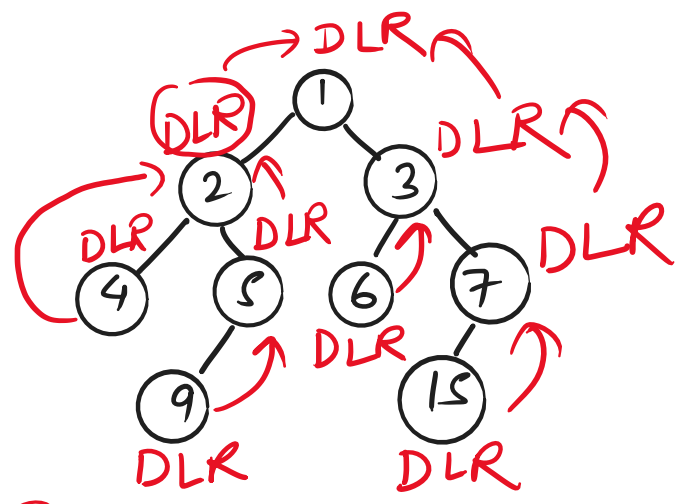
Folder Structure

Tree Traversals :→

DFS Traversal :

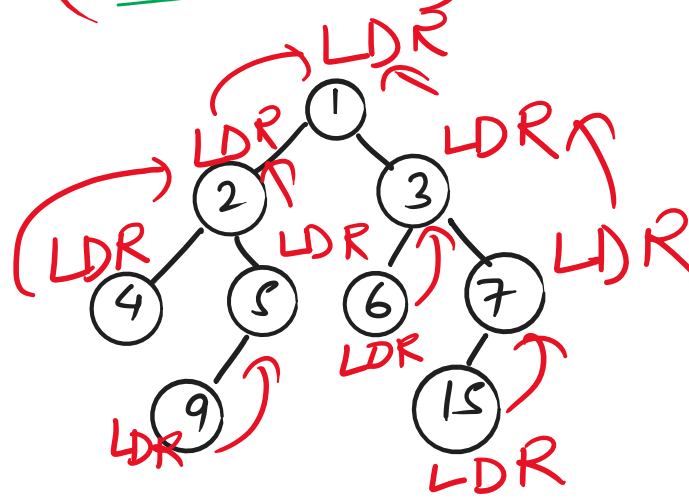
DFS Traversal / Depth First (Recursion) ***

BFS Traversal Breadth First



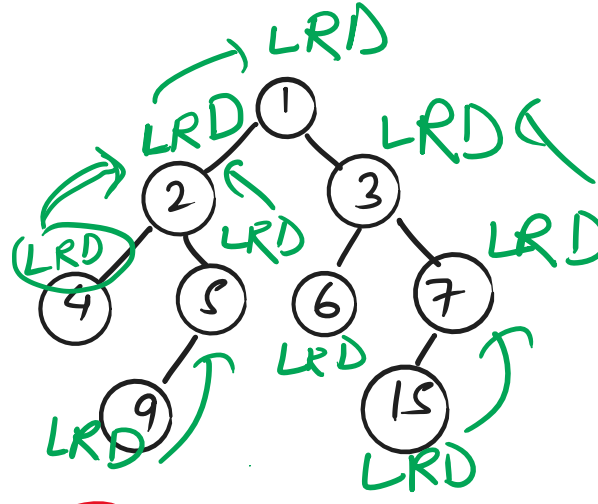
Pre-Order (DLR)

O/p - 1, 2, 4, 5, 9, 3, 6, 7, 15
↓
Root node



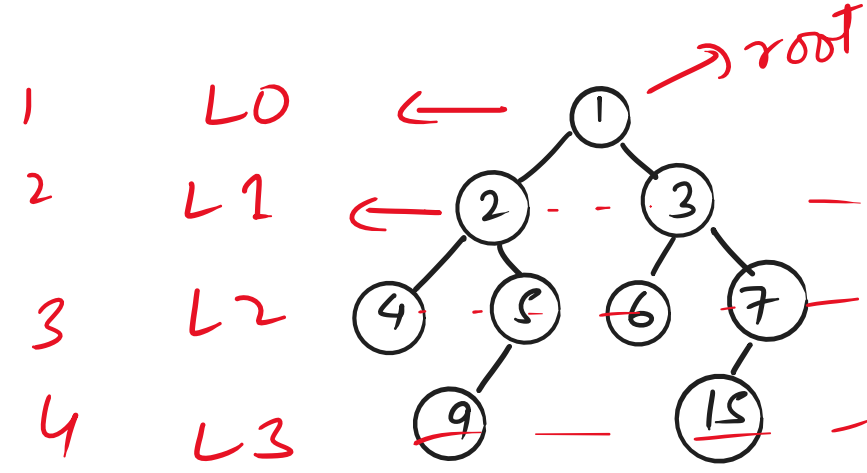
In-Order (LDR)

4, 2, 9, 5, 1, 6, 3, 15, 7



Post Order (LRD)

4, 9, 5, 2, 6, 15, 7, 3, 1
↓
Root



Formation :→

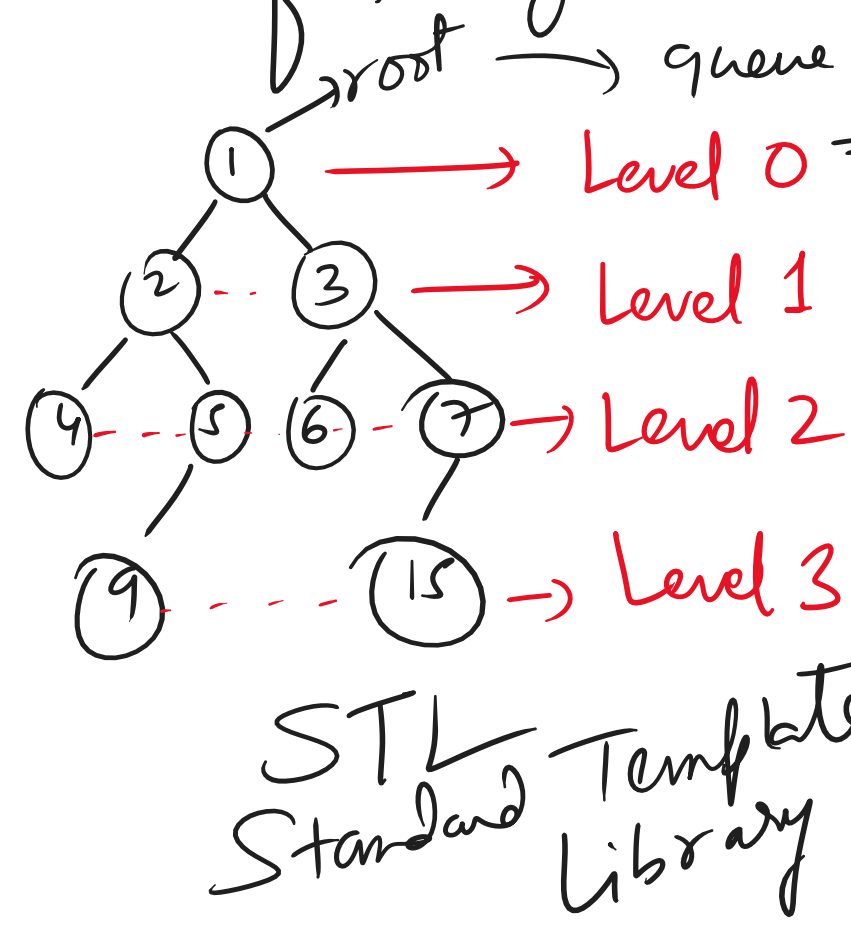
Root → Left → Right

Top → Bottom

Left → Right

Level By Level

* Write a function to perform the "Level Order Traversal" of the given Binary Tree. (BFS) Top → Bottom & Left to Right



O/p → 1, 2, 3, 4, 5, 6, 7, 9, 15

[queue data structure]

[BFS = queue]

[LOT = queue]

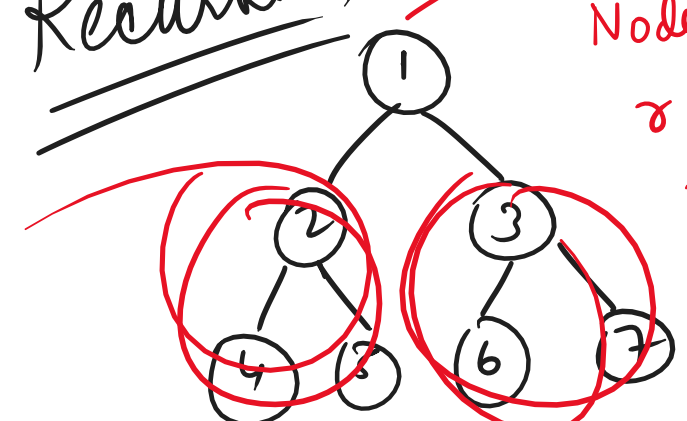
#include <queue>

STL Template Library

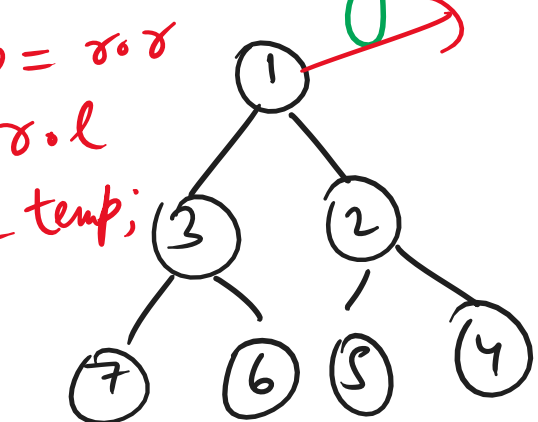
Binary Trees Important Interview Questions :→

① Mirror of a Binary Tree

Recursion



Node * temp = root
root = root->right
temp = temp->left



In-order
4 2 5 1 6 3 7

In-order
7 3 6 1 5 2 4

TCS → 2021 to 2024
Accenture → 2019 to 2024
Capgemini → Jan' 2025

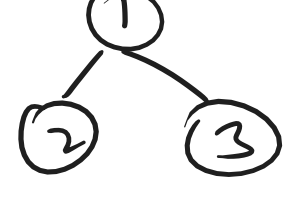
mirrorTree(root)
↳ mirror image

```
int temp = a;
a = b;
b = temp;
```

* Identical Trees :

bool areIdentical(t1, t2) { }

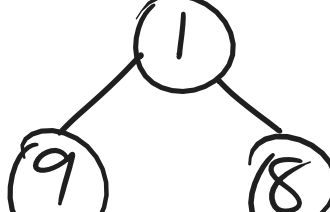
Tree 1



O/p → true



Tree 3



O/p → false

Tree 4



* Standard Template Library

stack, queue, map, list, set, vector

↓
graphs

↳ graphs

* Binary Search Tree (BST)

* Graphs → DFS
BFS