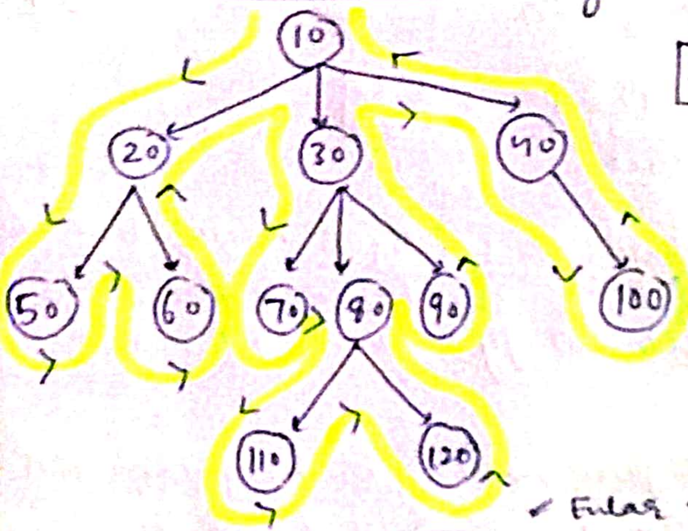
GENERIC TREE CONSTRUCTOR

Hume ek Integer array milega, and hume Tree Construct Krna hai array Ki values Ko lekar.

uske liye → Hume yeh smajhna hoga ki Array kese fill hua aur array ke har value ka matlab kya hai

∴ Eular Path and Filling the Input Array



0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
10	20	50	-1	60	-1	-1	30	70	-1	80	110	-1	120	-1	-1	90	-1	-1

19	20	21	22	23
40	100	-1	-1	-1

✓ Hum Eular-tree Banayengy Aur root se start krke har node aur uske child ko trace krengy aur jis-jis node ko trace krengy usko array me dalty rahengy!

↓

✓ Eular path check krega ki aur koi child toh nahi hai iss node ki around aur uski right side aaty waqt.

✓ Agar koi node nahi milegi toh (-1) array me jayega!

→ Essey krty krty pura tree trace hoga aur array fill hojayega!

Tree kese Construct krengy

✓ Data ko pass kiya (10), Node Banaya aur Node ka Data(10) set krdia!

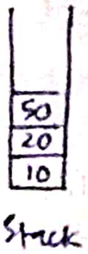
✓ Stack pe gaye! stack ko empty dekh kr node ko root bana dia aur data ko stack me push krdia!

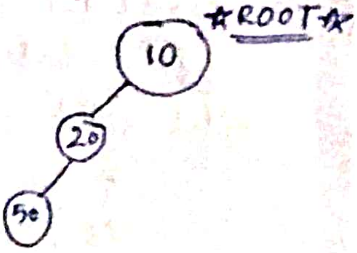
✓ Aab data me (20) ko pass kiya, Node Banaya → Data (20) set kiya, stack me push krne gaye dekha (20) pada hai, toh (20) ko (10) ke children me add, krdia, aur fir stack me (20) ko push krdia!

✓ Aab data me (50) ko pass, node banaya (50) data set krdia, stack me push krney gaye toh dekha phele se kuch hai, (50) ko uska (20) ka child banaya, (50) ko stack me push krdia!

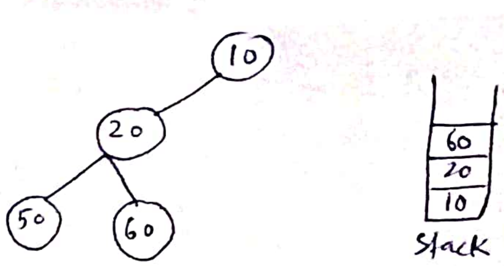
10
20
50
-1
60
-1
-1
30
70
-1
80
110
-1
120
-1
-1

90
-1
-1
40
100
-1
-1
-1





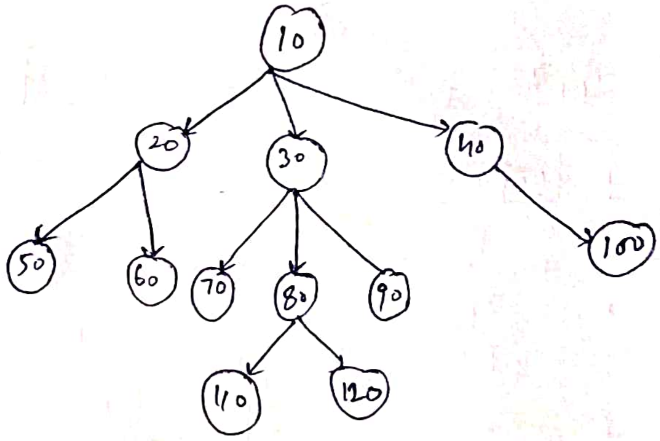
- Aab data -1 pass kiya, stack se Top most ko pop kradia!
- Aab Data me 60 pass kiya, node banaya, 60 data set kiya, stack me dekha phele se kuch hai, toh 60 ko 20 ka child bana dia! aur 60 ko stack me push kradia!



- Aab -1 pass hua, stack se pop hua!

★ Jab-Jab -1 ayega! Tab-Tab stack se pop hoga integer!

Aur yeh pura process essi chalega pure Tree Construction ke tym!



Complete Tree

Code

```
public class Main {
  public static class Node {
     int data;
     Arraylist<Node> children = new Arraylist<>();
  }
  public static void main(String[] args) {
     Node root;
     int[] arr = {10,20,50,-1,60,-1,-1,30,70,-1,80,110,-1,120,-1,-1,90,-1,-1,4,100,
                 -1,-1,-1};
     stack<Node> st = new Stack<>();
     for(int i=0; i< arr.length; i++) {
        if(arr[i]==-1) {
           st.pop();
        }else{
           Node t = new Node();
           t.data = arr[i];
        if(st.size() > 0) {
           st.peek().children.add(t);
        }else {
           root = t;
        }
        st.push(t);
      }
    }
  }
}
```