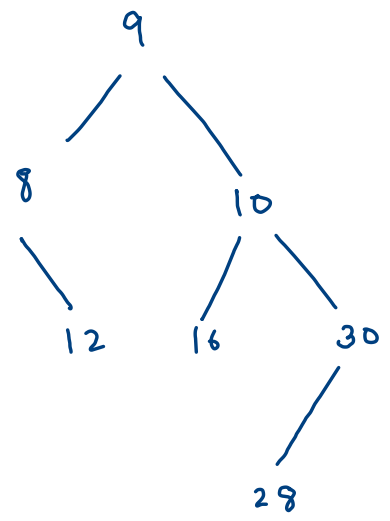
889. Construct Binary Tree from Preorder and Postorder Traversal



Pre : 9 8 12 10 16 30 28

0 1 2 3 4 5 6

Post : 12 8 16 28 30 10 9

idx

N L R (pre)

L R N (post)

pres, pre, pos, poe

9 -> pre[pres]

colse -> idx - pos

Create one of the tree having same preorder & postorder as above tree.

pres+1, pres+colse, pos, idx

pres+colse+1, pre, idx+1, poe-1

Pre :	9	8	12	10	16	30	28
	0	1	2	3	4	5	6
Post :	12	8	16	28	30	10	9

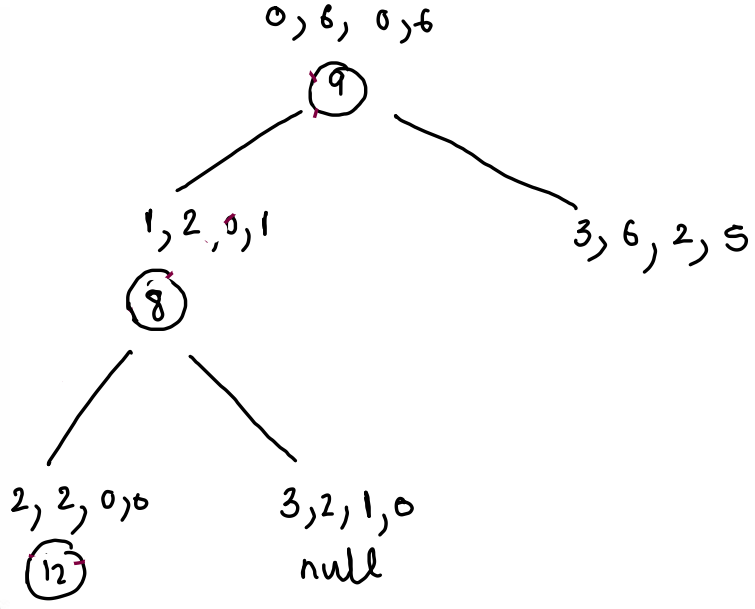
prs, poe, pos, poe

```
if(pos > poe) {
    return null;
}

if(prs == pre) {
    return new TreeNode(preorder[prs]);
}

TreeNode node = new TreeNode(preorder[prs]);

int val = preorder[prs+1];
int idx = -1;

for(int i = pos; i <= poe;i++) {
    if(postorder[i] == val) {
        idx = i;
        break;
    }
}

int cls = idx - pos + 1;

node.left = helper(preorder,prs+1,prs + cls,postorder,pos,idx);
node.right = helper(preorder,prs + cls + 1,pre,postorder,idx+1,poe-1);

return node;
```

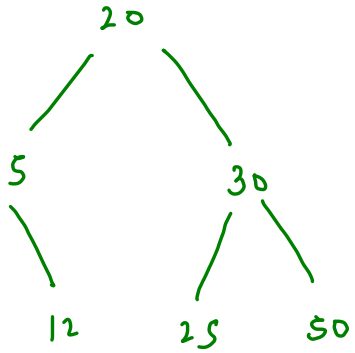
idx = 0

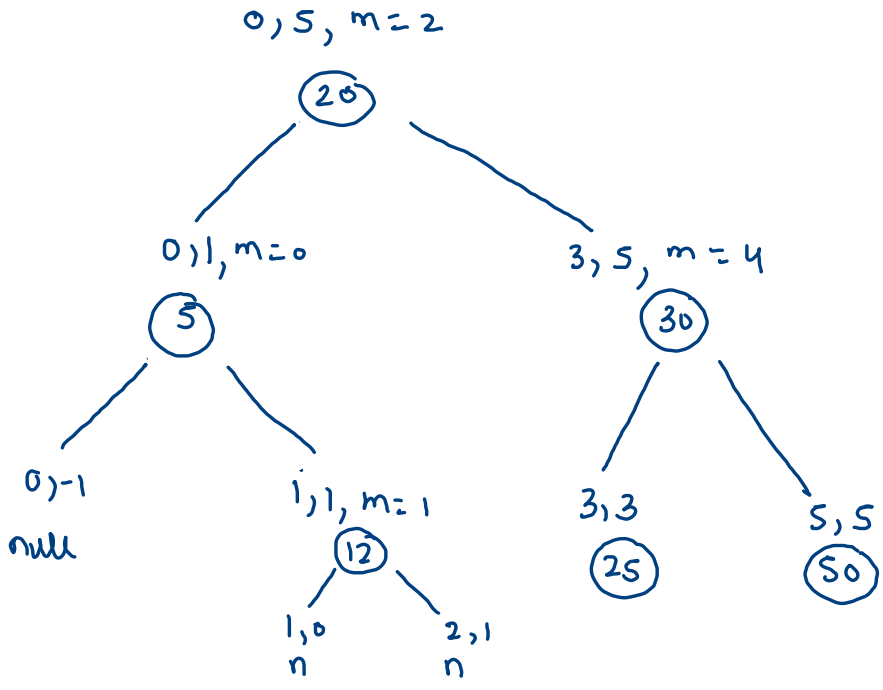
cls = 1

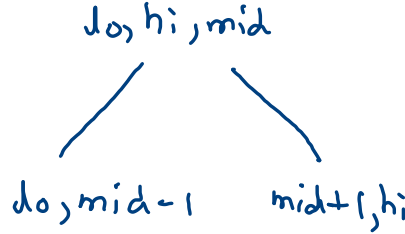
cls = 2



108. Convert Sorted Array to Binary Search Tree



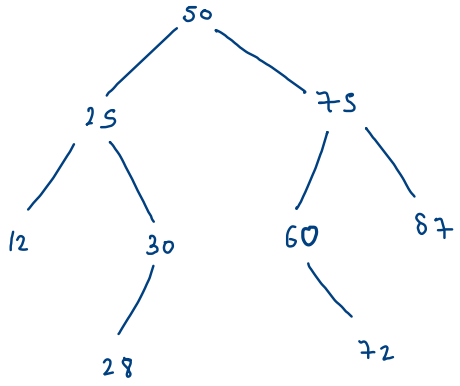


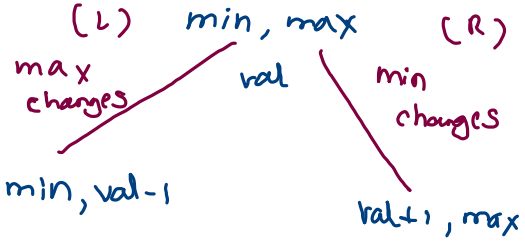


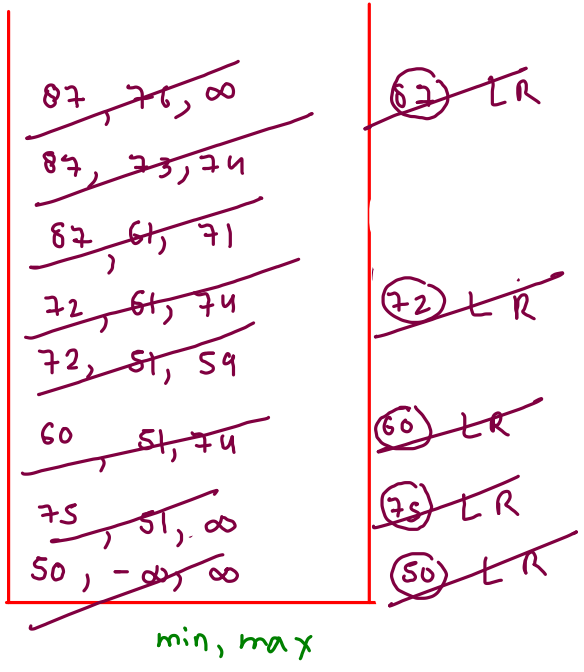
1008. Construct Binary Search Tree from Preorder Traversal

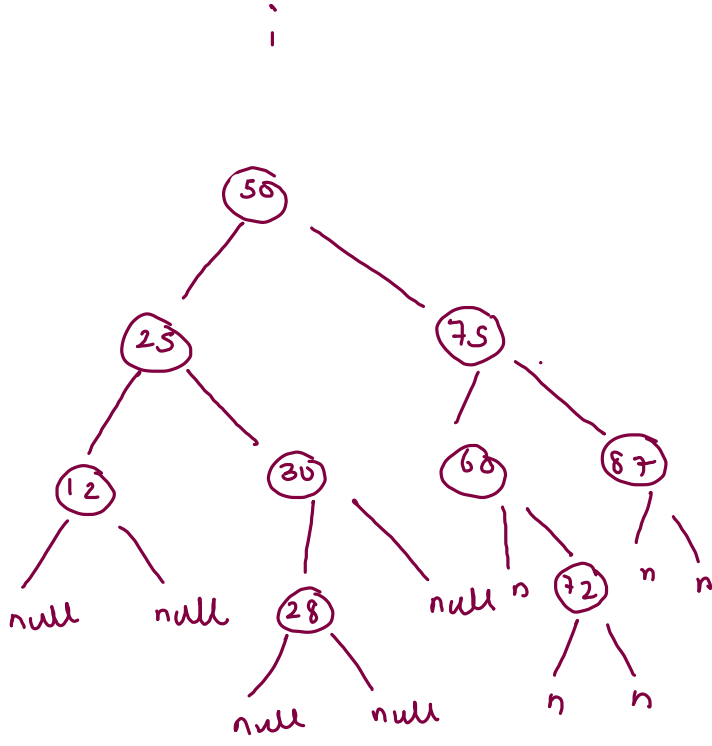
pre : 50 25 12 30 28 75 60 72 87

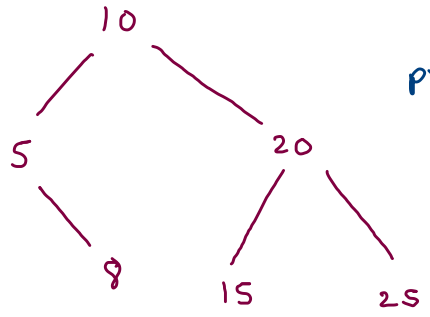
0 1 2 3 4 5 6 7 8







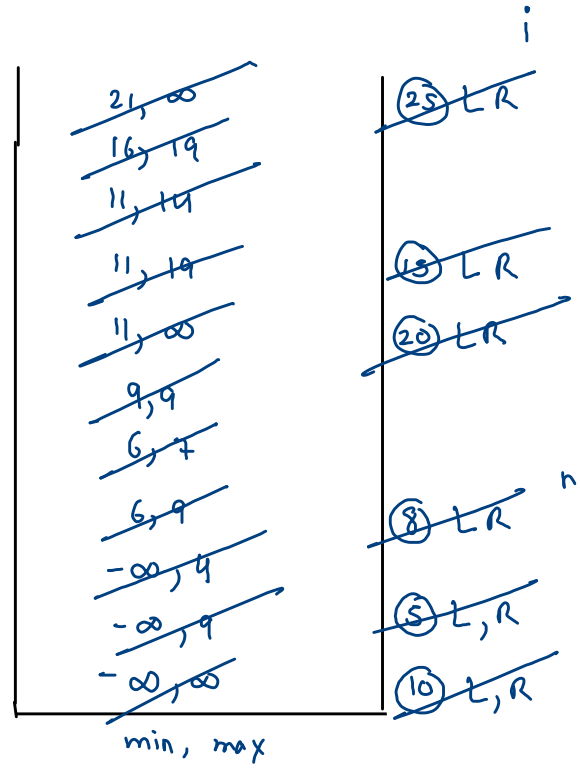


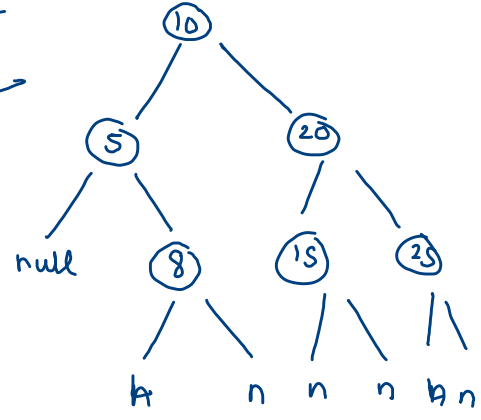


pre : 10 5 8 20 15 25

0 1 2 3 4 5

```
int idx;
public TreeNode bstFromPreorder(int[] preorder) {
    idx = 0;
    return helper(preorder,Integer.MIN_VALUE,Integer.MAX_VALUE);
}

public TreeNode helper(int[]preorder,int min,int max) {
    if(idx >= preorder.length || preorder[idx] < min || preorder[idx] > max) {
        return null;
    }
    else {
        TreeNode node = new TreeNode(preorder[idx]);
        idx++;

        node.left = helper(preorder,min,node.val-1);
        node.right = helper(preorder,node.val + 1,max);

        return node;
    }
}
```



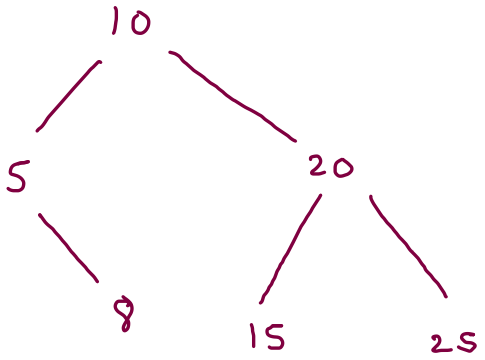


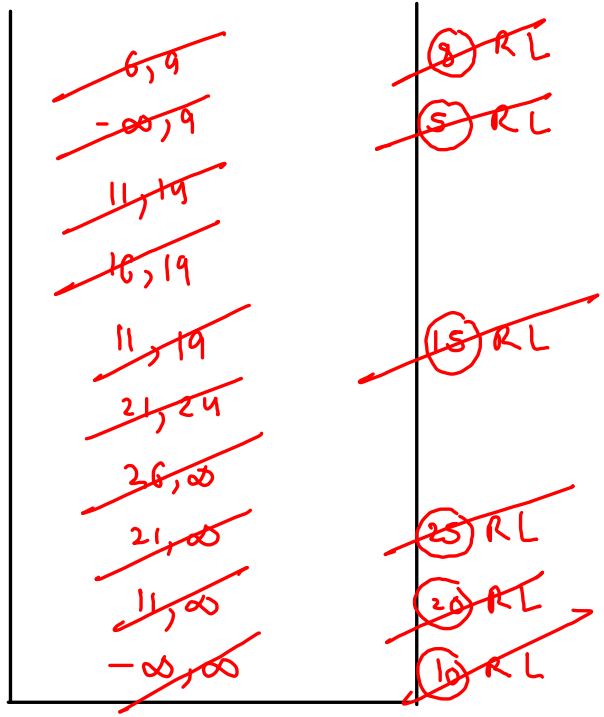
Construct Bst From Postorder Traversal

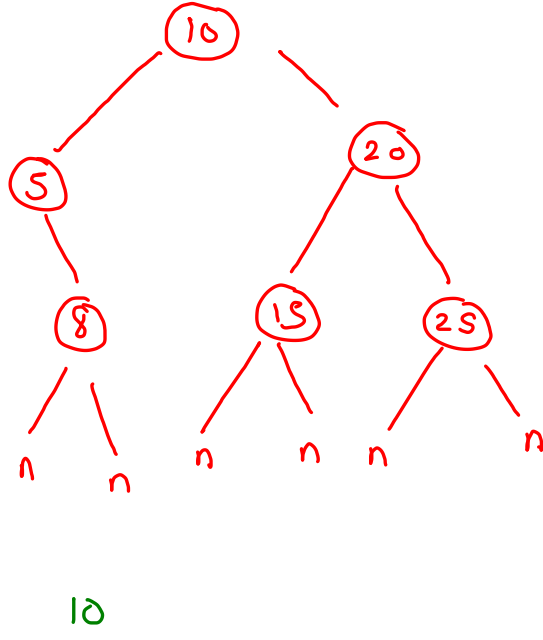
post : 8 5 15 25 20 10

0 1 2 3 4 5

i

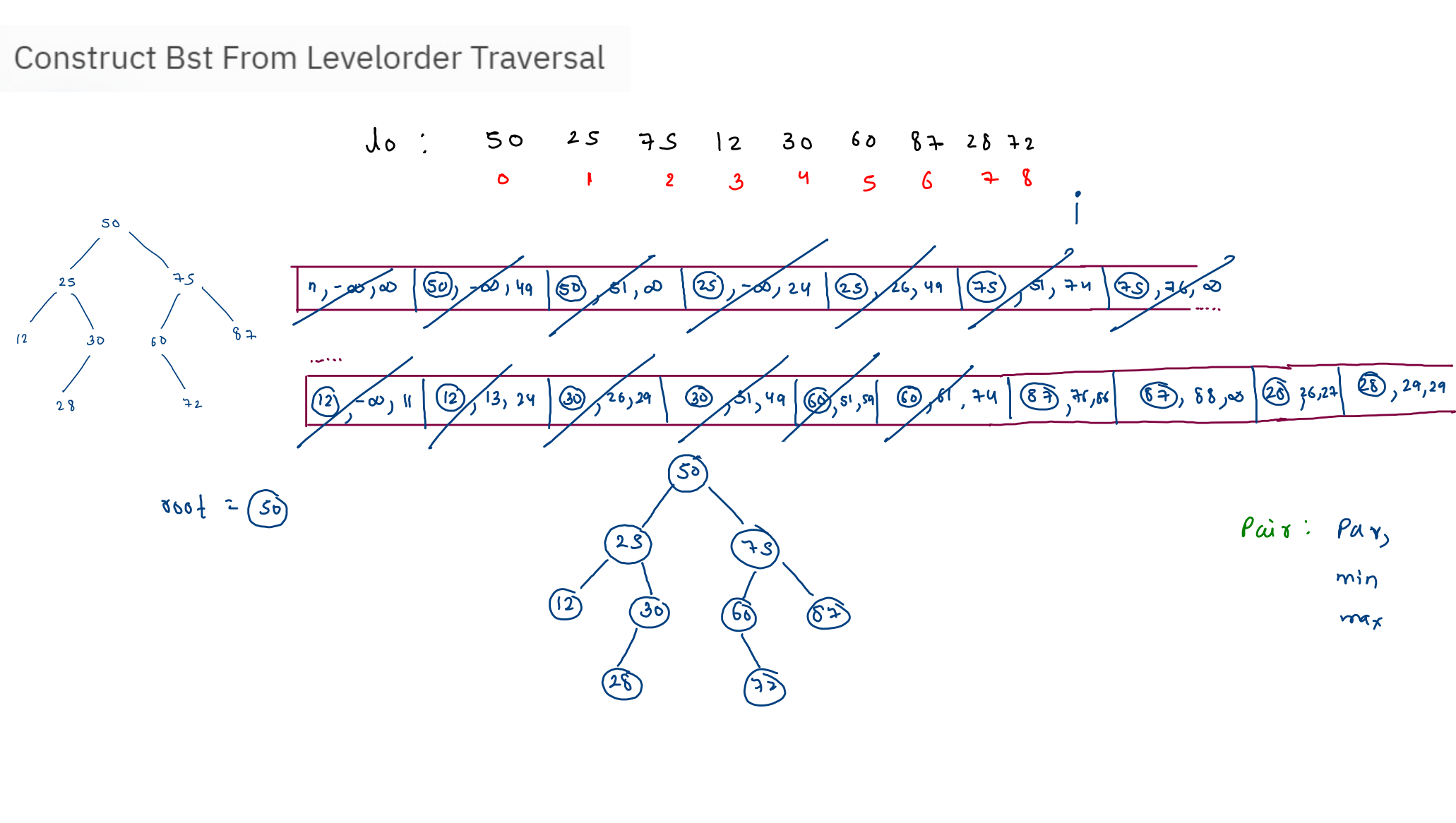


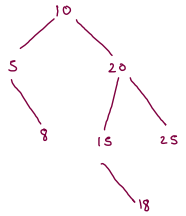




10

Construct Bst From Levelorder Traversal






```java
public Node constructBST(int[] arr)
{
    ArrayDeque<Pair>q = new ArrayDeque<>();
    q.add(new Pair(null,Integer.MIN_VALUE,Integer.MAX_VALUE));

    Node root = null;
    int idx = 0;

    while(q.size() > 0 && idx < arr.length) {
        Pair rem = q.remove();

        if(arr[idx] < rem.min || arr[idx] > rem.max) {
            continue;
        }

        Node node = new Node(arr[idx]);
        idx++;

        if(rem.par == null) {
            root = node;
        }
        else {
            if(rem.par.data > node.data) {
                rem.par.left = node;
            }
            else {
                rem.par.right = node;
            }
        }

        q.add(new Pair(node,rem.min,node.data-1));
        q.add(new Pair(node,node.data + 1,rem.max));

    }
```