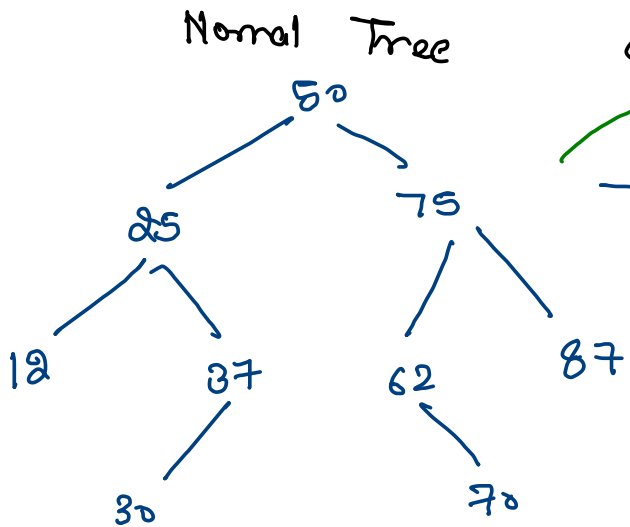
Transform to left cloned

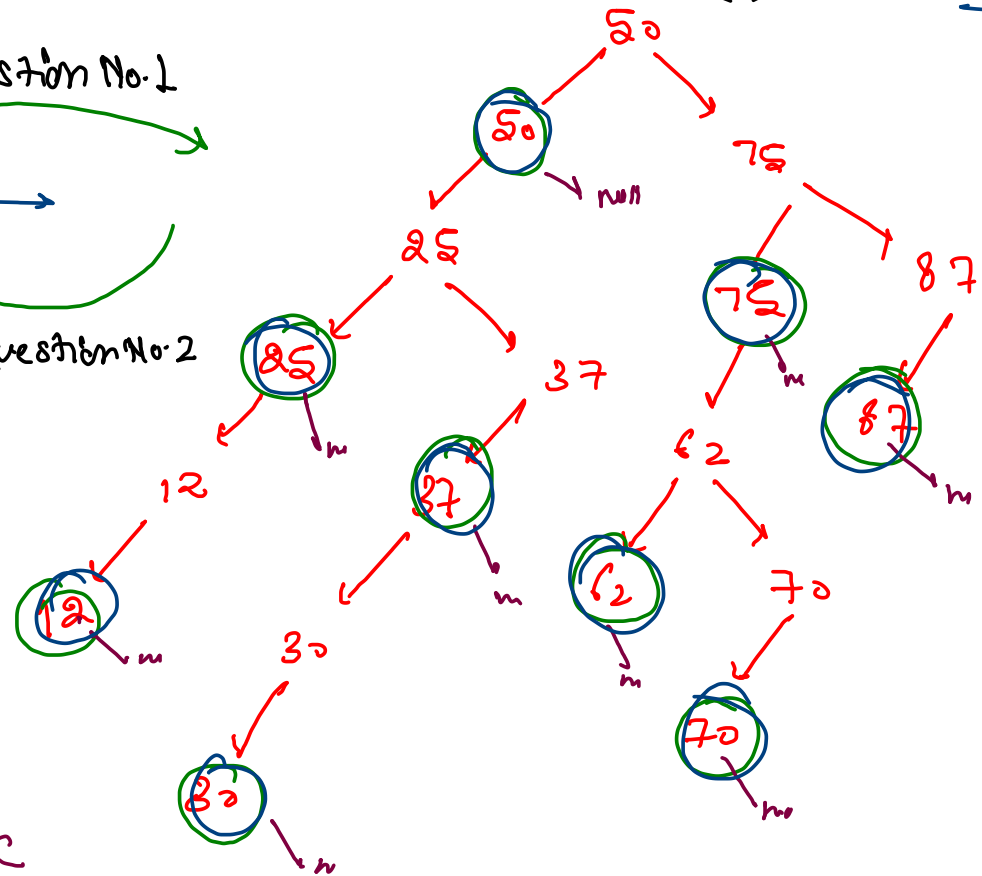


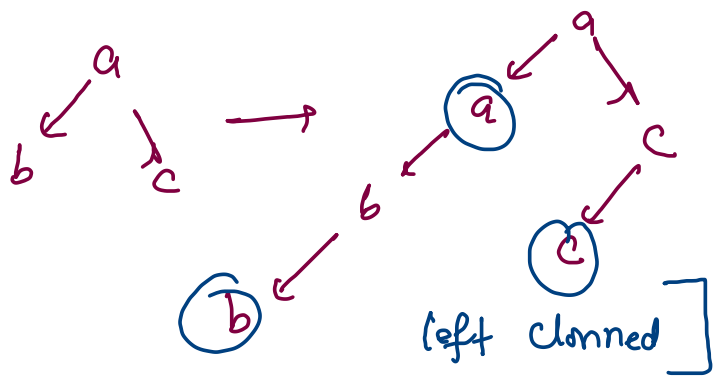
Question No. 1

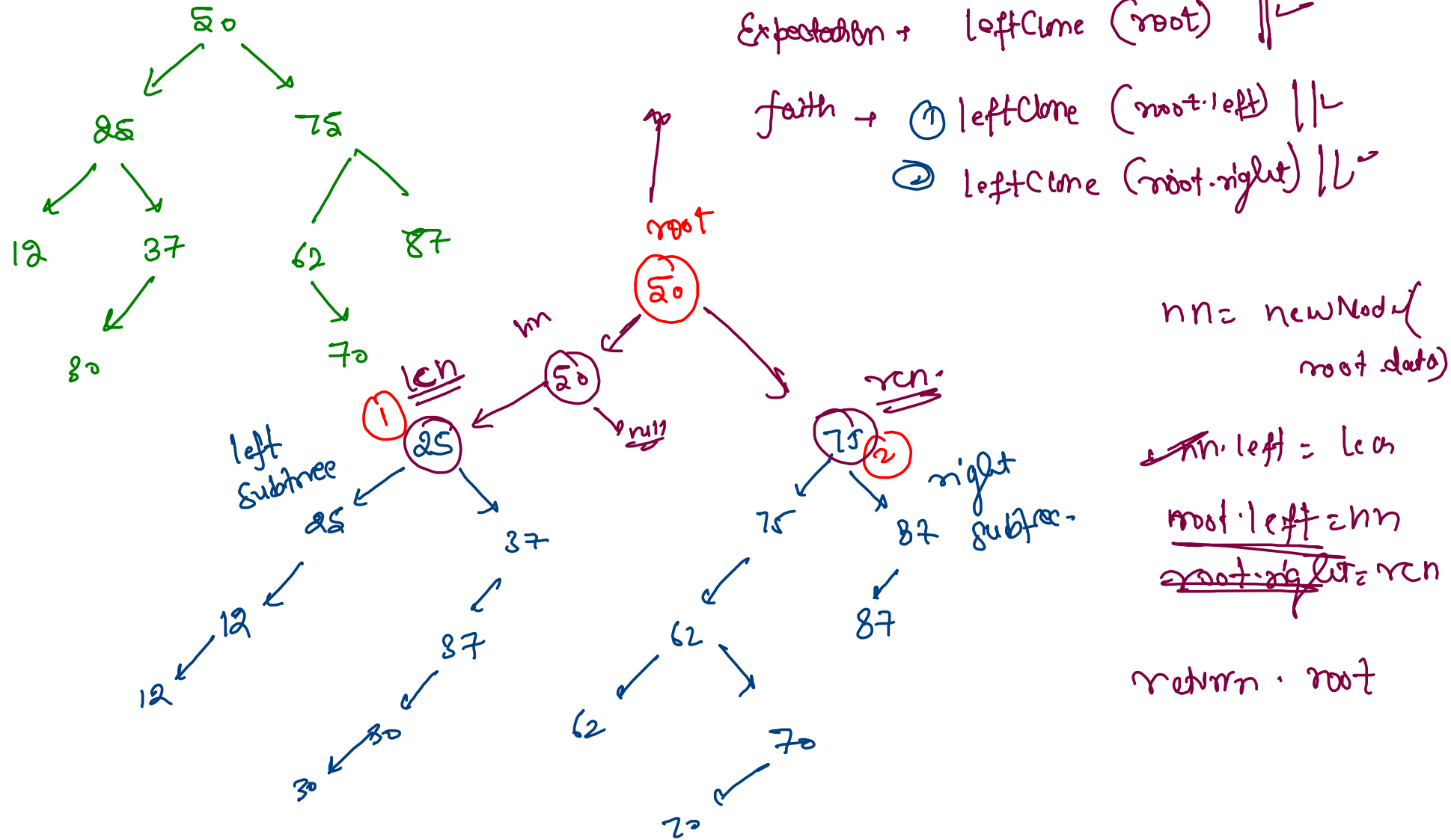
Question No. 2

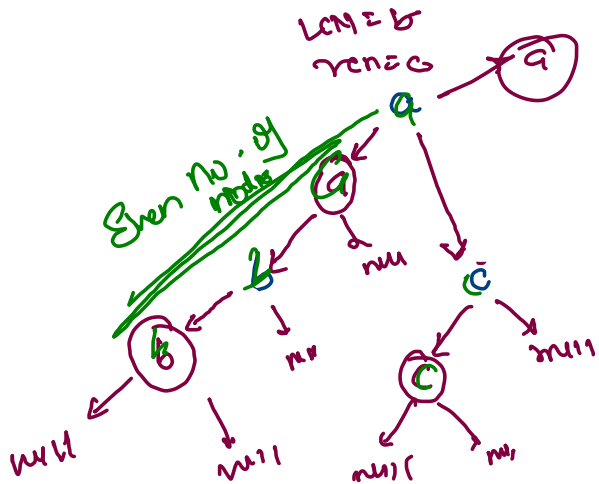
left Cloned Tree,

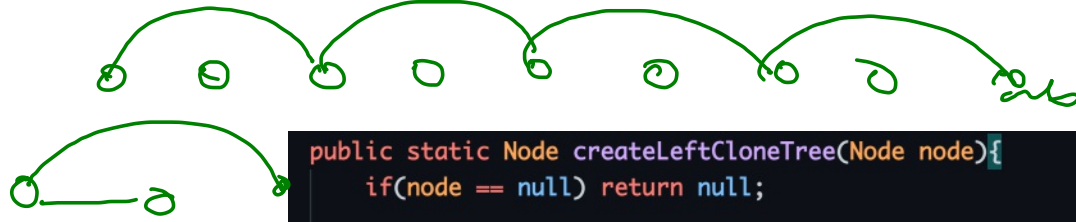
recursive space






50
25
75
12
37
62
87
30
70
root
lcn
rcn
nn
left subtree
right subtree
null
Expectation → leftClone (root) ||L
faith → ① leftClone (root.left) ||L
② leftClone (root.right) ||L
nn = newNode(root.data)
nn.left = lcn
root.left = nn
root.right = rcn
return root



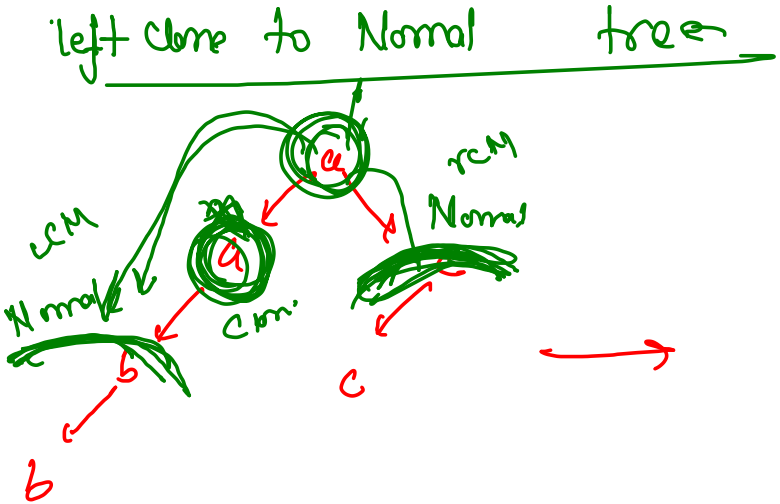


```java
public static Node createLeftCloneTree(Node node){
    if(node == null) return null;

    Node lcn = createLeftCloneTree(node.left); // left child node
    Node rcn = createLeftCloneTree(node.right);// right child node

    Node nn = new Node(node.data, lcn, null);
    node.left = nn;
    node.right = rcn;

    return node;
}
```



Expectation → normalTree(root)

faith LCN normalTree(root.left.left)

RCN normalTree(root.right)

root.left = LCN

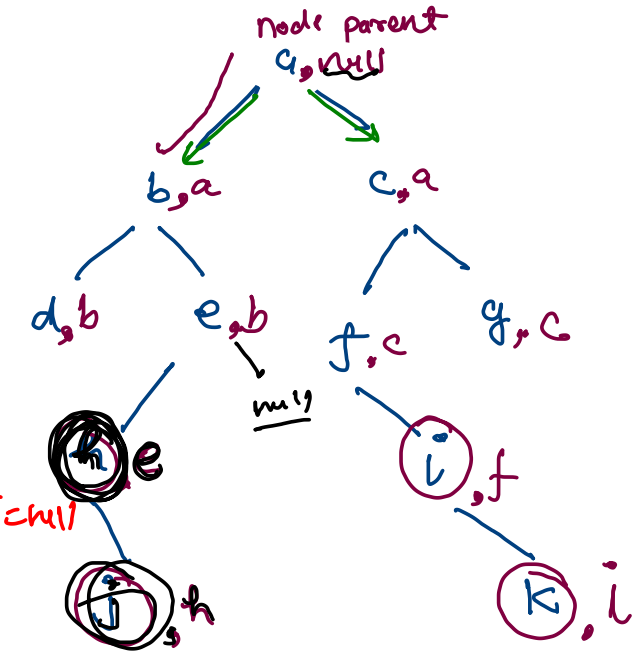
root.right = RCN

return root;

Print single child node :

Single child

h i i/c



Method :

left child ←
right child ×
print left child (single child)

right child →
left child ×
print right child (single child)

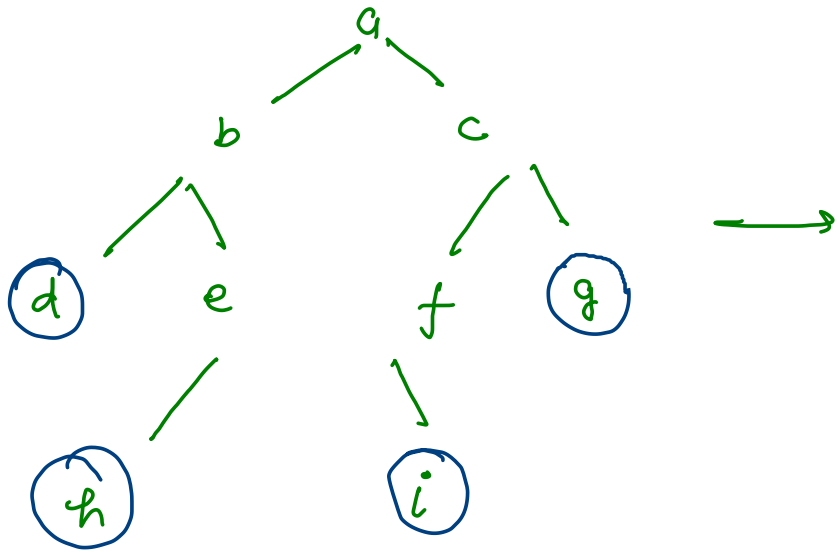
Single child :

Condition 1 → parent != null && } root
parent.left == node && parent.right == null

↳ I am single left child of my parent.

Condition 2 → parent != null } root
parent.right == node && parent.left == null

↳ I am single right child of my parent.


a
b
c
d
e
f
g
h
i