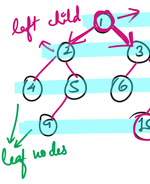
Introduction to non-linear Data Structures: →

* Trees: → Trees are non-linear data structures having entities called nodes. Each node of the tree is connected to it's children nodes.

For example: Here is an example of a Binary Tree

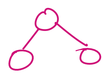
* The starting node is called root node
* If a node has no children, it is called a leaf node.



Folders in PC
Level by level
(T → B, L → R)

(DSP) → Trees

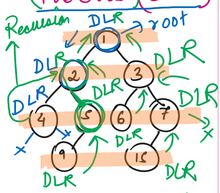
Binary Tree



Traversal: →

(DFS → Depth First Search)
→ (Pre, In, Post)

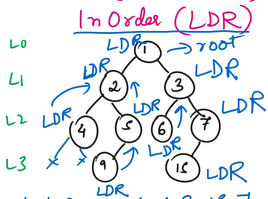
(Pre Order (DLR))



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

(Root)

In Order (LDR)

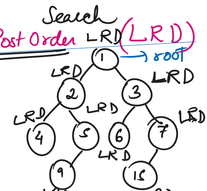


O/p: 4, 2, 9, 5, 1, 6, 3, 15, 7

(Middle)

BFS Breadth First Search

Post Order (LRD)



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

(Root)

D → Data
L → Left
R Right

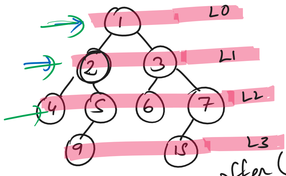
```
class TreeNode {
    int data;
    TreeNode left;
    TreeNode right;
}
```


```
TreeNode(int data) {
    this.data = data;
    this.left = this.right = null;
}
```

DFS / BFS Traversal: →

Top to Bottom, Left to Right, Level by Level:

Level Order → Queue (FIFO) L → R



O/p 1, 2, 3, 4, 5, 6, 7, 9, 15
L0 L1 L2 L3

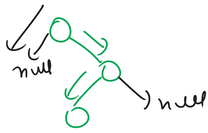
queue [1] front
queue [2, 3] F → R
queue [4, 5, 6, 7] F → R
queue [9, 15] F → R

(STL)

queue.offer( )
queue.poll( )

Collection

(java.util) ***

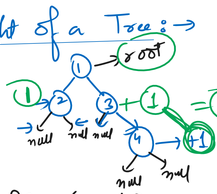


Trees Important Questions: →

* [LeetCode, Coding Ninjas]

* Height of a Tree.
* Identical Trees.
* Mirror of a Tree.
* Left & Right Views of a Tree [** imp].
* Lowest Common Ancestor of two nodes. (LCA)
* Serialize / Deserialize a Binary Tree.
** Sum of nodes with even grandparents. 2024, 2025, 2023
* House Robber III

Height of a Tree: →



Pseudo Code: →

* Calculate the left height
* Calculate the right height
* return max(lh, rh) + 1

findHeight(root) → fh(1)

fh(1) → fh(2) → l(null) + r(null) + 1
fh(1) → fh(3) → l(null) + 0
fh(3) → r(4) → 0 + 1

2 + 1 = 3