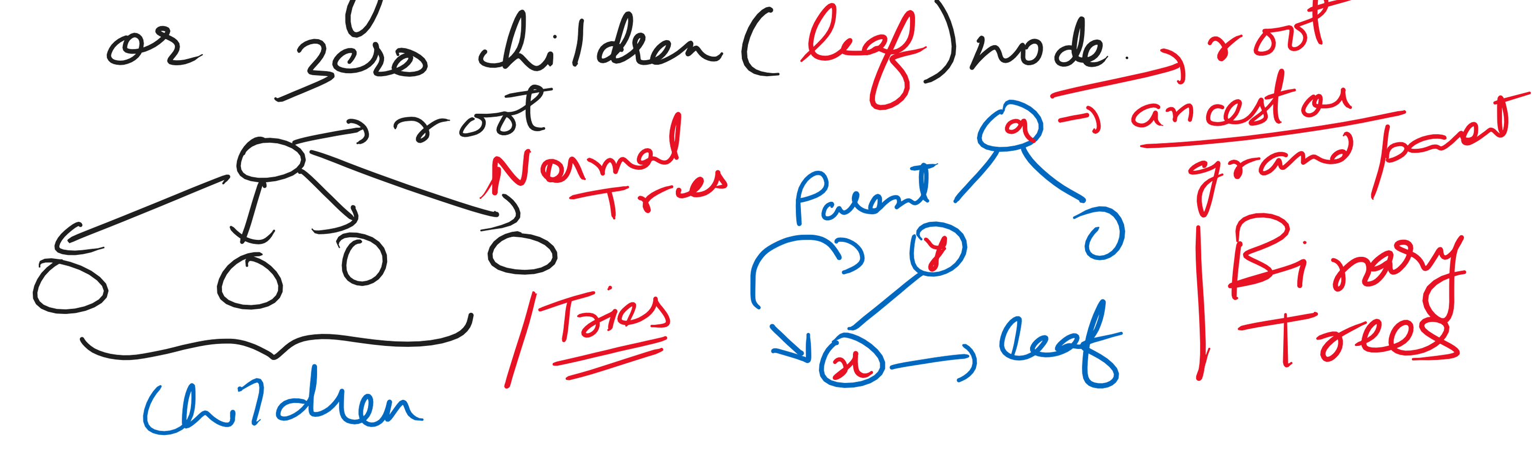
Introduction to non-linear Data Structures:

Trees :→ (Recursion)

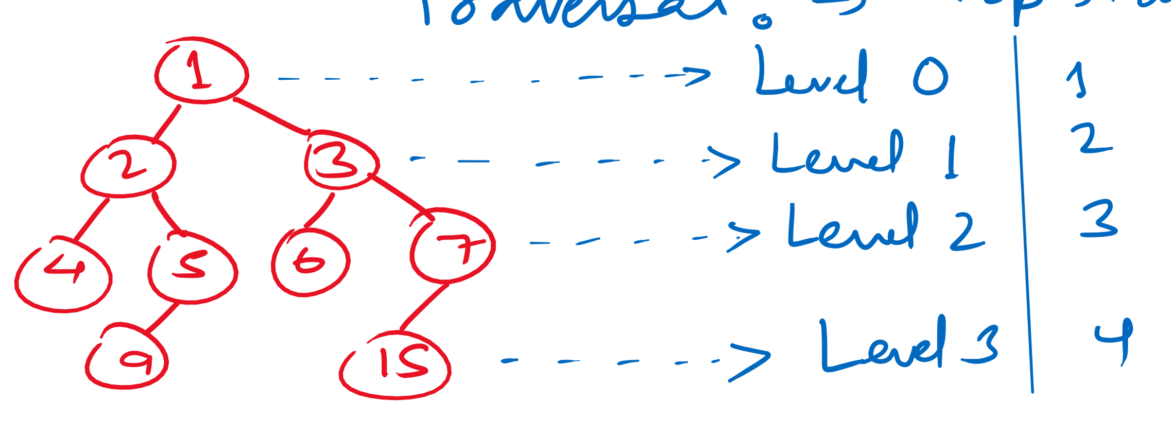
A tree is a non-linear data structure which consists of nodes. Each node has data associated with itself & pointers to it's children nodes. If any node does not have children, it is called a leaf node.

* A normal tree can have multiple children
* A binary tree can have at most 2 children or zero children (leaf) node



Binary Tree Representation :→

Traversal : → Top → Bottom & L → R



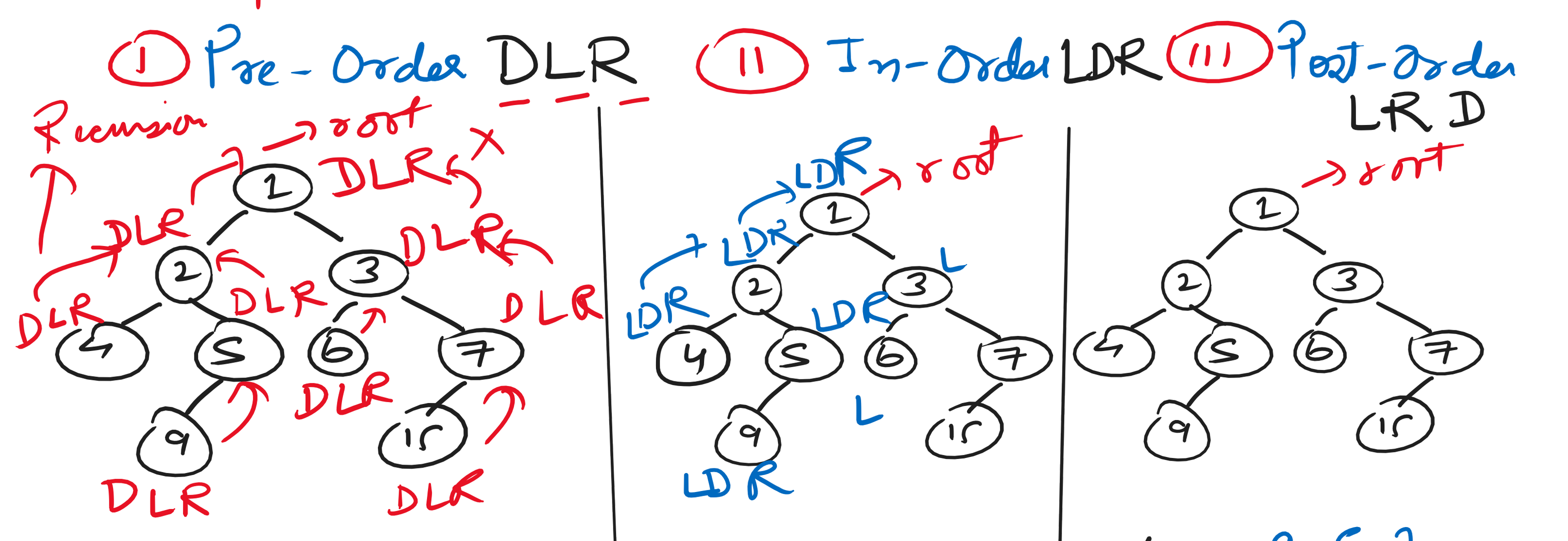
Breadth First Traversal :(Level Order Traversal)

O/P : 1, 2, 3, 4, 5, 6, 7, 9, 15

Queue (STL) #include<queue>

* Depth First Traversal :→

(I) Pre-Order DLR (II) In-Order LDR (III) Post-Order LRD

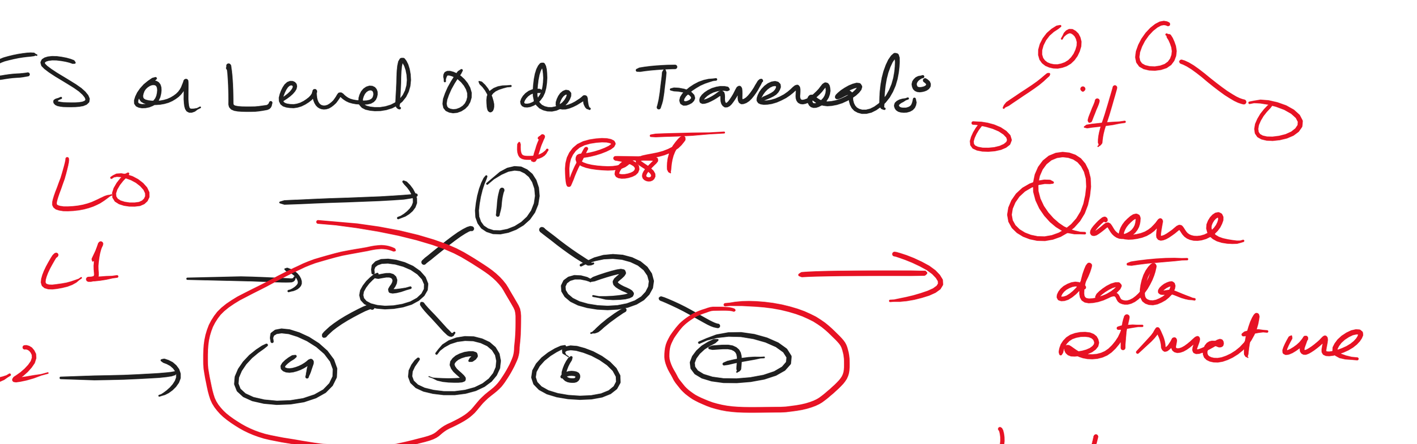


O/P :→ 1, 2, 4, 5, 9, 3, 6, 7, 15

O/P : 4, 2, 9, 5, 1, 6, 3, 15, 7

O/P 4, 9, 5, 2, 6, 15, 7, 3, 1

BFS or Level Order Traversal:



O/P → T → B / L → R

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

queue <TreeNode*>

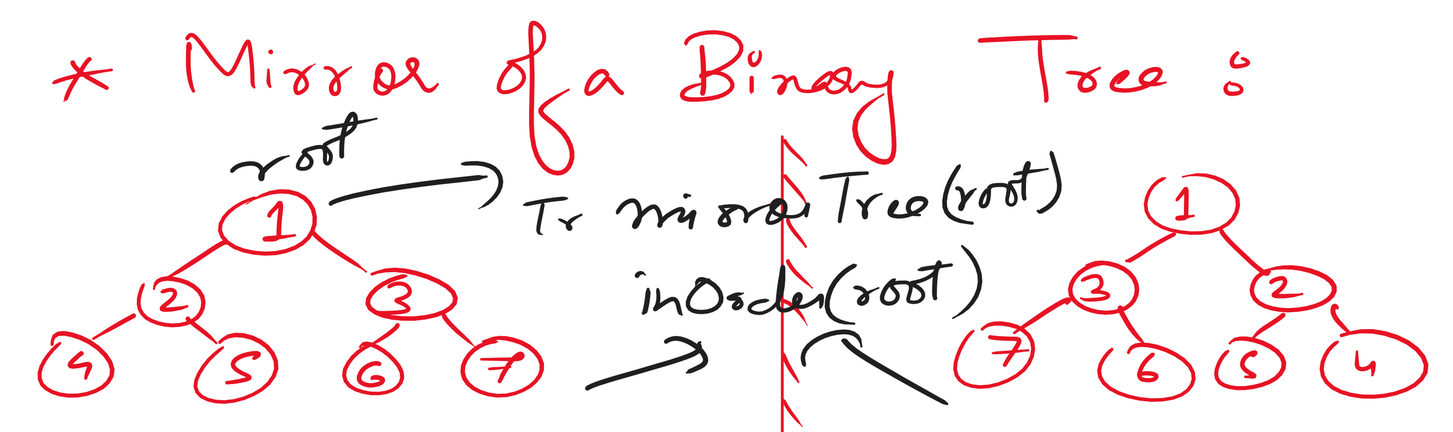
pop → print
1,

Tree Data Structure :→ Important Questions for Company Placement Drives :→

1. BFS or Level Order Traversal → HCL, TCS, IBM, Wipro, (2022, 23)(2024) Accenture
2. Mirror of a Binary Tree : TCS, Accenture, Capgemini, IBM, Oracle → 2025 Jan
3. Identical Binary Trees :→ Accenture, Capgemini, Infosys, Tech Mahindra, Wipro, HCL, Bosch → Feb → 2025, July → 2024, Sep → 2024

Recursion / Leet Code / Coding Ninjas

* Mirror of a Binary Tree :




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

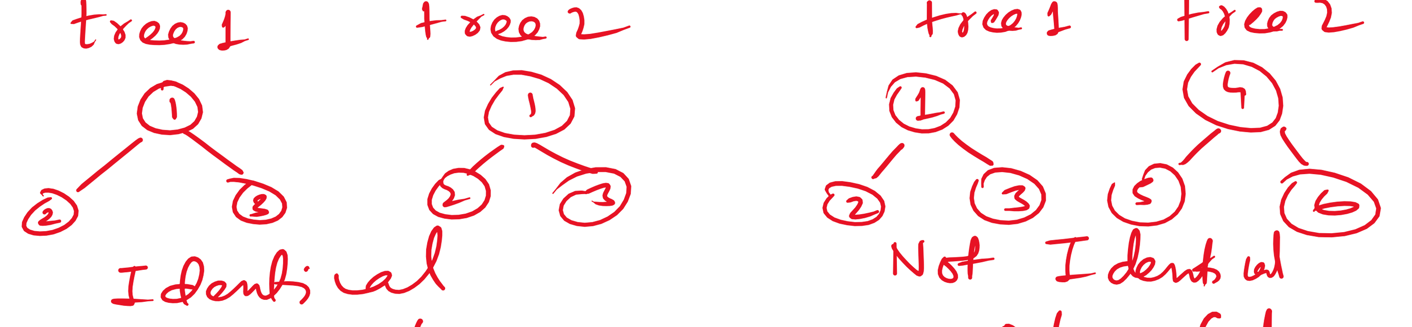
```
TreeNode* temp = root.left;
root→left = root→right;
root→right = temp;
```

InOrder:
4, 2, 5, 1, 6, 3, 7

Inorder:
7, 3, 6, 1, 5, 2, 4

* Given two Binary trees : tree1 & tree2 ; write a C++ Program to check whether the two trees are identical or not. If they are identical, return true or else return false.

Test Case 1 tree 1 tree 2 — Test Case 2 tree 1 tree 2



Identical O/P → true

Not Identical O/P → false

Post Lunch :→

Standard Template Library (STL)

* stack
* queue
* map
* list
* set
* vector
* priority_queue

BST

MST (Graphs & Heaps)