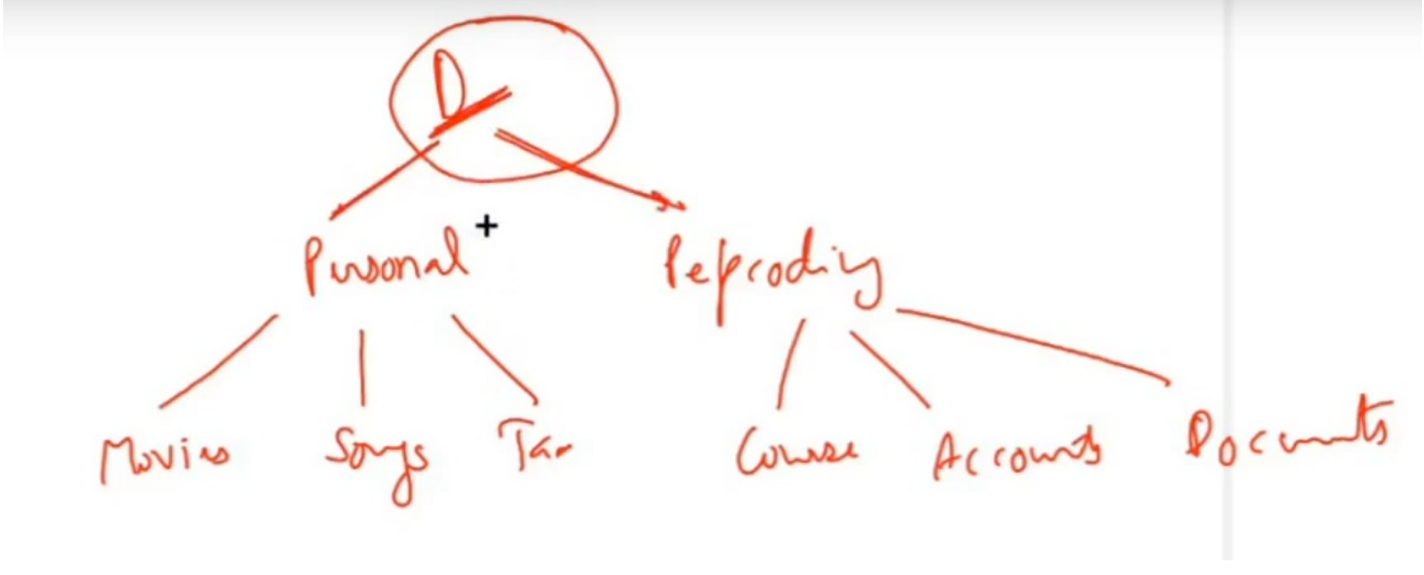
TREE:

required to store the hierarchical type data

Example: D-drive

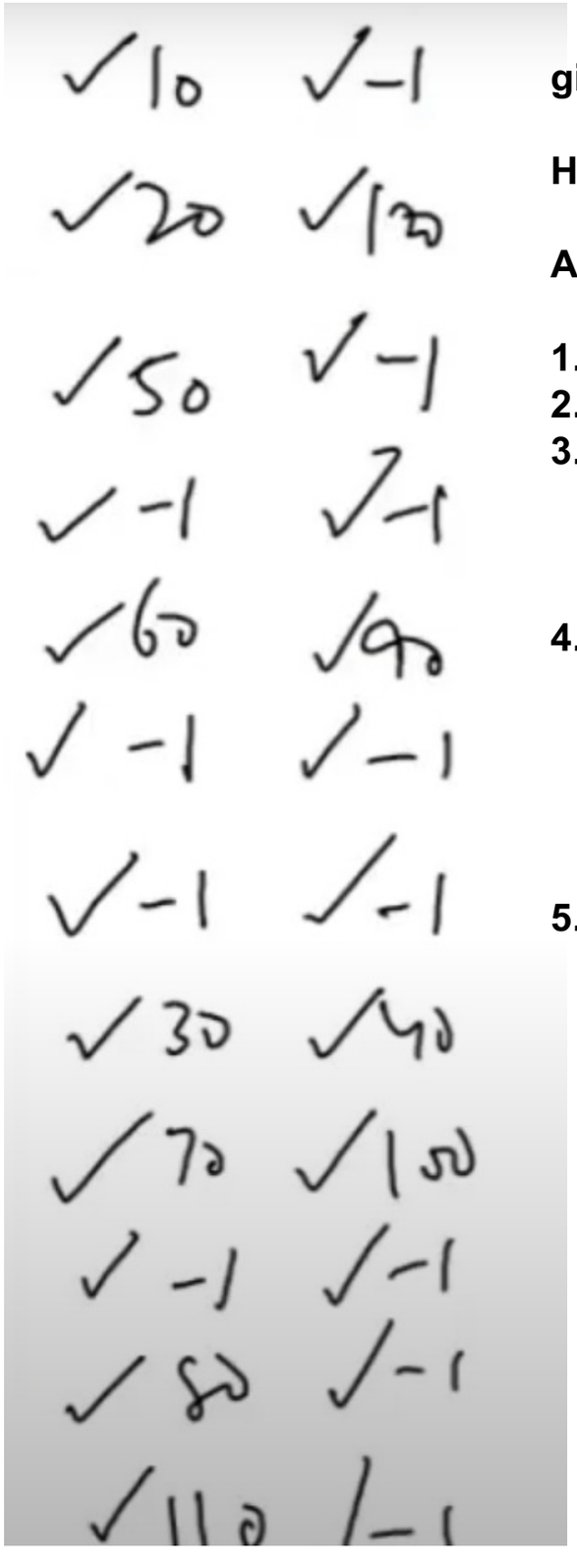


child has no leaf

movie has 2 parent : i.e personal and d-drive

Represenmtation

```
public class Main {
    private class Node {
        int data;
        ArrayList<Node> children = new ArrayList<>();
    }

    public static void main(String[] args) {
        Node root;

    }

}
```



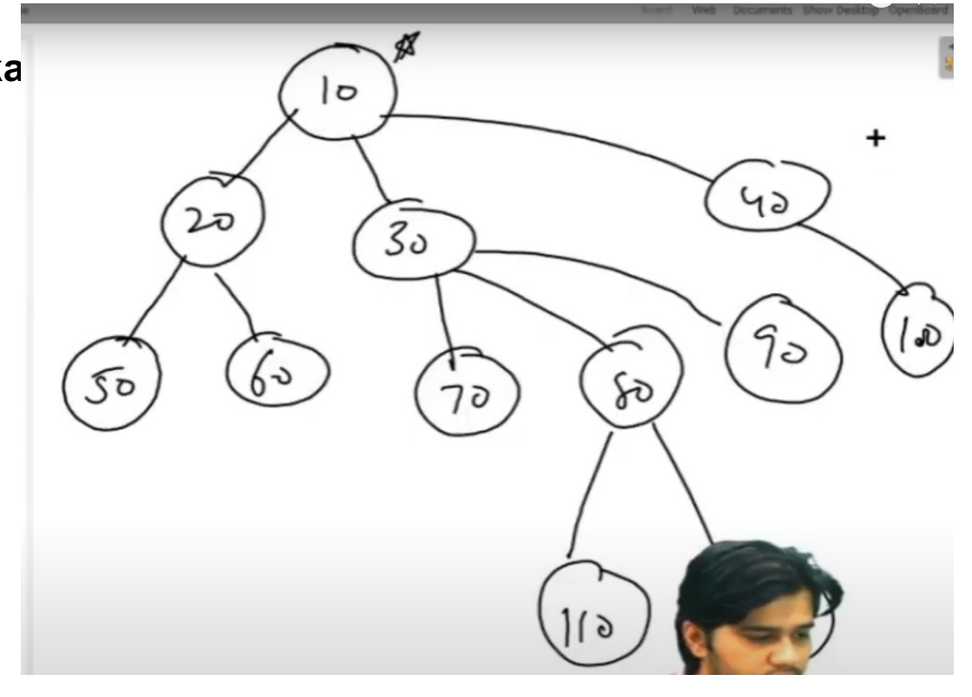
given this data : you need to create tree

How??

Approach

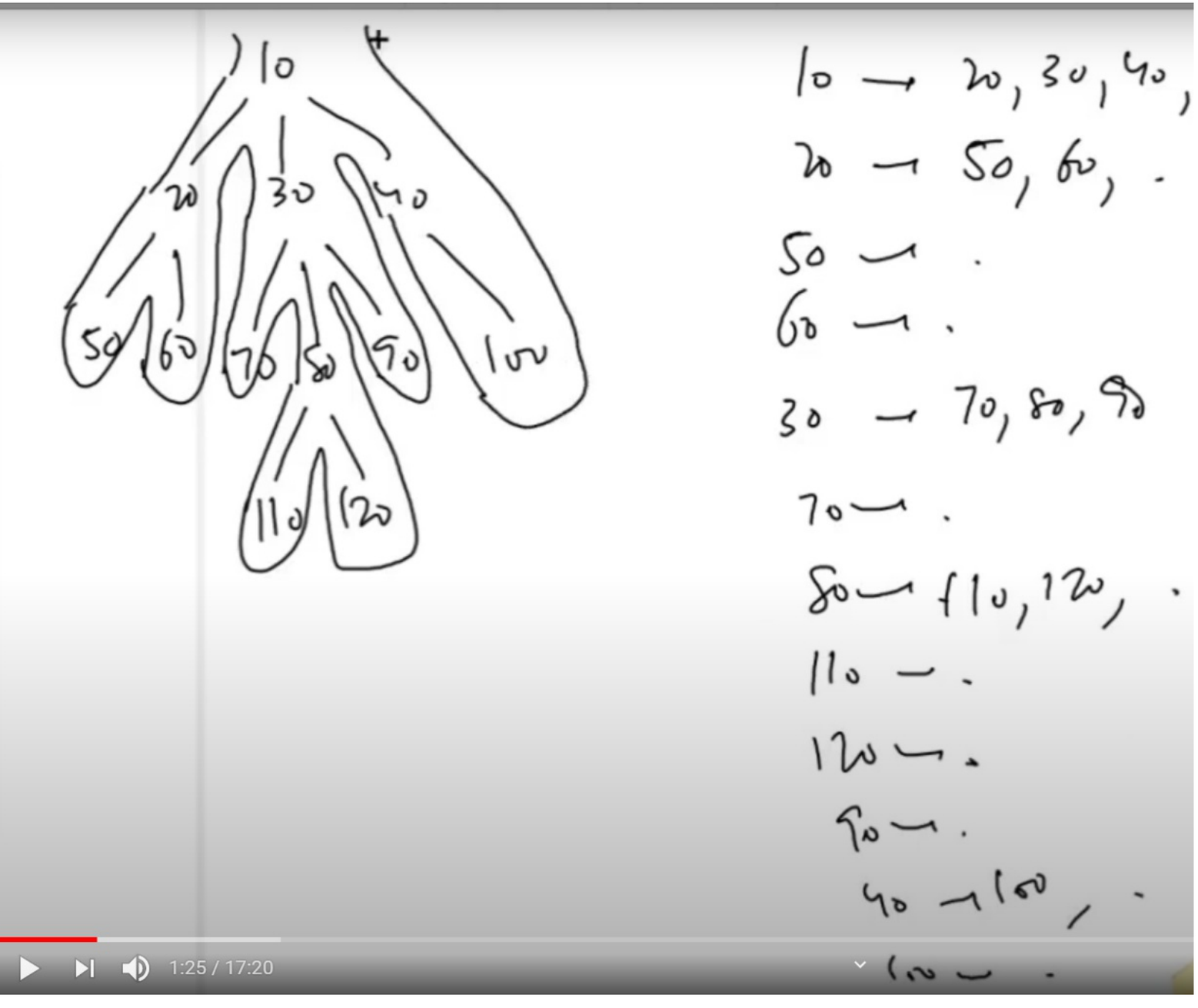
1. Stack use karna he
2. stack empty ho ga start me
3. 1st elemtn aya -
   - create node
   - set data as 10
   - push 10 in stack - mark as root
4. 2nd elemetn aya push
   - creat node
   - set data as 20
   - check the top of stack i.e 10
   - attach 20 to 10
   - push 20
5. do step 4 , and
   - jab bhi -1 ayega toh pop ka

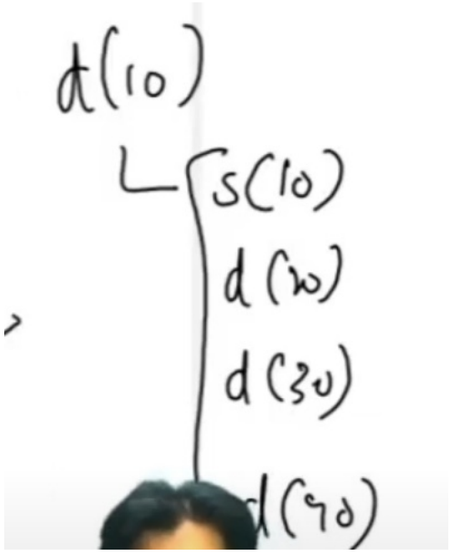
-1 represrent when you traverse euler then at post side we are popping the elemtn



```java
public static void main(String[] args) {
    int[] arr = {10, 20, 50, -1, 60, -1, -1, 30, 70, -1, 80, 110, -1, 120

    Node root;
    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);
        }
    }
```

Display generic tree

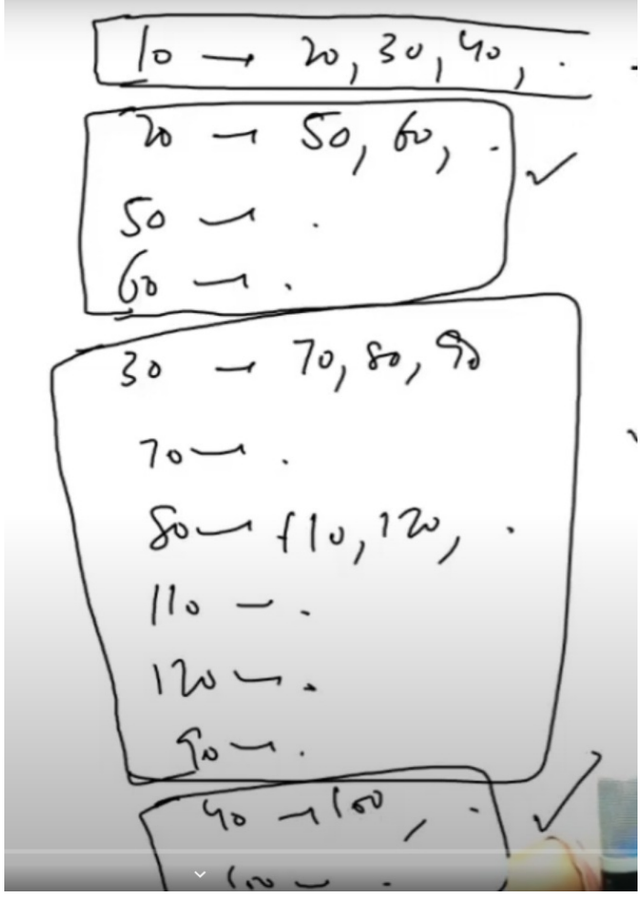




Appraoch :

jab dispaly(10) ayega tab

1. **show 10**
2. **call all children(aur child bhio vahi kaam karenge**



Faith:

display(20)- khudko aur uske family ko print karana janta he

similarly
display(30)and 40

Exp: disp(10)
self ko print kararke
call 20 30 40

```java
// d(10) -> 10 will print itself and it's family
// d(20), d(30), d(40) will print themselves and their families
// d(10) = s(10) + d(20) + d(30) + d(40)
public void display(Node node){
    String str = node.data + " -> ";
    for(Node child: node.children){
        str += child.data + ", ";
    }
    str += ".";
    System.out.println(str);

    for(Node child: node.children){
        display(child);
    }
}
```

knwo how to print root
and its family(i.e child)

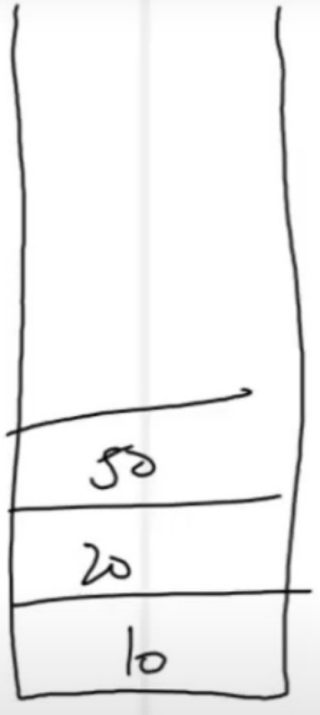
children know how to
print themselve - -faith

level-option

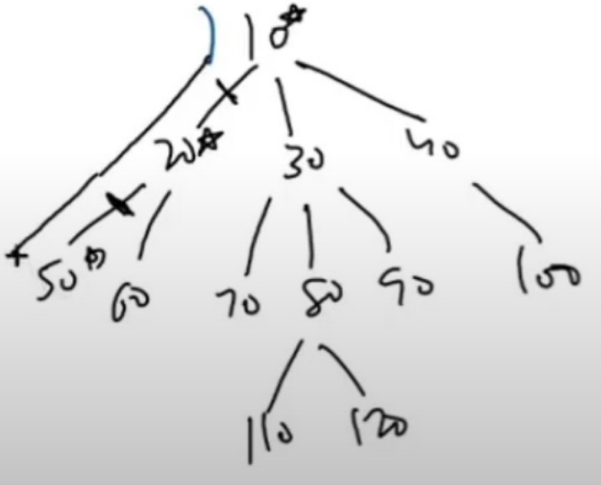
niche jate hue print karate hue jaynge
aur

khudko aur child ko print karaya
next level pe child behj dia

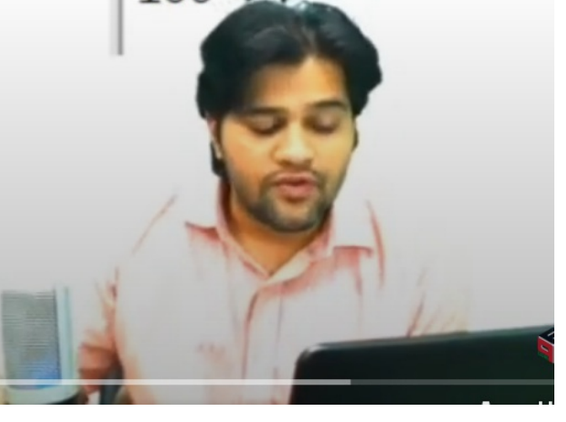
Display a Generic Tree




```java
public static void display(Node node){
    String str = node.data + " -> ";
    for(Node child: node.children){
        str += child.data + ", ";
    }
    str += ".";
    System.out.println(str);

    for(Node child: node.children){
        display(child);
    }
}
```




```
10 -> 20, 30, 40, .
20 -> 50, 60, .
50 -> .
60 -> .
30 -> 70, 80, 90, .
70 -> .
80 -> 110, 120, .
110 -> .
120 -> .
90 -> .
40 -> 100, .
100 -> .
```



Size Of Generic Tree

Question

1. You are given a partially written GenericTree class.
2. You are required to complete the body of size function. The function is expected to count the number of nodes in the tree and return it.
3. Input and Output is managed for you.

Input Format

Input is managed for you

Output Format

Output is managed for you

Constraints

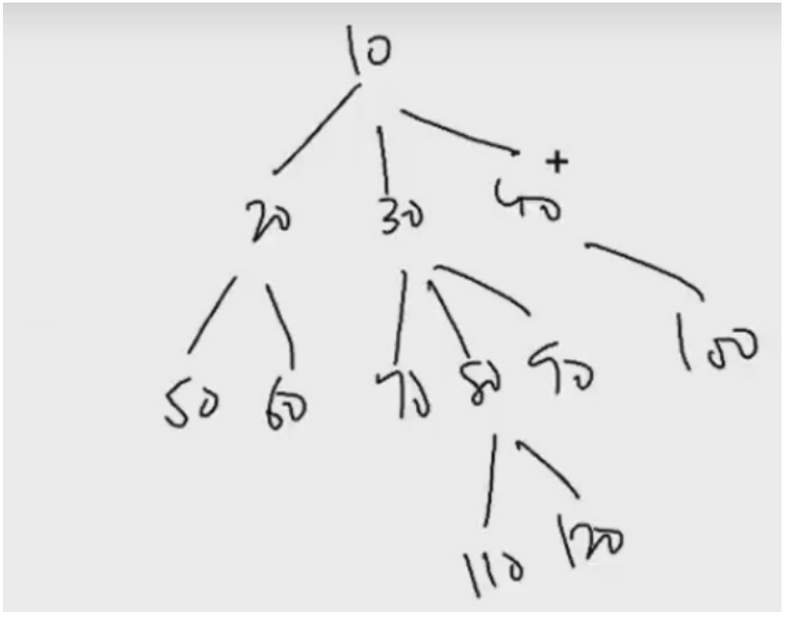
None

Sample Input

```
12
10 20 -1 30 50 -1 60 -1 -1 40 -1 -1
```

Sample Output

```
6
```

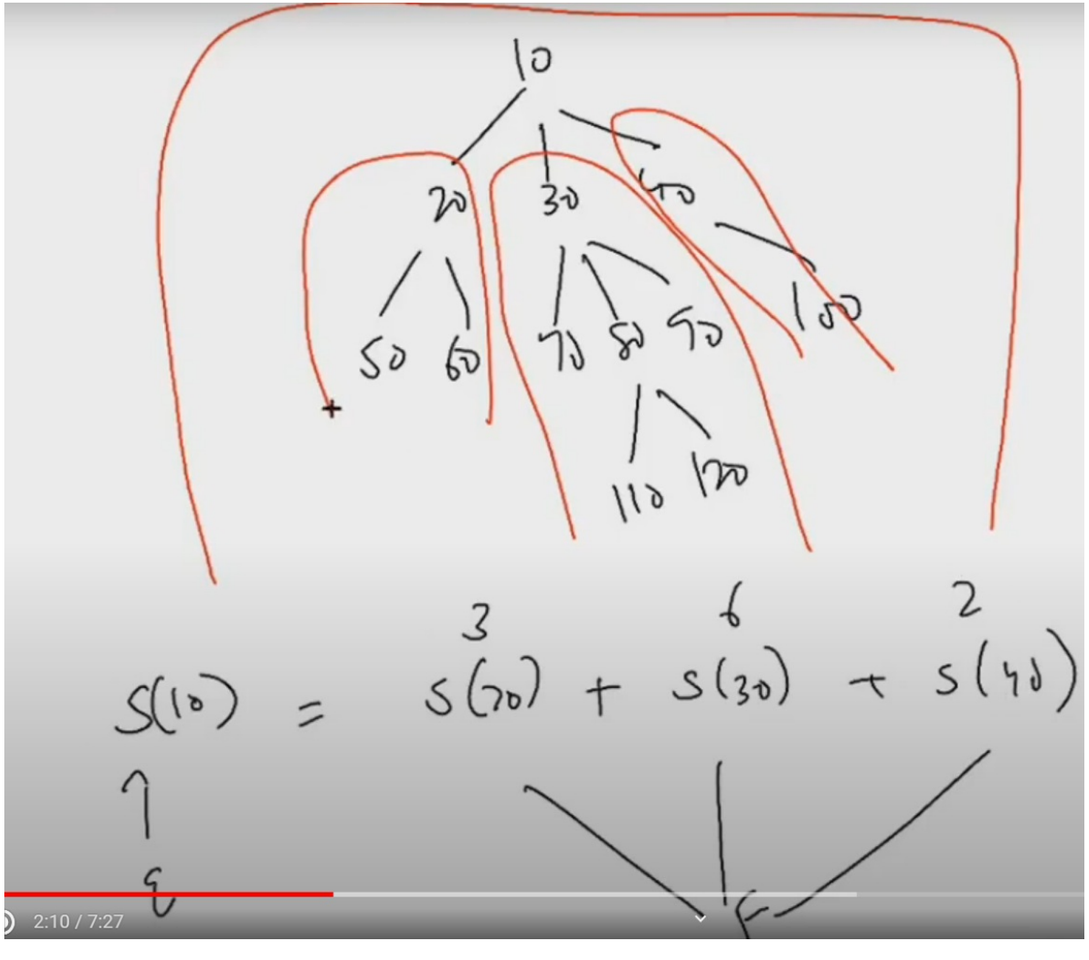


tree dia he

size of nodes
count karne he

level-option
next level pe root k child bej dia
aur jo bhi return hoga use sum me
add kar dia
and end me 1 add kia WHY??vo khud
is consider aas ocunt

```
public static int size(Node node){
  int s = 0;

  for(Node child: node.children){
      int cs = size(child);
      s = s + cs;
  }
  s = s + 1;

  return s;
}
```



EXP
size(10) = 1+ size(20)+size(30)+size(40)

FAITH
faithhe ki size(20)= 3 mil jayega
similary - size(30) = 6 mil jayega
similarly -size(40) = 2 miljayega

usme sirf 1 (yane 10 - vo khud) add hona he

Maximum In A Generic Tree

Question

1. You are given a partially written GenericTree class.
2. You are required to complete the body of max function. The function is expected to find the node with maximum value and return it.
3. Input and Output is managed for you.

Input Format

Input is managed for you

Output Format

Output is managed for you

Constraints

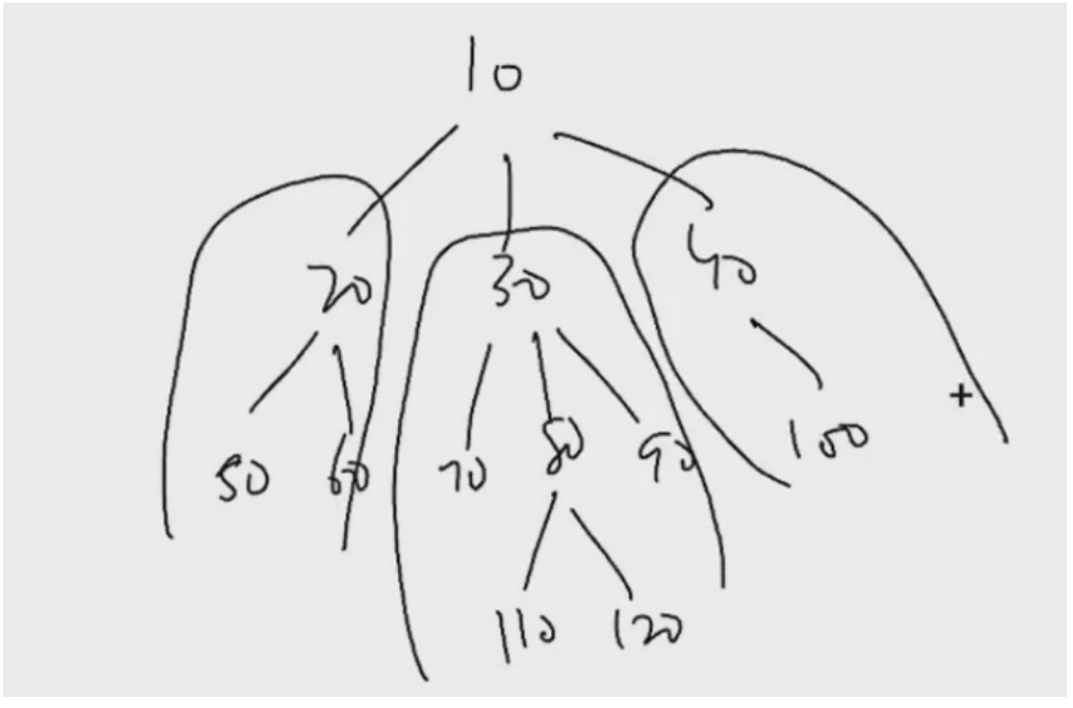
None

Sample Input

```
12
10 20 -1 30 50 -1 60 -1 -1 40 -1 -1
```

Sample Output

```
6
```




```java
public static int max(Node node) {
  int max = Integer.MIN_VALUE;

  for(Node child: node.children){
      int cm = max(child);
      max = Math.max(cm, max);
  }
  max = Math.max(node.data, max);

  return max;
}
```

option-level

next level pe root ke child bhej dia
all child me se max nikala

fir max ko root k sath compare

Height Of A Generic Tree

Question

1. You are given a partially written GenericTree class.
2. You are required to complete the body of height function. The function is expected to find the height of tree. Depth of a node is defined as the number of edges it is away from the root (depth of root is 0). Height of a tree is defined as depth of deepest node.
3. Input and Output is managed for you.

Input Format

Input is managed for you

Output Format

Output is managed for you

Constraints

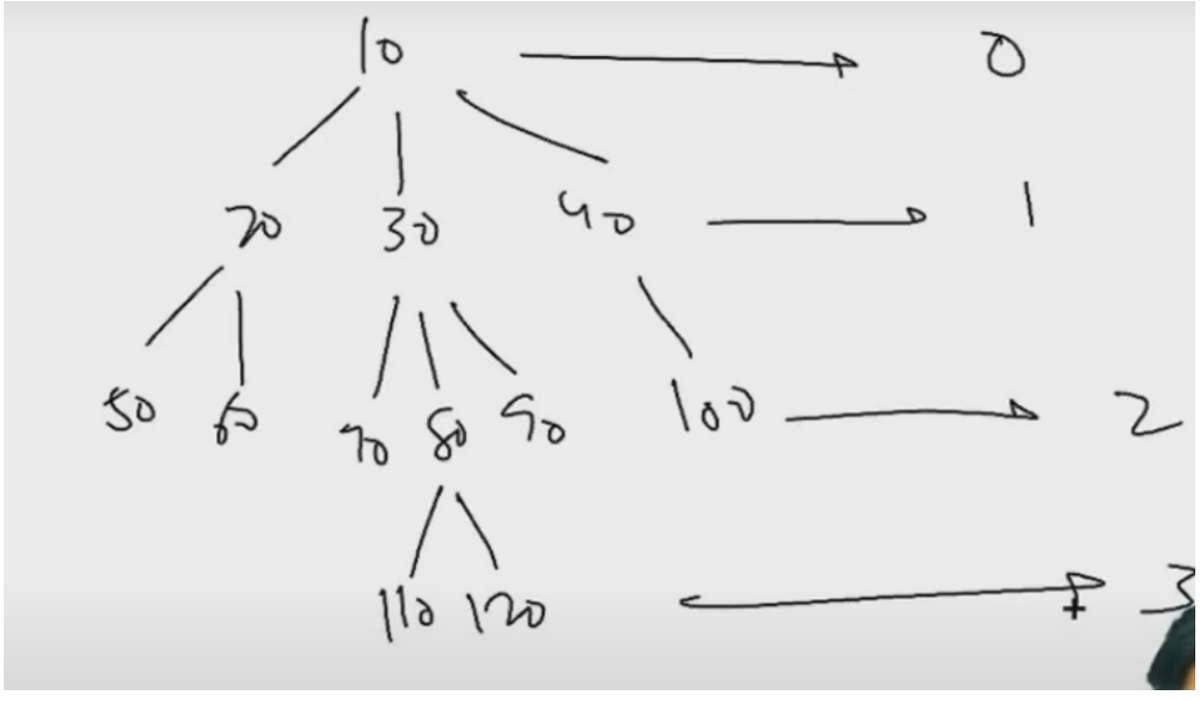
None

Sample Input

```
12
10 20 -1 30 50 -1 60 -1 -1 40 -1 -1
```

Sample Output

```
2
```

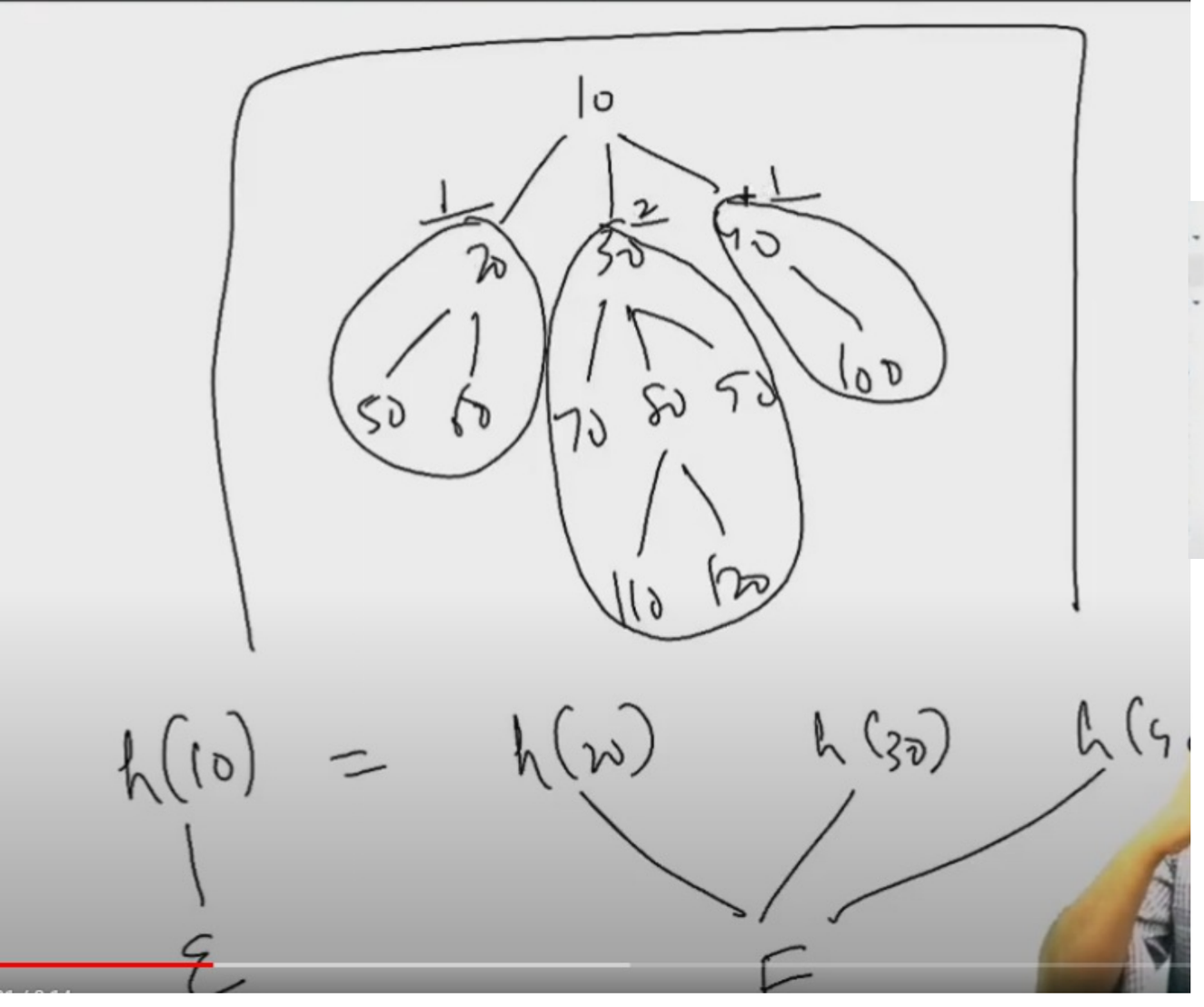


depth: kis node ki depth define ki jati he vo kitne edges door he root se
for eg - 40 - 1 edge door he
100 - 2 edge door he

height of tree =depth of deepest node

node k terms me height - 4 he
edge k terms me height =3 he

upar hummne edge k term me baat kit he
agar **node ke terms me** baat kare toh
height of tree will be = 1+depth of deepest node



ht=-1 for single node to get ans =0;
(getting ans based on edge terms

```
public static int height(Node node) {
  int ht = -1;
  for(Node child: node.children){
      int ch = height(child);
      ht = Math.max(ch, ht);
  }
  ht += 1;

  return ht;
}
```

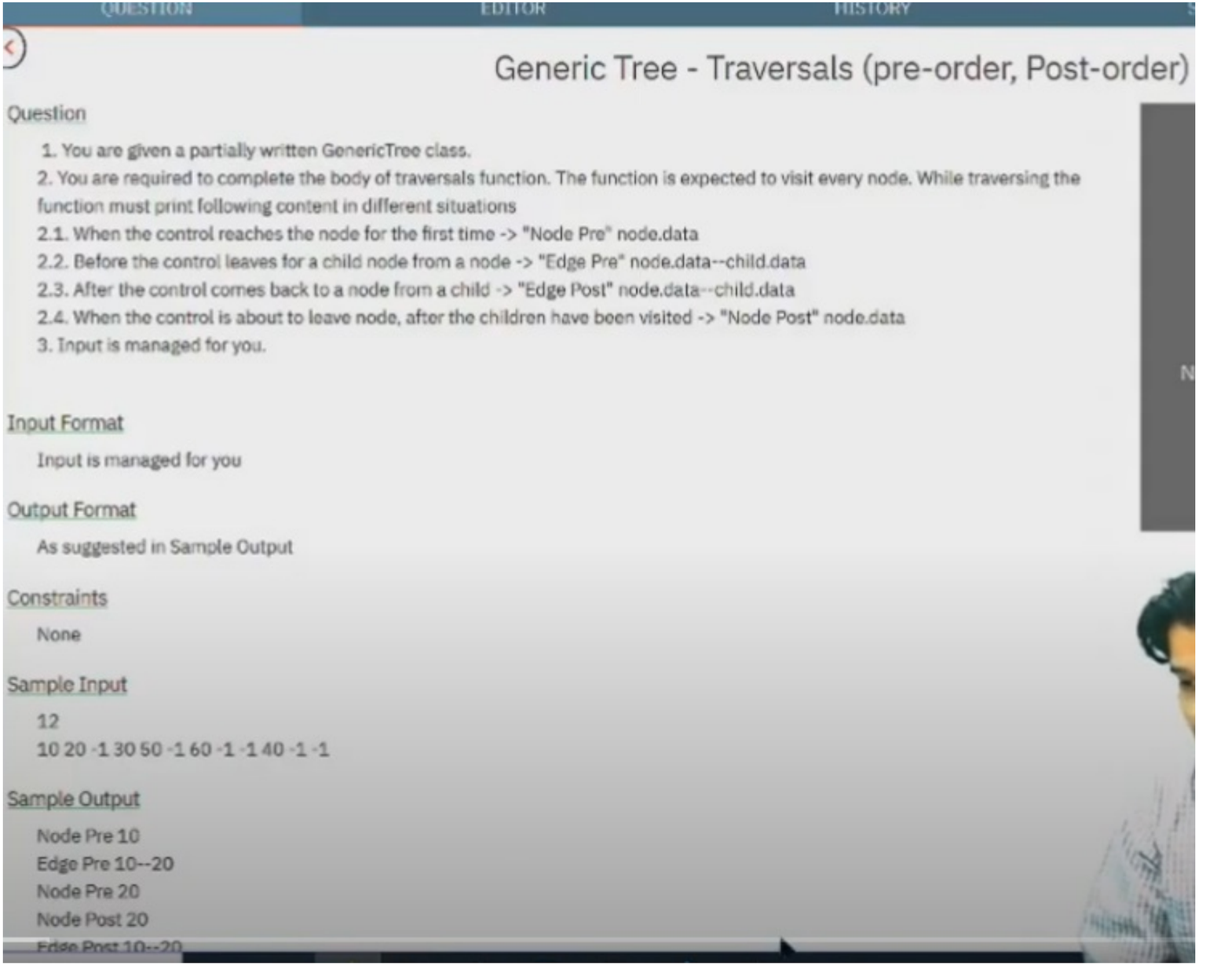
level option

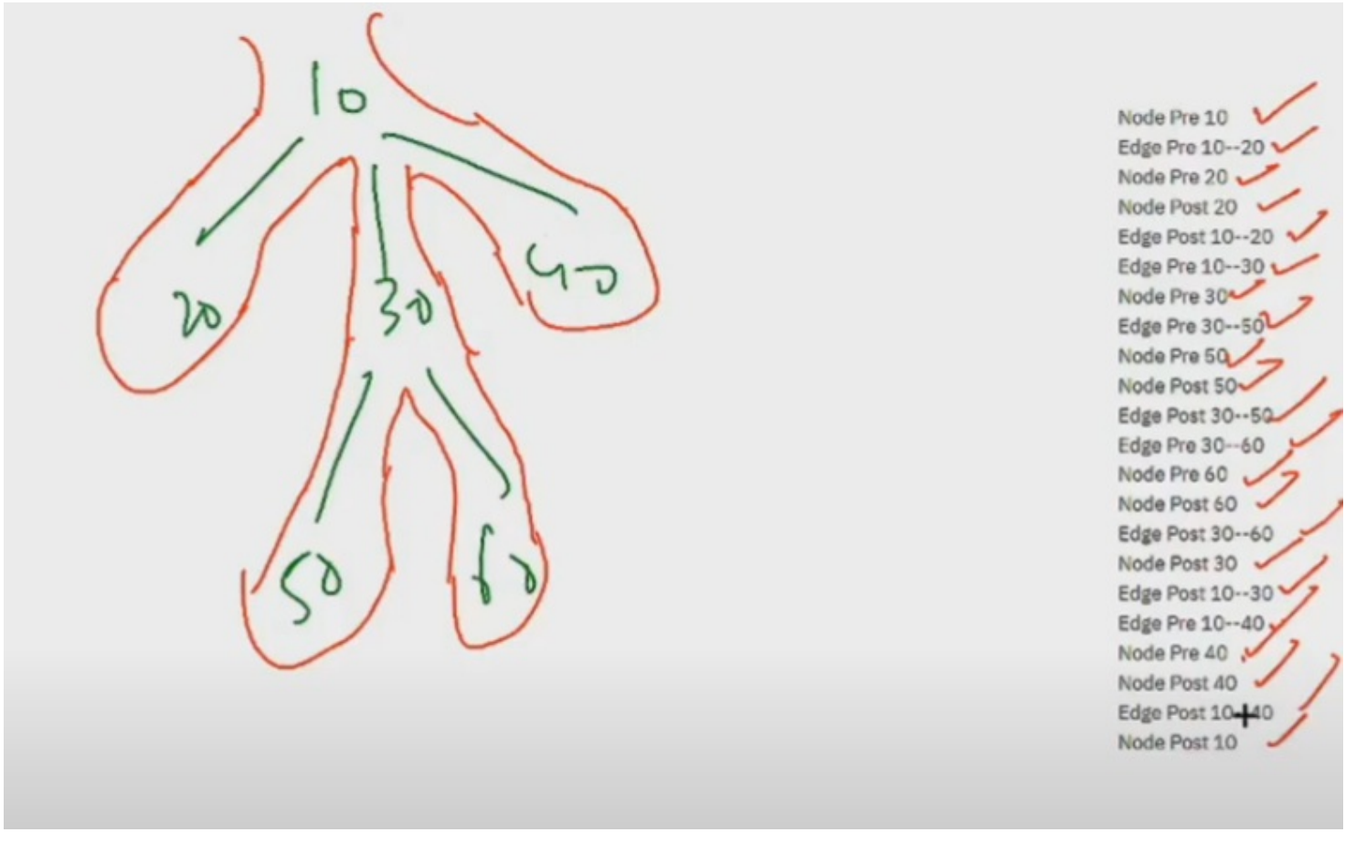
next level pe root ke child bhej dia

vo edge k terms me height denge

current level pe 3 child ka max nikalunga
why? bec uski depth jyada hogi

and then usem 1 add kardunga


QUESTION
EDITOR
HISTORY
Generic Tree - Traversals (pre-order, Post-order)
Question
1. You are given a partially written GenericTree class.
2. You are required to complete the body of traversals function. The function is expected to visit every node. While traversing the function must print following content in different situations
2.1. When the control reaches the node for the first time -> "Node Pre" node.data
2.2. Before the control leaves for a child node from a node -> "Edge Pre" node.data--child.data
2.3. After the control comes back to a node from a child -> "Edge Post" node.data--child.data
2.4. When the control is about to leave node, after the children have been visited -> "Node Post" node.data
3. Input is managed for you.
Input Format
Input is managed for you
Output Format
As suggested in Sample Output
Constraints
None
Sample Input
12
10 20 -1 30 50 -1 60 -1 -1 40 -1 -1
Sample Output
Node Pre 10
Edge Pre 10--20
Node Pre 20
Node Post 20
Edge Post 10--20

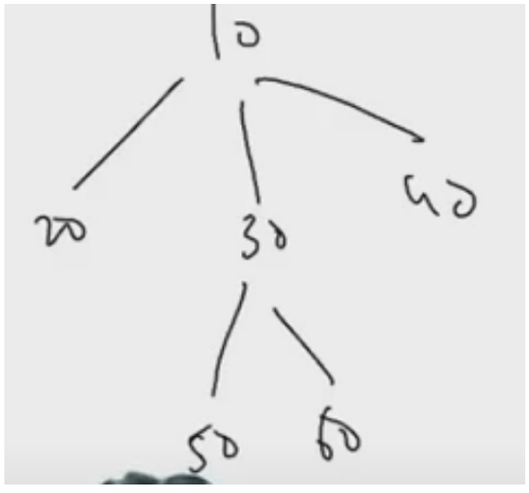


ye karna he

print
Node pre-post
edge pre-post

need to undersand the pattern

jab vo node pe ya edge pe hota he
oth code me kis line pe
correspond karta he



N>C means node k baad child visit hota he

C>N means Child k baad node visit hota he

before recusive call

Pre Order

Euler path → Node's left side

→ Before going deep in recursion

N > C, root is first

Postorder

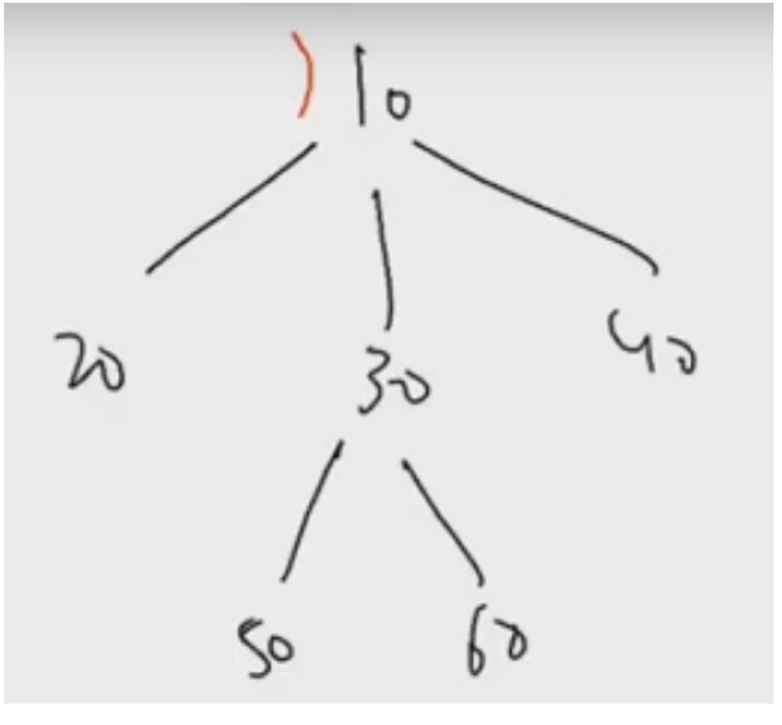
Euler path → Node's right side

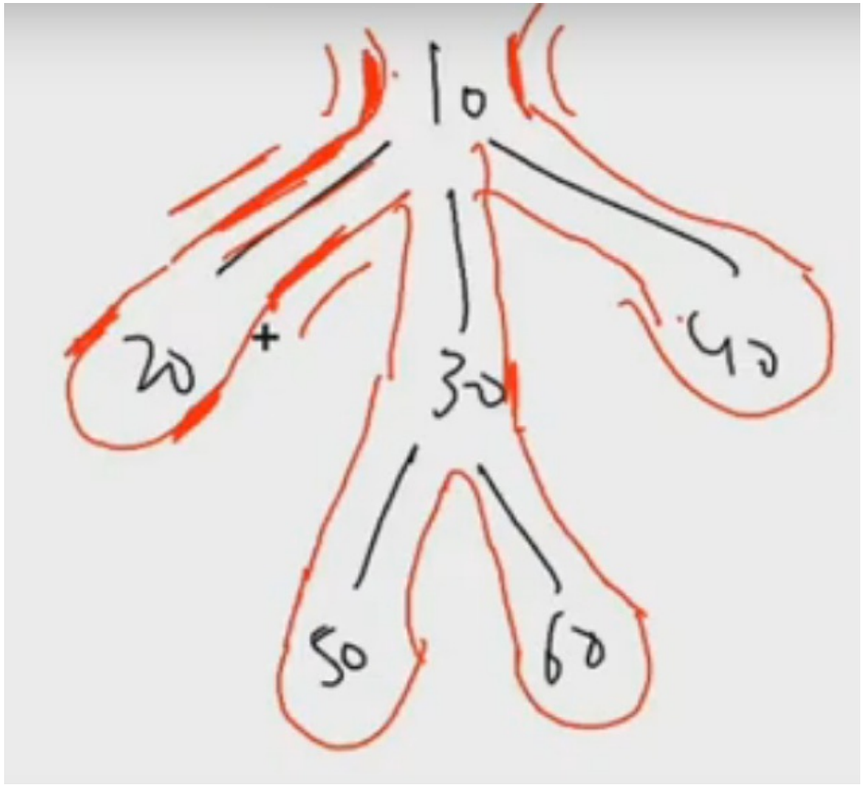
→ while coming out of recursion

C > N, root is last

after recusive call

```java
public static void traversals(Node node){
  // euler's left, on the way deep in the recursion, node's pre area
  System.out.println("Node Pre " + node.data);
  for(Node child: node.children){
      // edge pre
      System.out.println("Edge Pre " + node.data + "--" + child.data);
      traversals(child);
      System.out.println("Edge Post " + node.data + "--" + child.data);
      // edge post
  }
  System.out.println("Node Post " + node.data);
  // euler's right, on the way out of recursion, node's post area
}
```






```
public static void traversals(Node node){
    System.out.println("Node Pre " + node.data);
    for(Node child: node.children){
        System.out.println("Edge Pre " + node.data +
        traversals(child);
        System.out.println("Edge Post " + node.data
    }
    System.out.println("Node Post " + node.data);
}
```

QUESTION EDITOR HISTORY

Level-order Of Generic Tree

Question

1. You are given a partially written GenericTree class.
2. You are required to complete the body of levelorder function. The function is expected to visit every node in "levelorder fashion". Please check the question video for more details.
3. Input is managed for you.

Input Format

Input is managed for you

Output Format

All nodes from left to right (level by level) all separated by a space and ending in a "."

Constraints

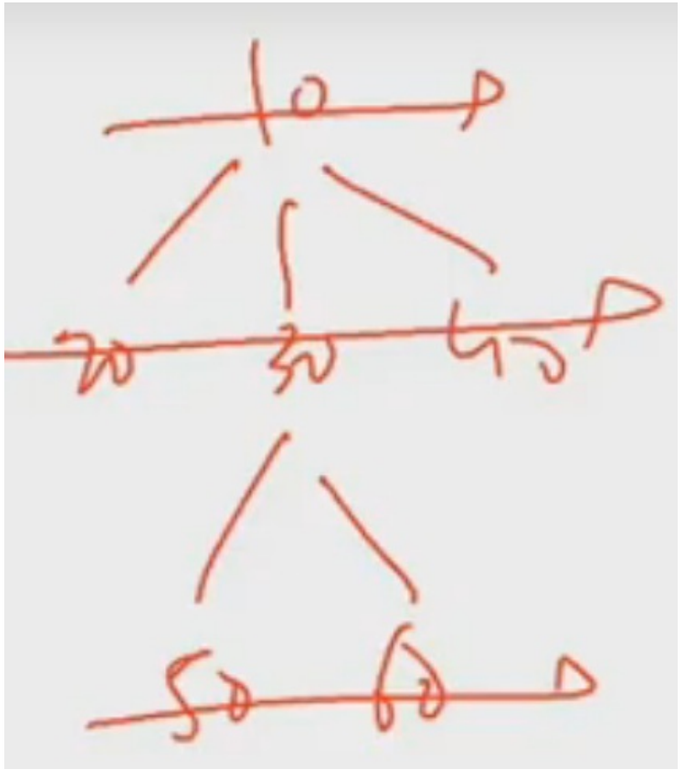
None

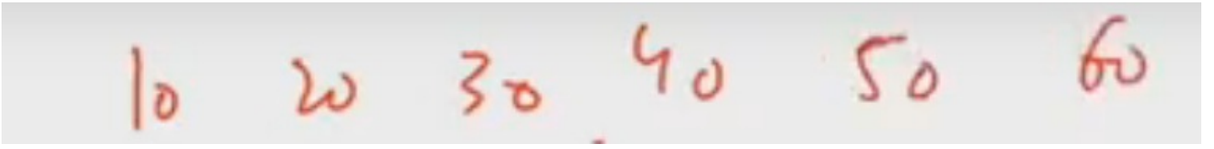
Sample Input

```
24
10 20 50 -1 60 -1 -1 30 70 -1 80 110 -1 120 -1 -1 90 -1 -1 40 100 -1 -1 -1
```

Sample Output

```
10 20 30 40 50 60 70 80 90 100 110 120 .
```



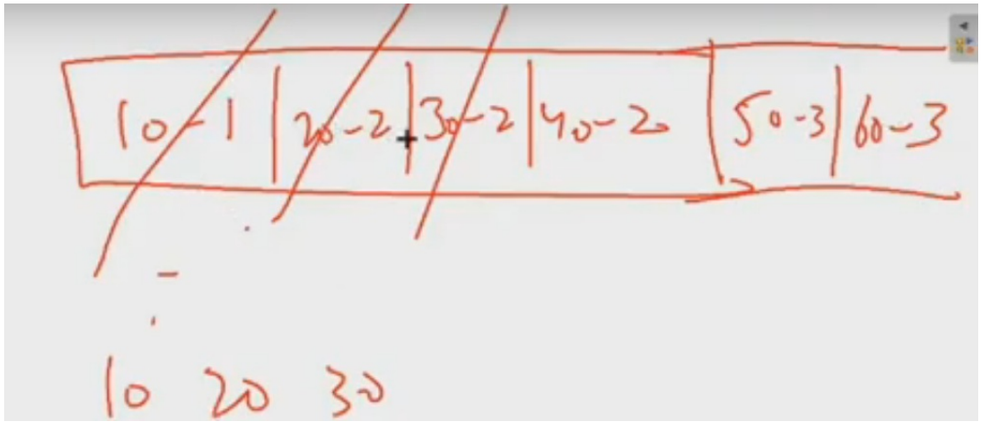


level by level traverse karke print karna he

Use Queue DS

Approach - r p a(remove print add)

1. add first elemtn in queue
2. jab tak size zero nhi hota
remove
print
add child



why r-p-a is working?

bec- queue pehele 1st level add karta he
then 2nd level add karta he
and jab tak 1st level remove hoke print nhi honge
2nd level ki bari nhi ayegi

aur queue last me add karta he
isko hum leverage kar rhe he

```
public static void levelOrder(Node node){
    Queue<Node> q = new ArrayDeque<>();
    q.add(node);

    while(q.size() > 0){
        node = q.remove();
        System.out.print(node.data + " ");

        for(Node child: node.children){
            q.add(child);
        }
    }

    System.out.print(".");
}
```

Levelorder Linewise (generic Tree)

Question

1. You are given a partially written GenericTree class.
2. You are required to complete the body of levelorderLineWise function. The function is expected to visit every node in "levelorder fashion" and print them from left to right (level by level). All nodes on same level should be separated by a space. Different levels should be on separate lines. Please check the question video for more details.
3. Input is managed for you.

Input Format

Input is managed for you

Output Format

All nodes from left to right (level by level) all separated by a space.
All levels on separate lines starting from top to bottom.

Constraints

None

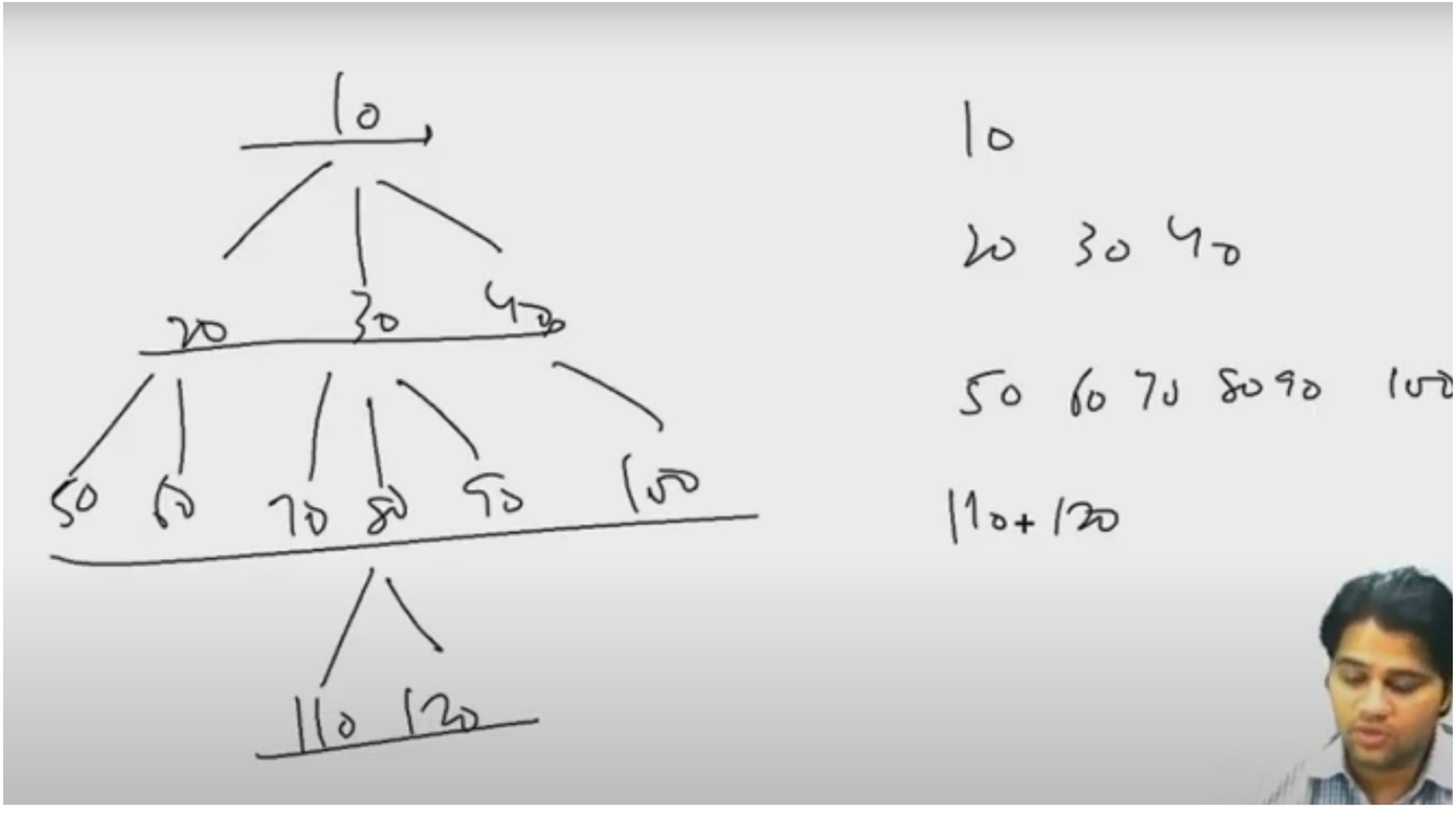
Sample Input

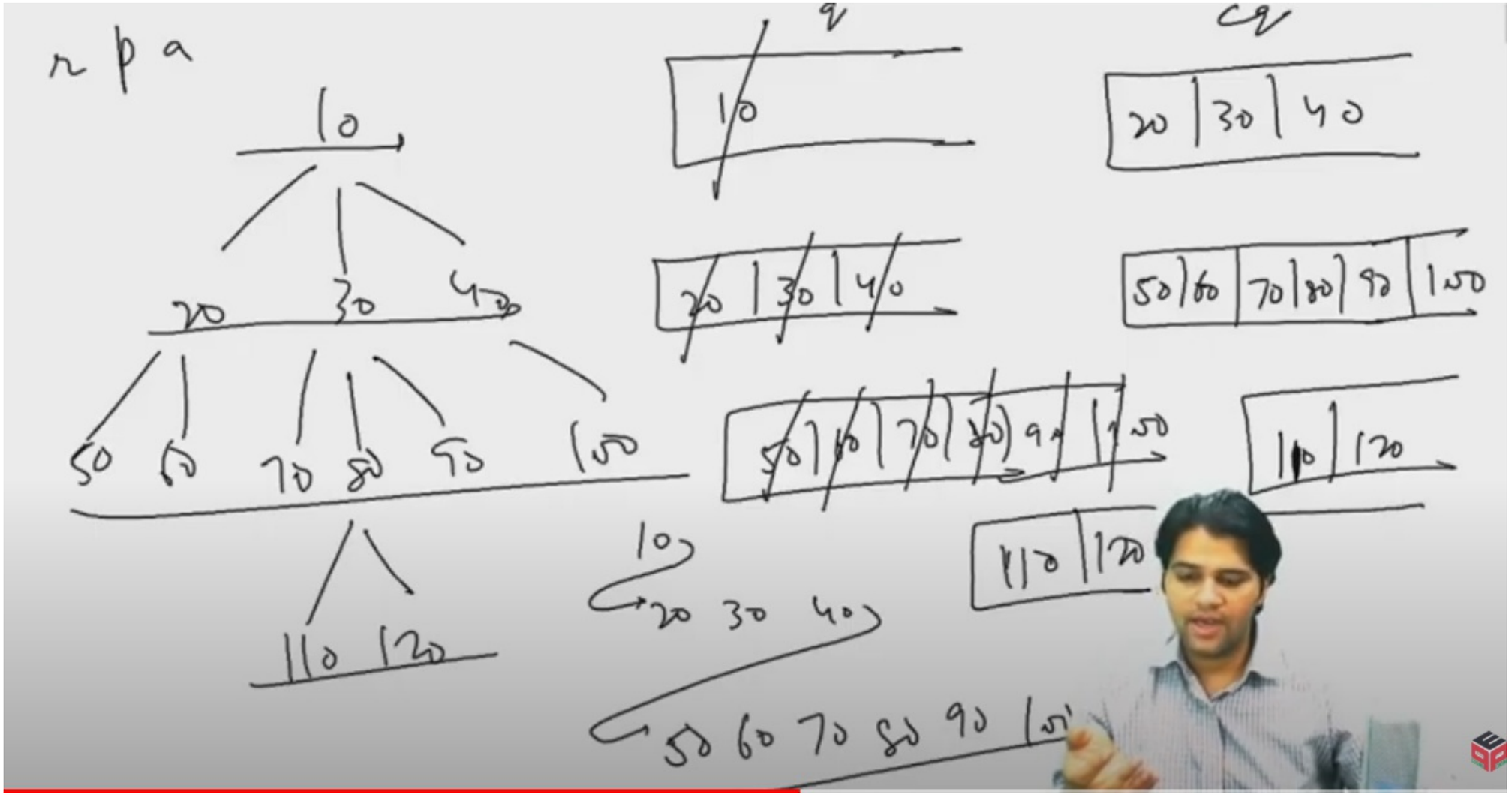
```
24
10 20 50 -1 60 -1 -1 30 70 -1 80 110 -1 120 -1 -1 90 -1 -1 40 100 -1 -1 -1
```

Sample Output

```
10
20 30 40
50 60 70 80 90 100
110 120
```

previous question madhe ekach line madhe print karat hoto ya question madhe next line madhe print karaycha ah e ani left to right





Approach - -2 queue (main queue,child queue)
add fisr node in q
jab tak main queue empty nhi hote
remove from q
print from q
add child in child queue
jab q empty hoga tab
main queue =cq and
cq = new ArrayDequeue() and
\n

```
public static void levelOrderLinewise(Node node){
    Queue<Node> mq = new ArrayDeque<>();
    mq.add(node);

    Queue<Node> cq = new ArrayDeque<>();
    while(mq.size() > 0){
        node = mq.remove();
        System.out.print(node.data + " ");

        for(Node child: node.children){
            cq.add(child);
        }

        if(mq.size() == 0){
            mq = cq;
            cq = new ArrayDeque<>();
            System.out.println();
        }
    }
}
```

QUESTION | EDITOR | HISTORY

Levelorder Linewise Zig Zag

Question

1. You are given a partially written GenericTree class.
2. You are required to complete the body of levelorderLineWiseZZ function. The function is expected to visit every node in "levelorder fashion" but in a zig-zag manner i.e. 1st level should be visited from left to right, 2nd level should be visited from right to left and so on. All nodes on same level should be separated by a space. Different levels should be on separate lines. Please check the question video for more details.
3. Input is managed for you.

Input Format

Input is managed for you

Output Format

All nodes on the same level should be separated by a space.
1st level should be visited left to right, 2nd from right to left and so on alternately.
All levels on separate lines starting from top to bottom.

Constraints

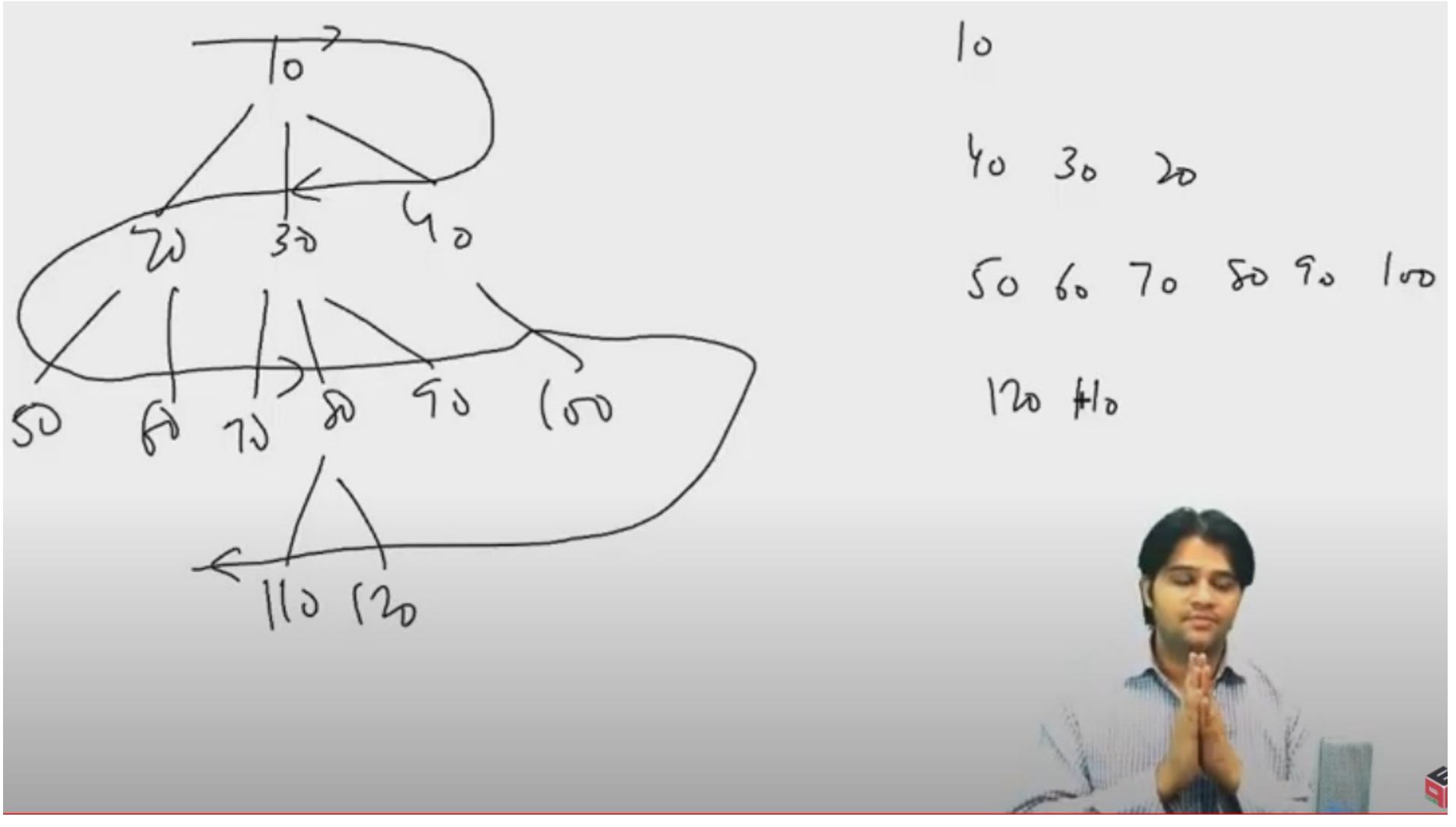
None

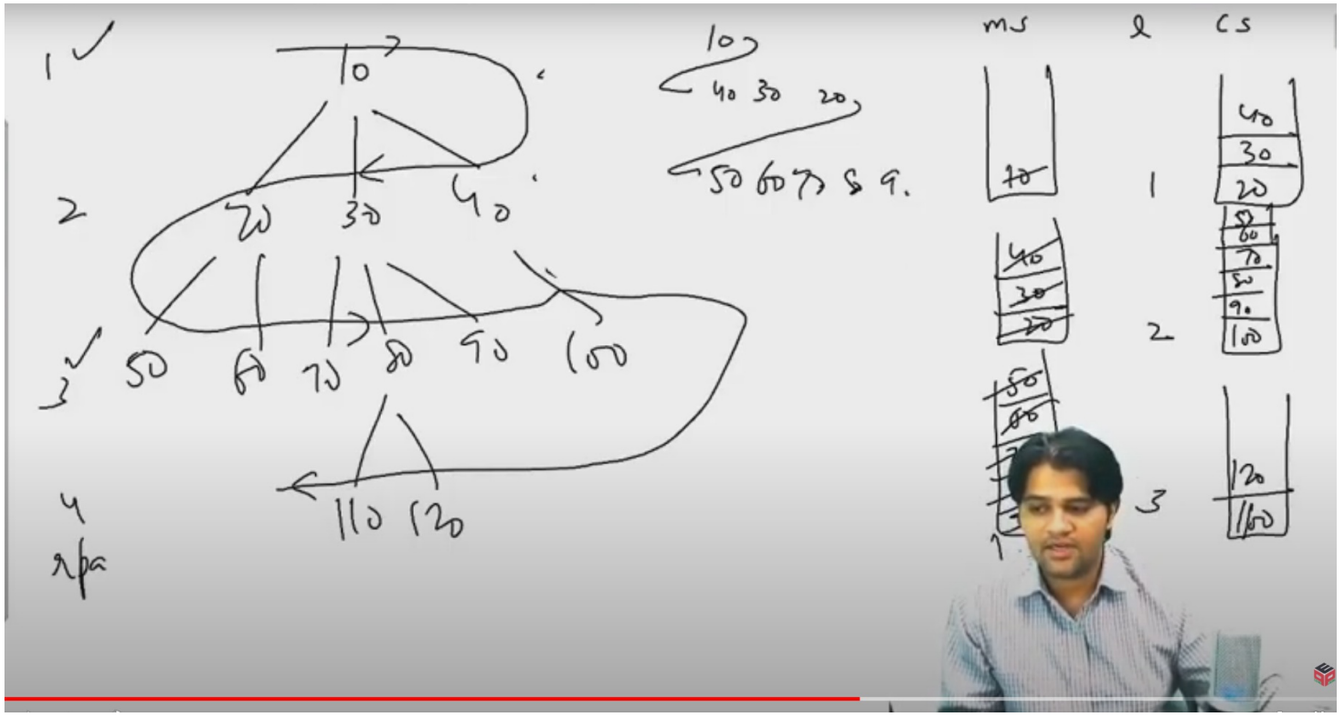
Sample Input

```
24
10 20 50 -1 60 -1 -1 30 70 -1 80 110 -1 120 -1 -1 90 -1 -1 40 100 -1 -1 -1
```

Sample Output

```
10
40 30 20
50 60 70 80 90 100
120 110
```


10
20
30
40
50
60
70
80
90
100
110
120
10
40 30 20
50 60 70 80 90 100
120 110



approach : 2 stack

Approach - -2 stack (main stack,child stack)
add fisr node in ms
jab tak main ms empty nhi hote
remove from ms
print from ms
add child in child stack(from the same direction
jab ms empty hoga tab
ms =child 1stack and
cstack = new stack() and
\n

make sure
jis side se node remove kar rhe
he toh
usi side se child add karna he

```java
public static void levelOrderLinewiseZZ(Node node){
  Stack<Node> ms = new Stack<>();
  ms.push(node);

  Stack<Node> cs = new Stack<>();
  int level = 1;

  while(ms.size() > 0){
      node = ms.pop();
      System.out.print(node.data + " ");

      if(level % 2 == 1){
          for(int i = 0; i < node.children.size(); i++){
              Node child = node.children.get(i);
              cs.push(child);
          }
      } else {
          for(int i = node.children.size() - 1; i >= 0; i
              Node child = node.children.get(i);
              cs.push(child);
          }
      }

      if(ms.size() == 0){
          ms = cs;
          cs = new Stack<>();
          level++;
          System.out.println();
      }
```