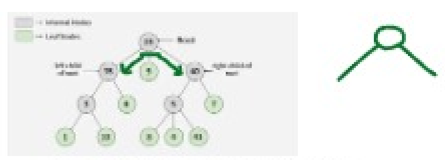
Tree:-

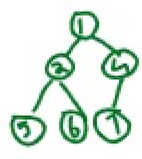
Tree is a data structure which stores the data in the form of node

Tree ==> stores the value inside a node



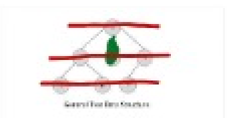
the default left and right child nodes will have null

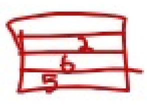
nodes that does not have any children ==> leaf nodes

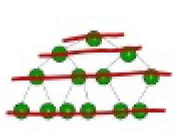


Reading the data from a tree ==> Traversing
1) Reading the data level wise

1,2, 4, 5, 6, 7



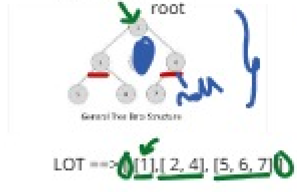




Level OrderTraversal ==> Reading the data level wise

2 ways
1) without recursion(10+)
2) with recursion(3)

1) without recursion



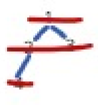
null, null , null, null, null, null

LOT ==> [[1],[2, 4], [5, 6, 7]]

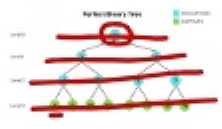
read that data ==> 1
1 ==> left and right

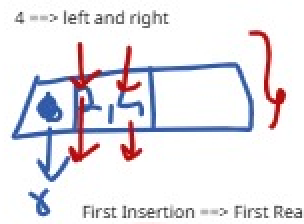
read that 2
2 ==> left and right

read that 4
4 ==> left and right



What is the max depth of this tree
max depth ==> totl no of levels

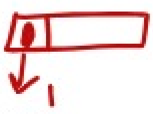




First Insertion ==> First Reading
==> FIFO ==> Queue

List(I) ==> AL, LL and Stack
Set(I) ==> HS, LHS, SS, TS
Queue(I) ==> LL

Tree ==> Binary Tree ,
Binary Search Tree
Red Black Tree
....



List(I) ==> 10 method

Queue(I) ==> 15 method

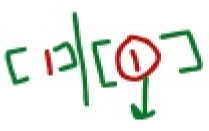
Binary Tree ==> a node can have maximun 2 childreb

LinkedList

List l1 = new LinkedList()
l1 ==> List(I) methods

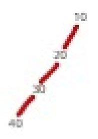
Queue q1 = new LinkedList()
q1 ==> Queue(I) methods

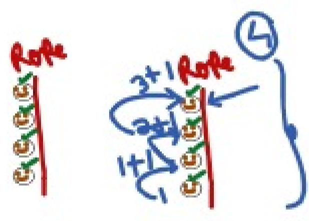
LinkedList l = new LinkedList()
l ==> List(I) methods + Queue(I) methods + extra



count = 2

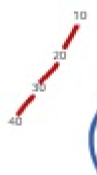
104 ==> return the max depth of that binary tree
count no of levels





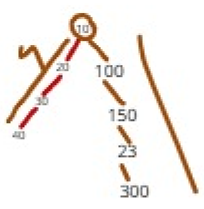
recursion(same thing repeat again and again)

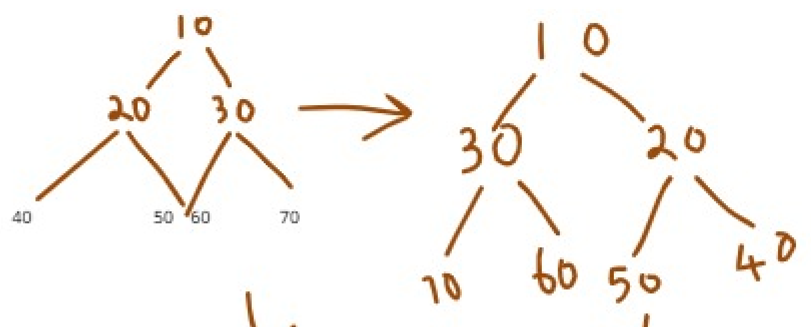
1st ==> call ==> 2nd ==> call ==> 3rd ==> call ==> 4th

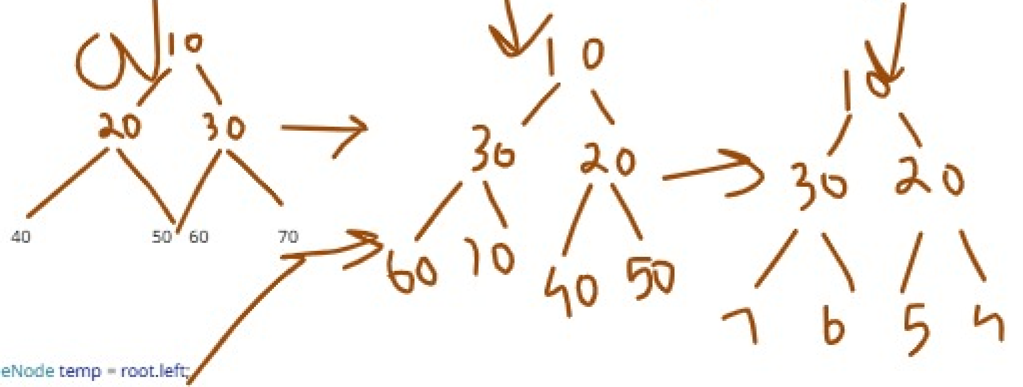


```
public void maxDepth(TreeNode root)
{
 maxDepth(20)
  maxDepth(30) return 1 + 1 ==> 2
   maxDepth(40) return 1 + 0 => return 1
}

maxDepth(10)
```



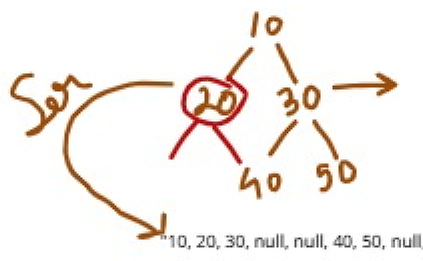




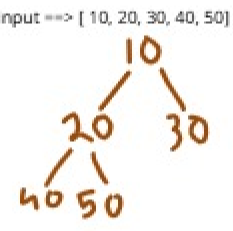
```
TreeNode temp = root.left;
 root.left = root.right;
  root.right = temp;
```

root.left ==> 30 ==>
root.right ==> 20

Serialize and Deserialize Binary Tree

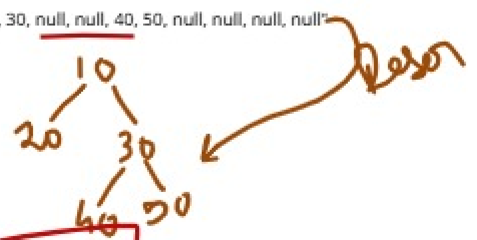


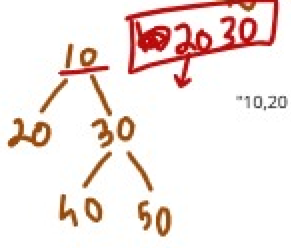
input ==> [10, 20, 30, 40, 50]



"10, 20, 30, null, null, 40, 50, null, null, null, null"

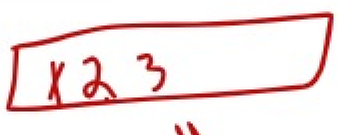
"10, 20, 30, null, null, 40, 50, null, null, null, null"





"10,20 "

[1,2,3,null,null,4,5]



1 2 null null