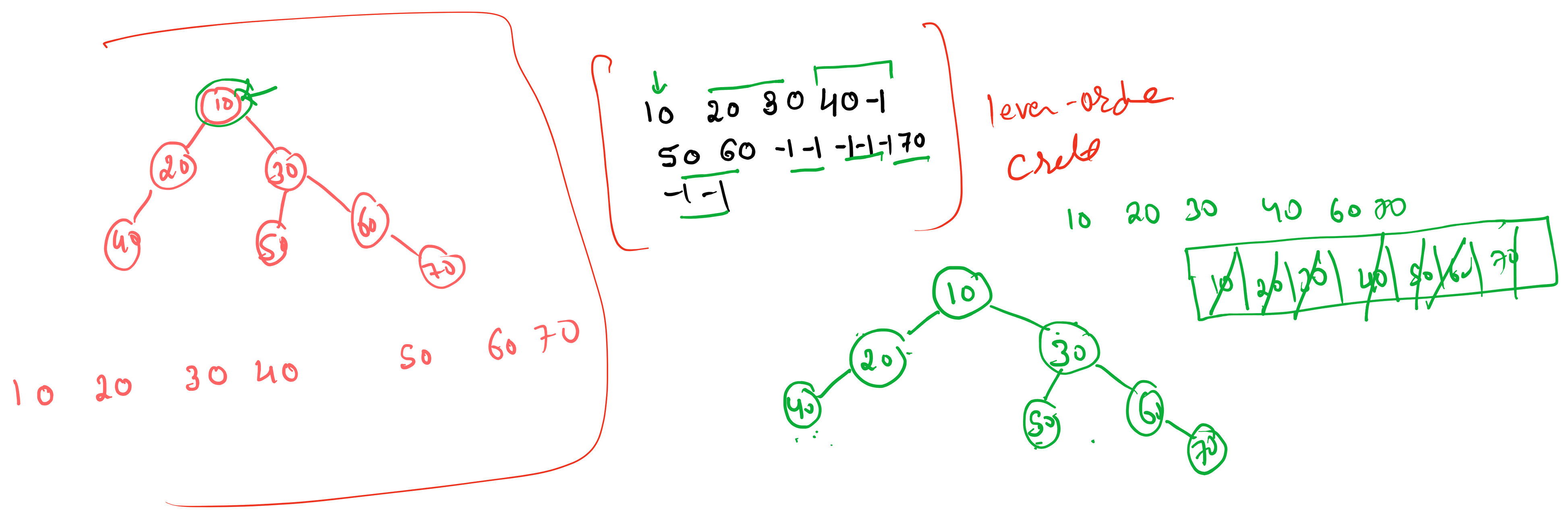
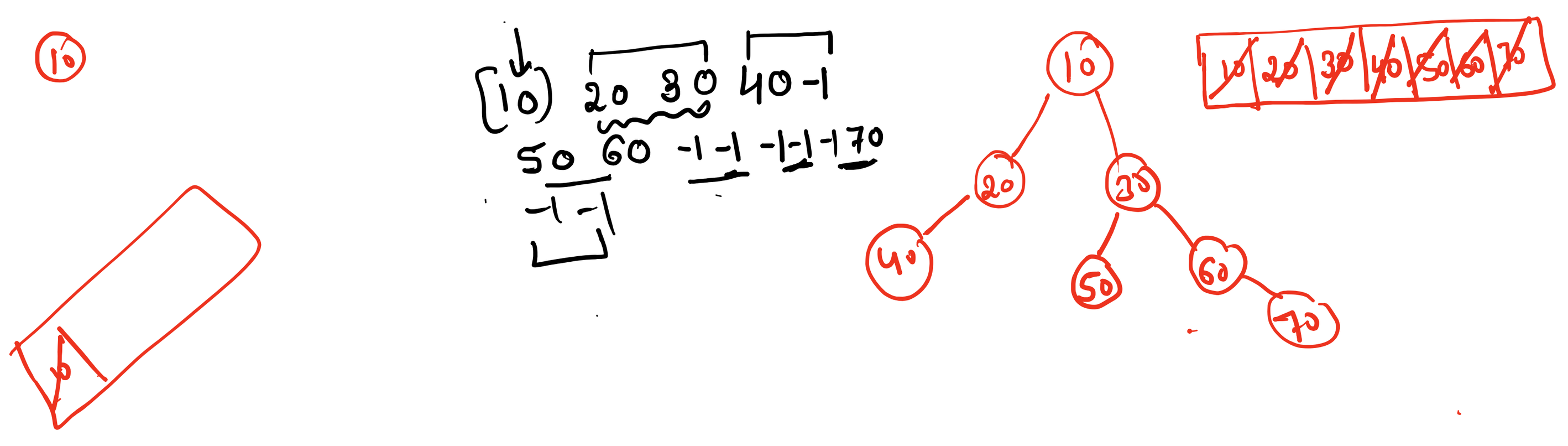
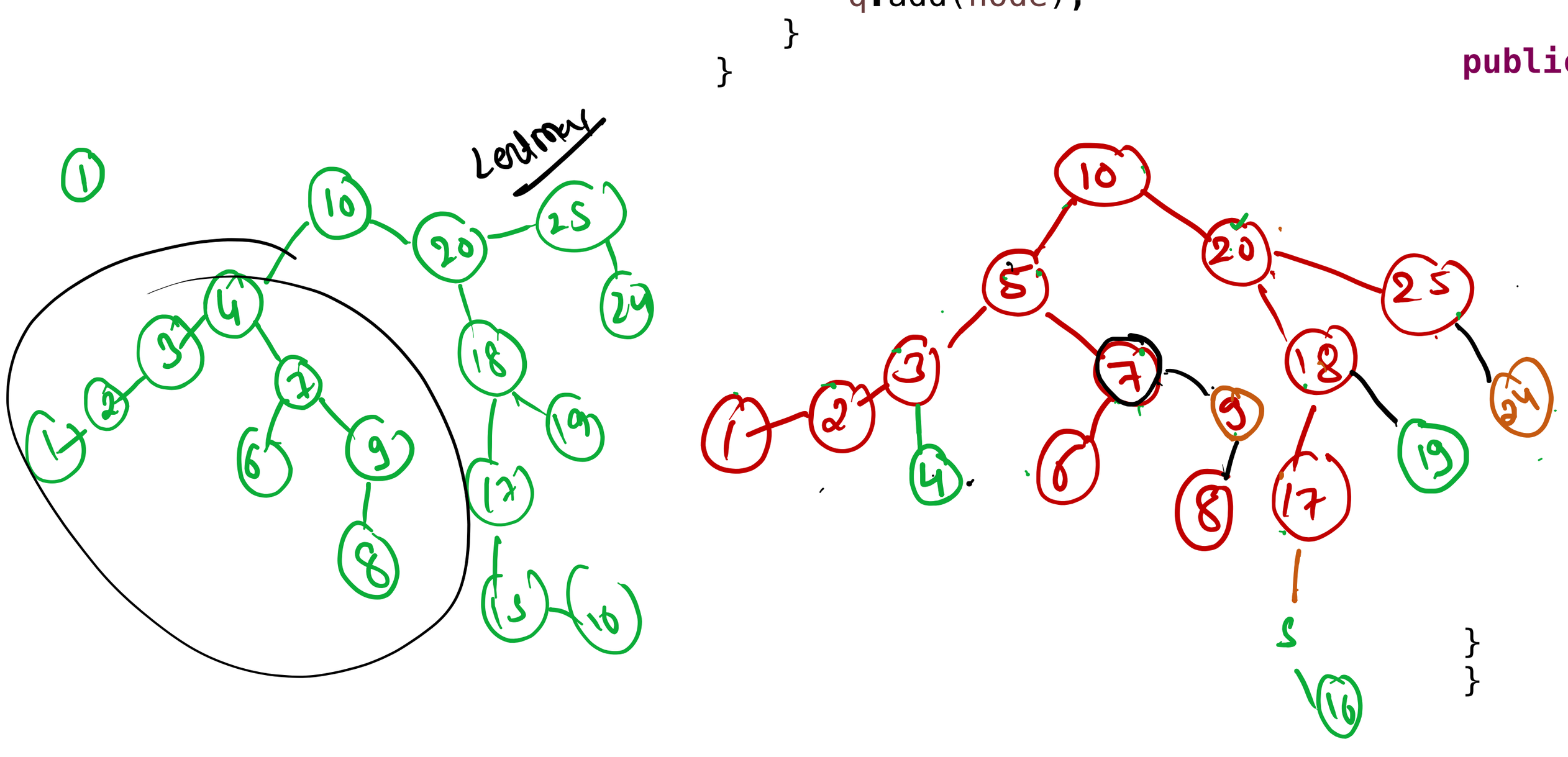
Lec-44

24/05/24 7:43 PM

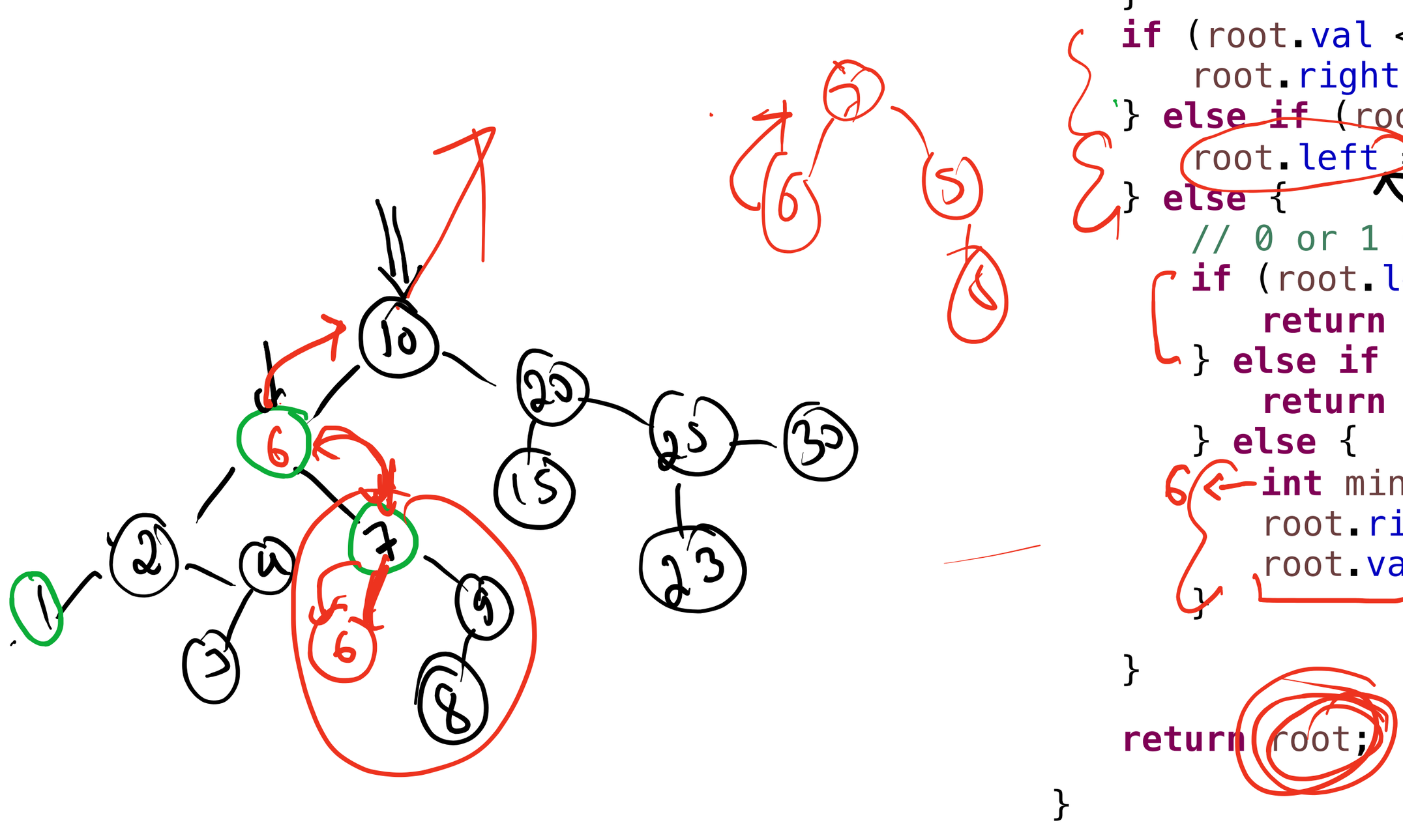




```java
private void createTree() {
        // TODO Auto-generated method stub
        int v = sc.nextInt();
        Node nn = new Node();
        nn.val = v;
        root = nn;
        Queue<Node> q = new LinkedList<>();
        q.add(nn);
        while (!q.isEmpty()) {
            Node rn = q.poll();
            int c1 = sc.nextInt();
            int c2 = sc.nextInt();
            if (c1 != -1) {
                Node node = new Node();
                node.val = c1;
                rn.left = node;
                q.add(node);

            }
            if (c2 != -1) {
                Node node = new Node();
                node.val = c2;
                rn.right = node;
                q.add(node);
            }
        }
```



```java
public TreeNode deleteNode(TreeNode root, int key) {
            if (root == null) {
                return null;
            }
            if (root.val < key) {
                root.right = deleteNode(root.right, key);
            } else if (root.val > key) {
                root.left = deleteNode(root.left, key);
            } else {
                // 0 or 1 Child
                if (root.left == null) {
                    return root.right;
                } else if (root.right == null) {
                    return root.left;
                }
}
}
```

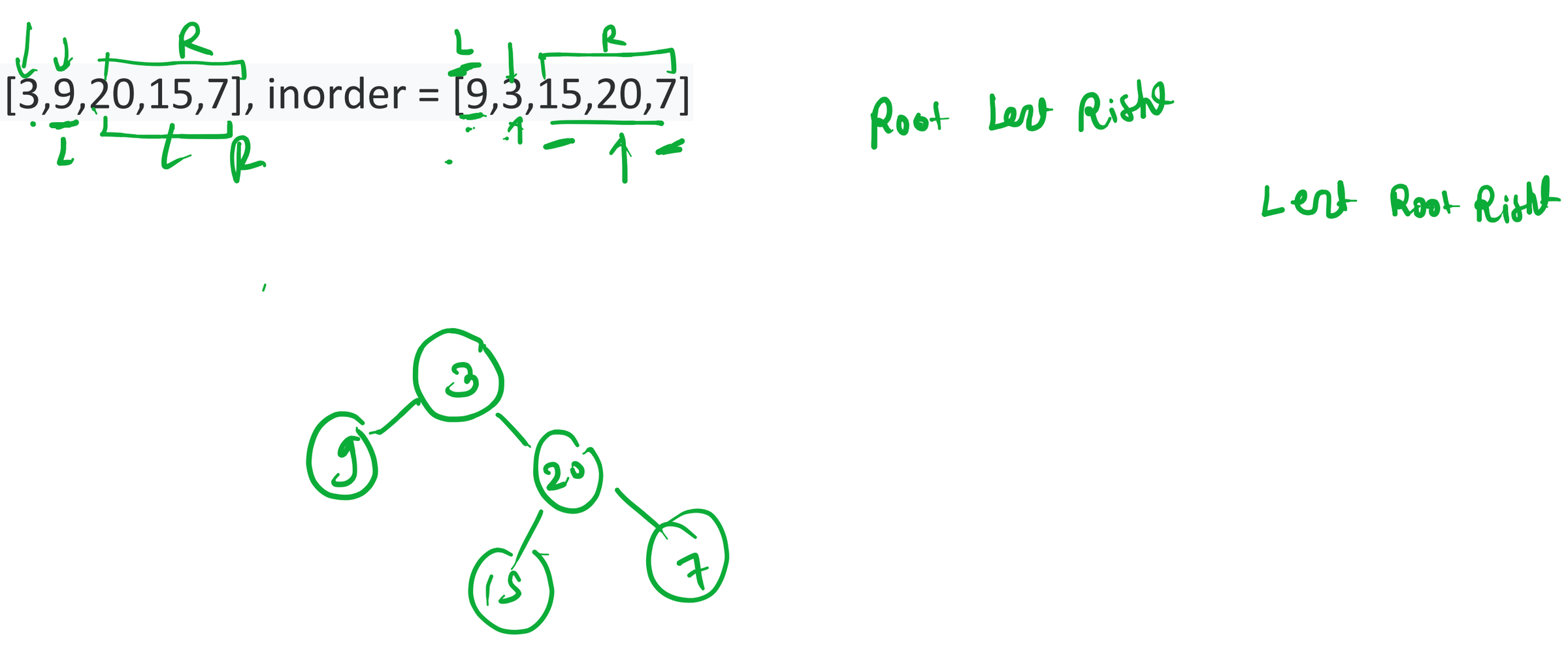


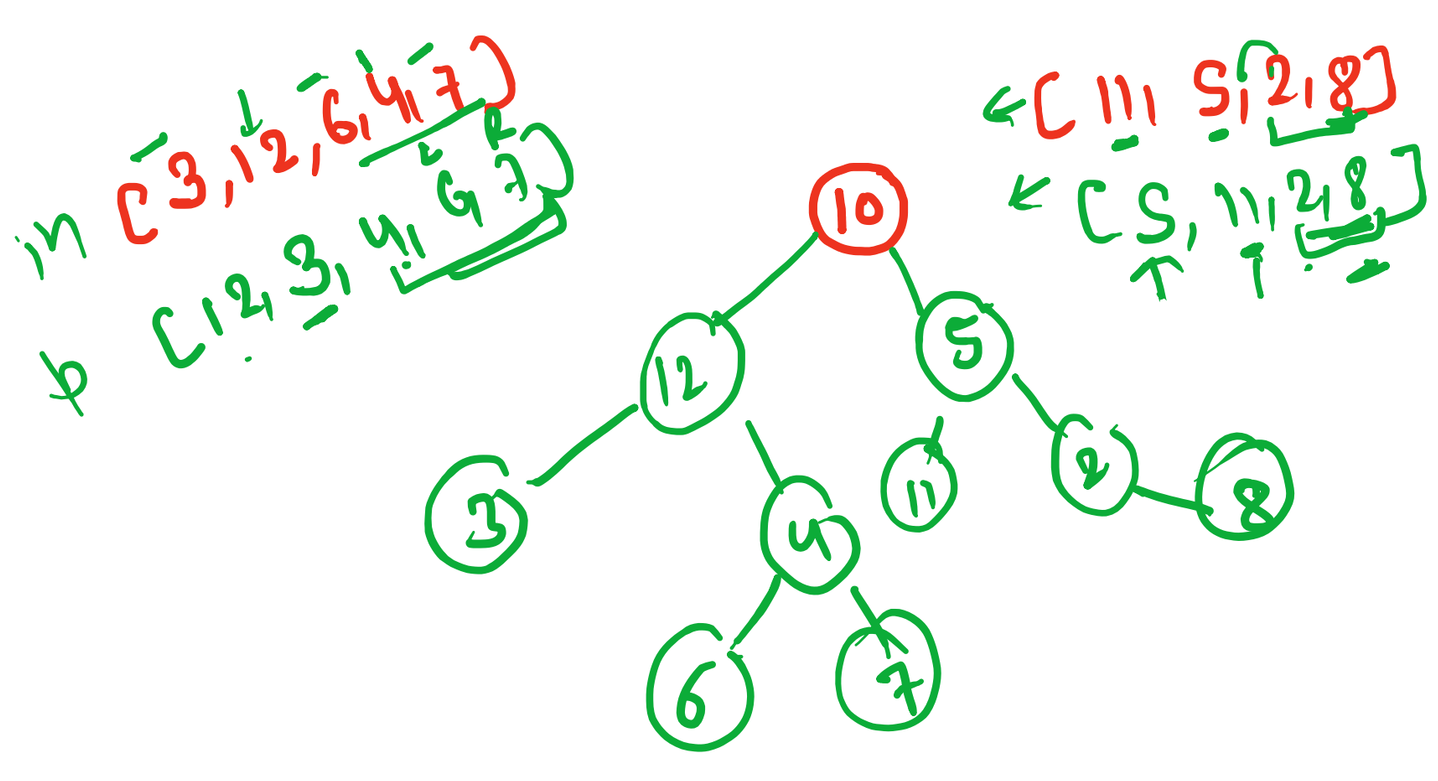
```java
public TreeNode deleteNode(TreeNode root, int key) {
            if (root == null) {
                return null;
            }
            if (root.val < key) {
                root.right = deleteNode(root.right, key);
            } else if (root.val > key) {
                root.left = deleteNode(root.left, key);
            } else {
                // 0 or 1 Child
                if (root.left == null) {
                    return root.right;
                } else if (root.right == null) {
                    return root.left;
                } else {
                    int min = min(root.right);
                    root.right = deleteNode(root.right, min);
                    root.val = min;
                }

            }

            return root;

        }
```

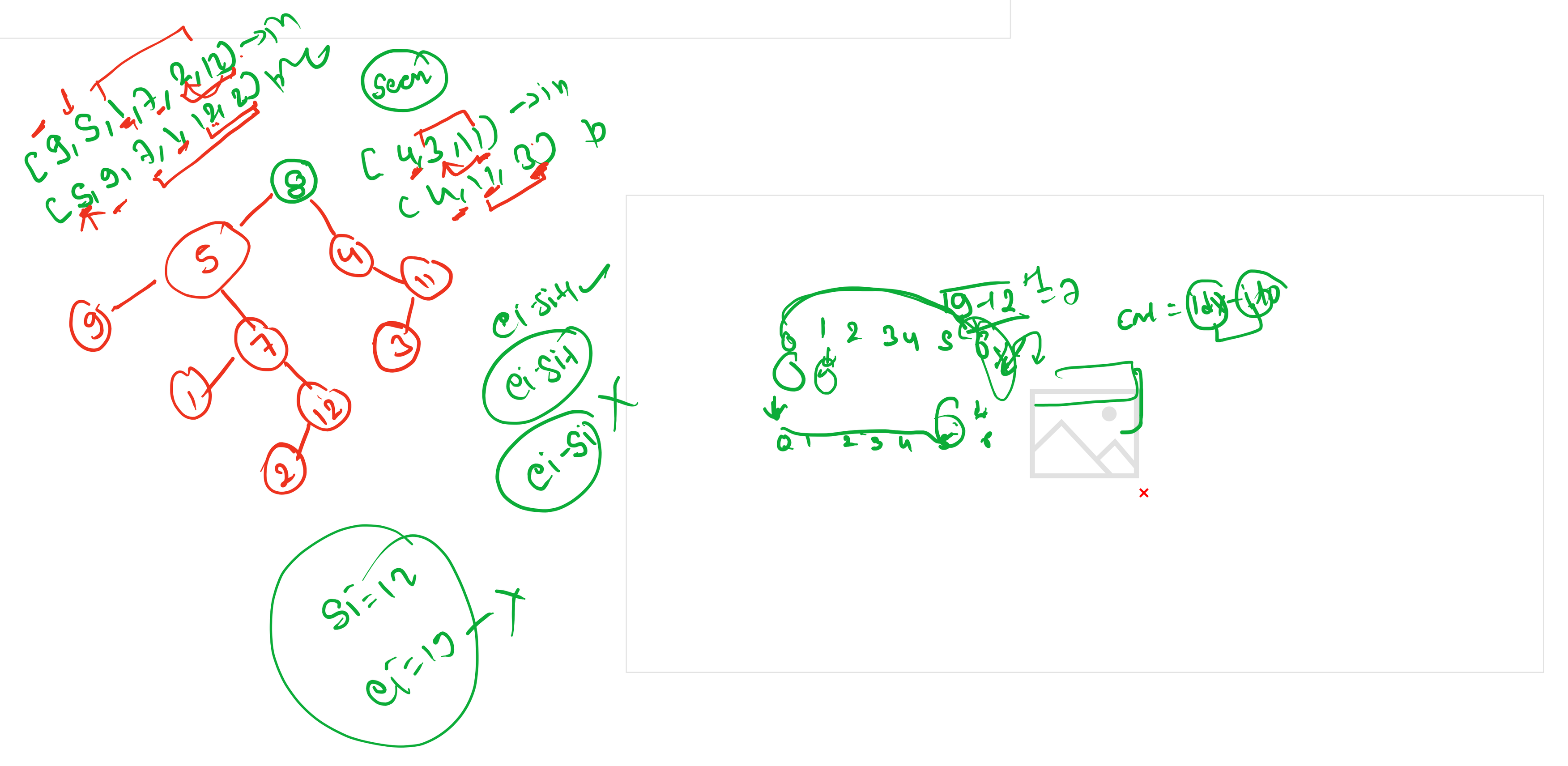
Input: preorder = [3,9,20,15,7], inorder = [9,3,15,20,7]

Output: [3,9,20,null,null,15,7]

Input: preorder = [3,9,20,15,7], inorder = [9,3,15,20,7]







```java
public TreeNode CreateTree(int[] pre, int[] in, int plo, int phi, int ilo, int ihi) {
        if (ilo > ihi || plo > phi) {
            return null;
        }
        TreeNode node = new TreeNode(pre[plo]);
        int idx = Serach(in, ilo, ihi, pre[plo]);
        int count = idx - ilo;
        node.left = CreateTree(pre, in, plo + 1, plo + count, ilo, idx);
        node.right = CreateTree(pre, in, plo + count + 1, phi, idx + 1, ihi);
        return node;
    }
```