Introduction to Trees :→

* Normal Tree :→ An entity containing nodes and children.

* The nodes are pointing having an address & some data.
* The starting node is called the "root" node.

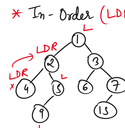
* Binary Tree :→ A tree having at most two children nodes (left & right) is called a binary tree. → Folder Structure in an OS

* Traversals :→
  - ↳ Depth First Traversal
  - ↳ Breadth First Traversal (Level Order Traversal)

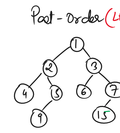
Depth First Traversal :→ (Recursion) ← (DFS)

Pre-order (DLR)

* In-order (LDR)

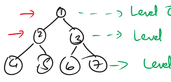


Post-order (LRD)



The tree should be constructed Top to Bottom & left to right (Level By Level)

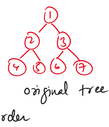
* Level Order Traversal :→ BFS (Queue)



o/p is always top to bottom & left to right :
→ 1, 2, 3, 4, 5, 6, 7

Binary Tree Important Interview Questions :→

* Mirror of a Binary Tree (TCS / Accenture / Infosys)



Node * mirrorTree (root)
{
Node * temp = r.l
r.l = r.r
r.r = temp
}

InOrder
4 2 5 1 6 3 7

InOrder
7 3 6 1 5 2 4

* Given two binary trees T1 & T2 write a function to determine whether they are identical or not.

Tree 1

true

Tree 2

bool areIdentical (t1, t2) {

Tree 3

}

false

Tree 4

(t1 == null && t2 == null) → true
because empty trees

(t1 == null || t2 == null) → false
one is empty

(t1.data != t2.data) → false

recursion

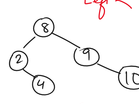
* Height of a Binary Tree :→

The maximum distance from the root node to any of it's leaf nodes.

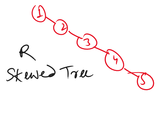
Binary Search Trees :

Every Node has a very important property.

Left < Node < Right (L < N < R) (√n)



1, 2, 3, 4, 5



5, 4, 3, 2, 1

STC
O(n)
ITC
O(n)