

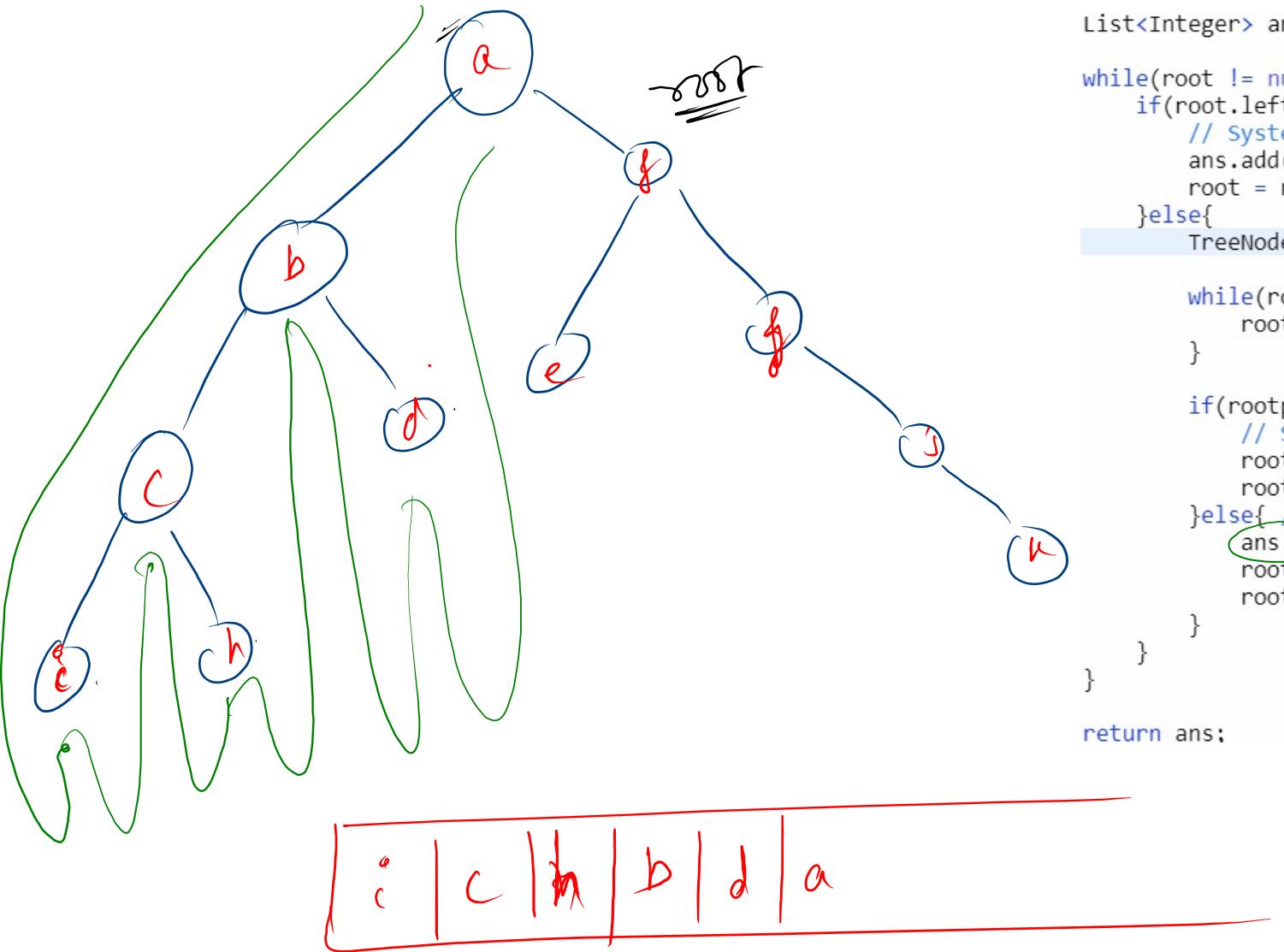
n nodes \rightarrow $(n-1)$ Edges

```

List<Integer> ans = new ArrayList<>();
while(root != null){
  if(root.left == null){
    // System.out.println(root.val);
    ans.add(root.val);
    root = root.right;
  }else{
    TreeNode rootp1 = root.left;

    while(rootp1.right != null && rootp1.right != root){
      rootp1 = rootp1.right;
    }

    if(rootp1.right == null){ // 1st visit
      // System.out.println(root.val);
      ans.add(root.val);
      rootp1.right = root;
      root = root.left;
    }else{ // 2nd visit
      rootp1.right = null;
      root = root.right;
    }
  }
}
return ans;
  
```



```

List<Integer> ans = new ArrayList<>();

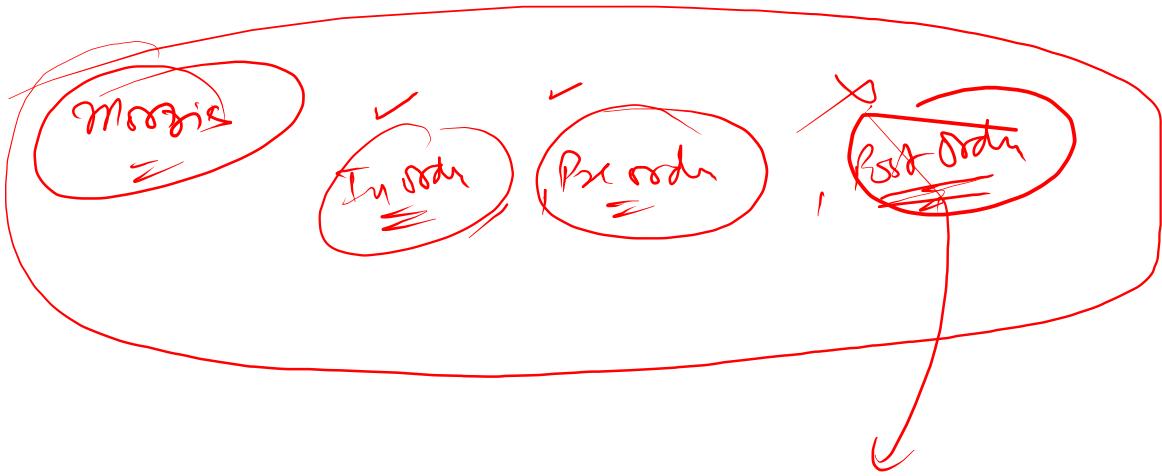
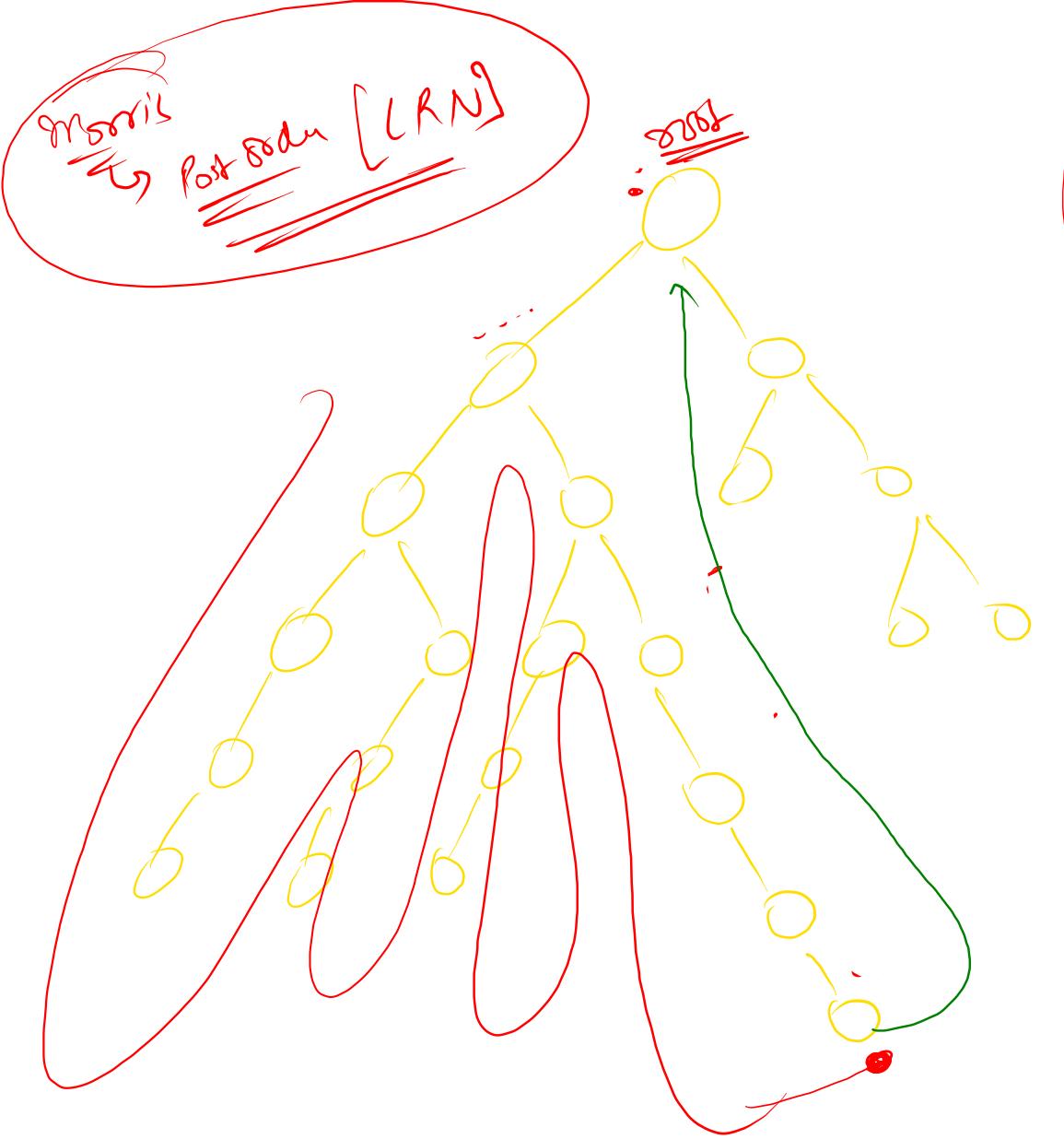
while(root != null){
    if(root.left == null){
        // System.out.println(root.val);
        ans.add(root.val);
        root = root.right;
    }else{
        TreeNode rootp1 = root.left;

        while(rootp1.right != null && rootp1.right != root){
            rootp1 = rootp1.right;
        }

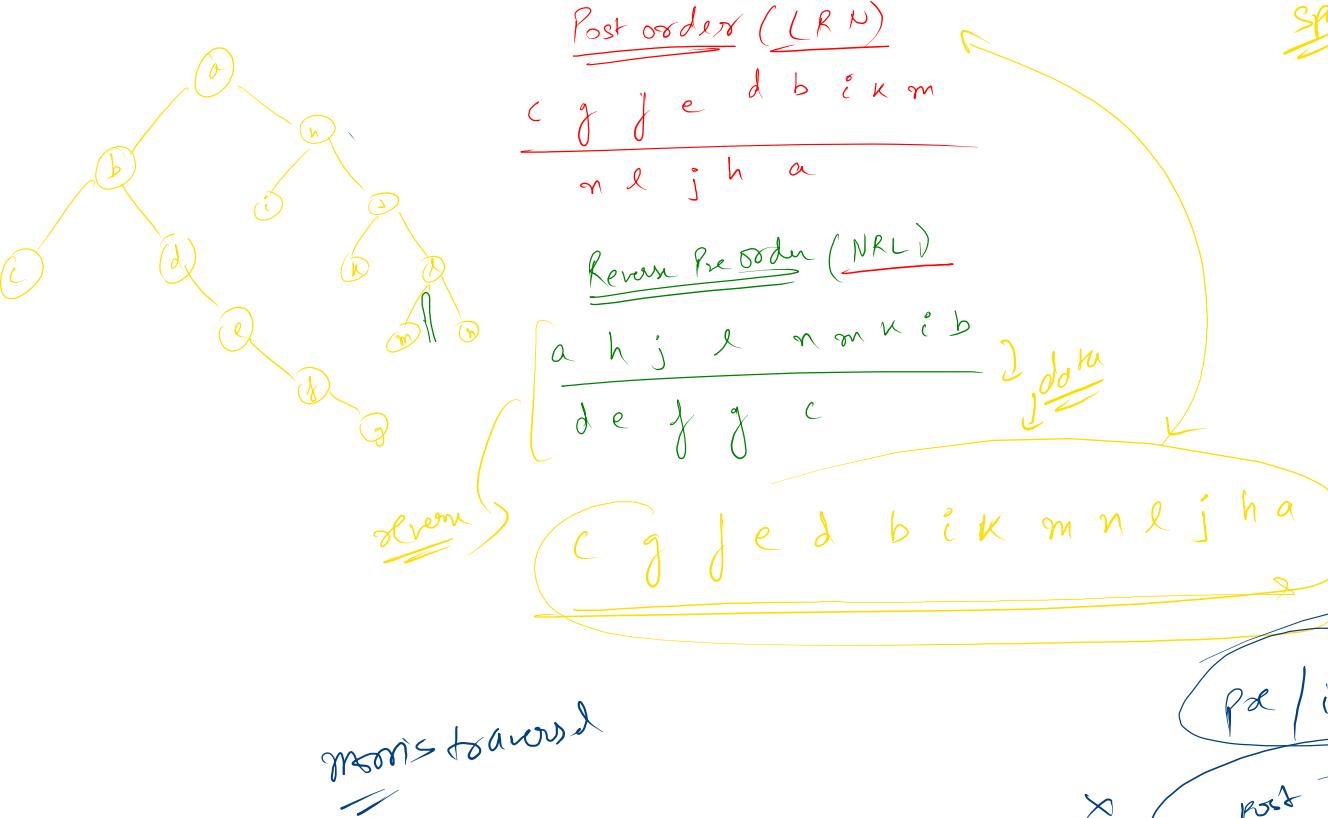
        if(rootp1.right == null){ // 1st visit
            // System.out.println(root.val);
            rootp1.right = root;
            root = root.left;
        }else{ // 2nd visit
            ans.add(root.val);
            rootp1.right = null;
            root = root.right;
        }
    }
}

return ans;

```



Collektions-Server (=)



spec

Inorder [return \downarrow Ary \rightarrow AL, process / point]
 $O(n)$ $O(1)$

+ Postorder

view & pointer

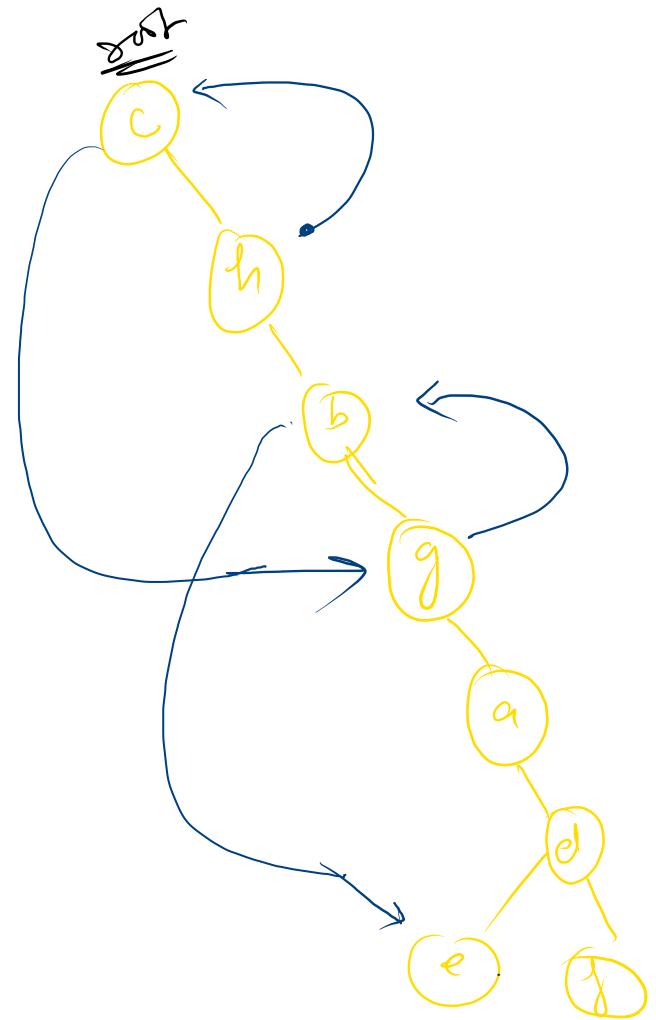
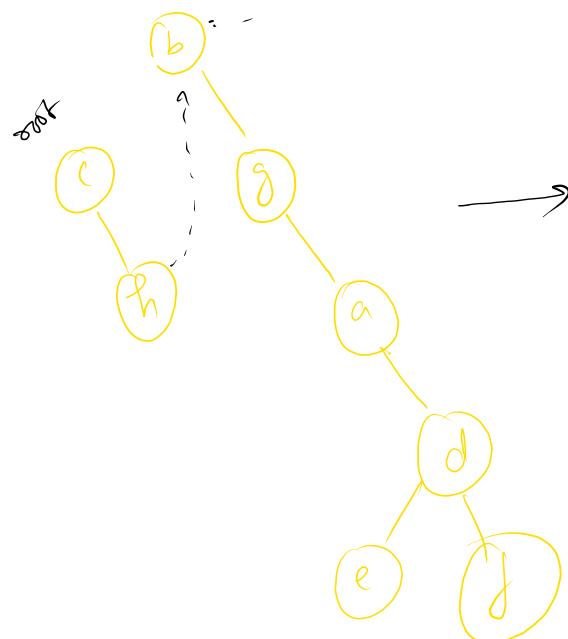
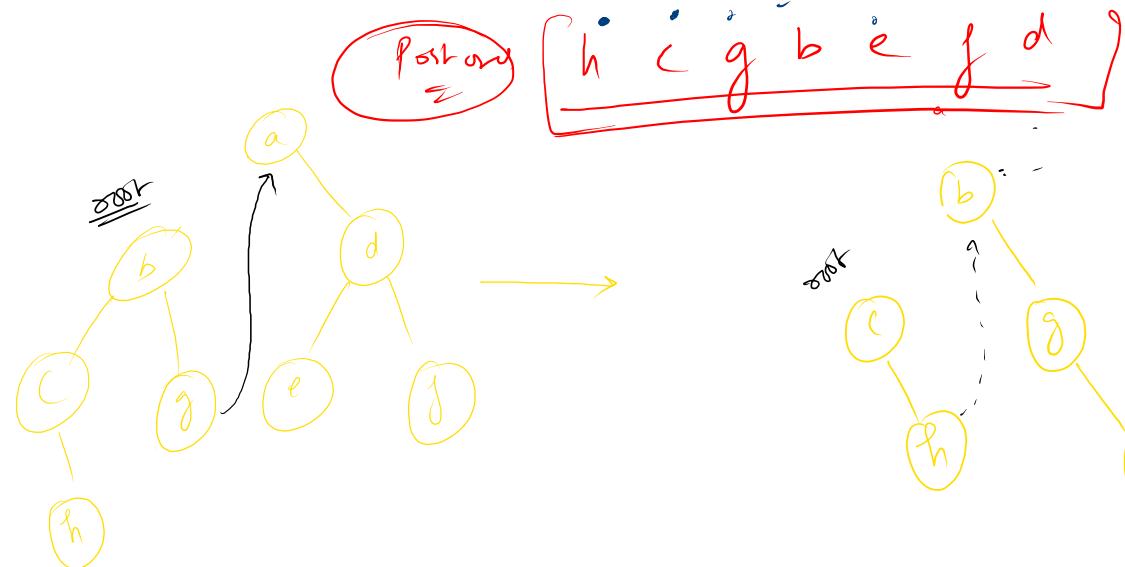
Postorder Pointing ≠ Postorder Traversed

view process

$O(n)$

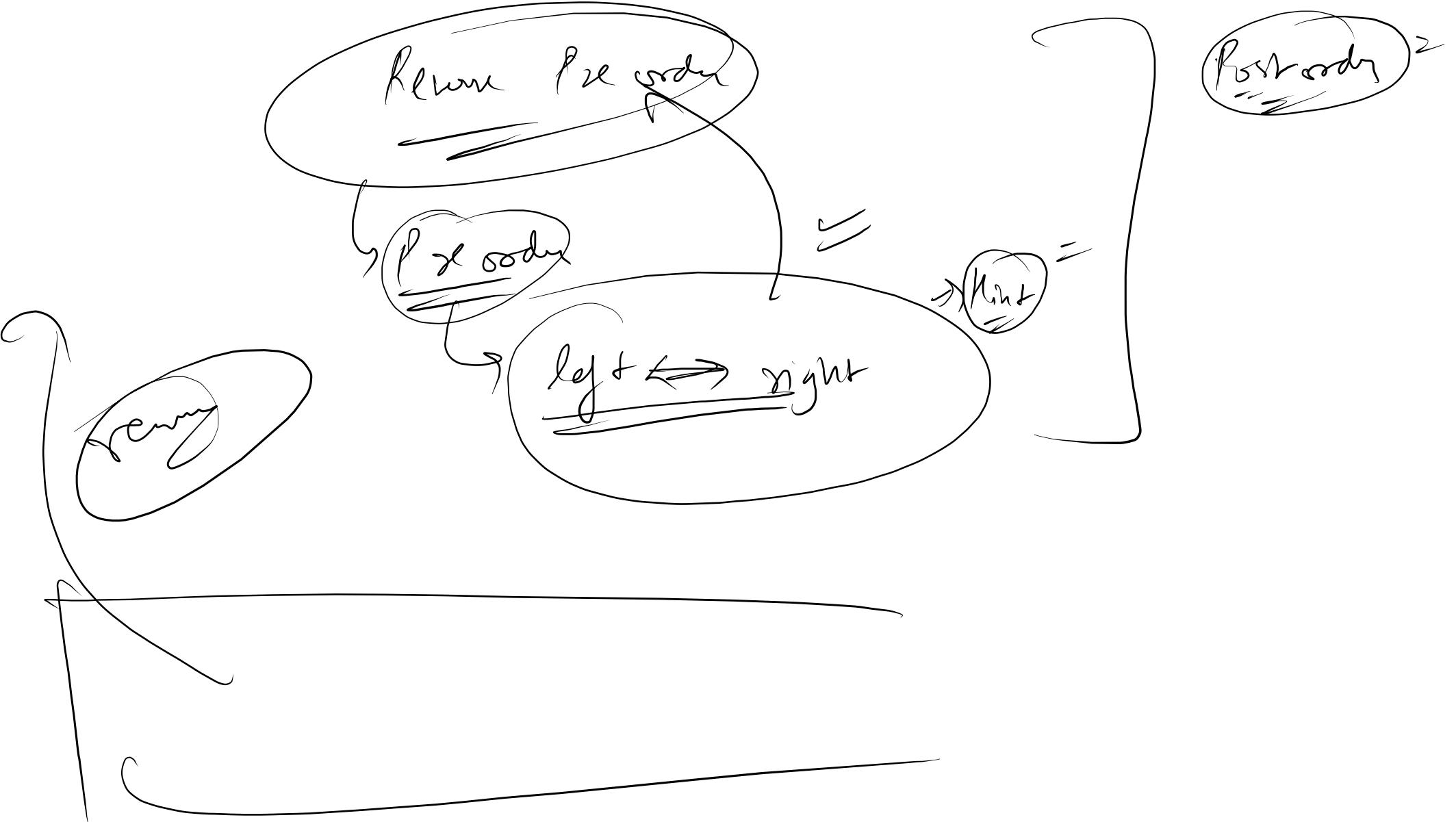
post in → optimised / process

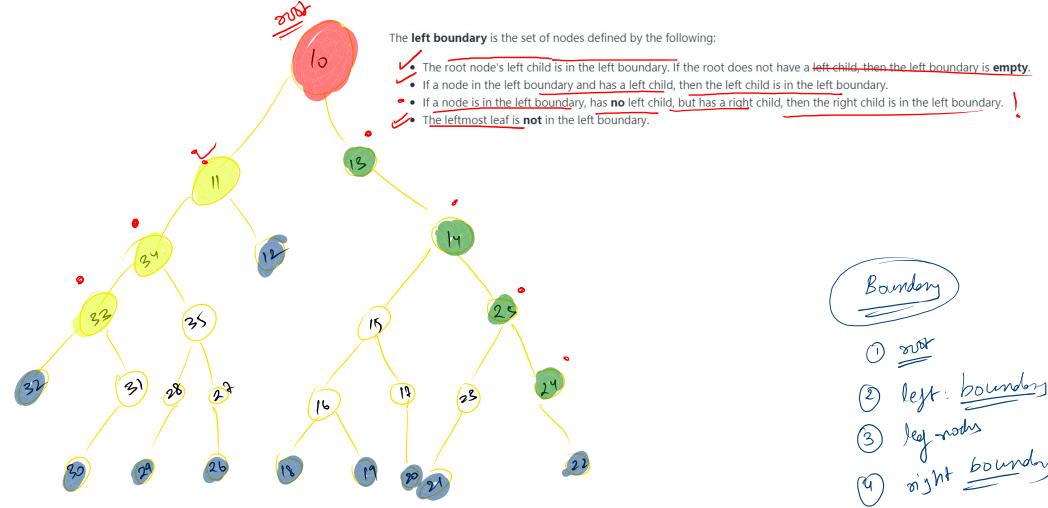
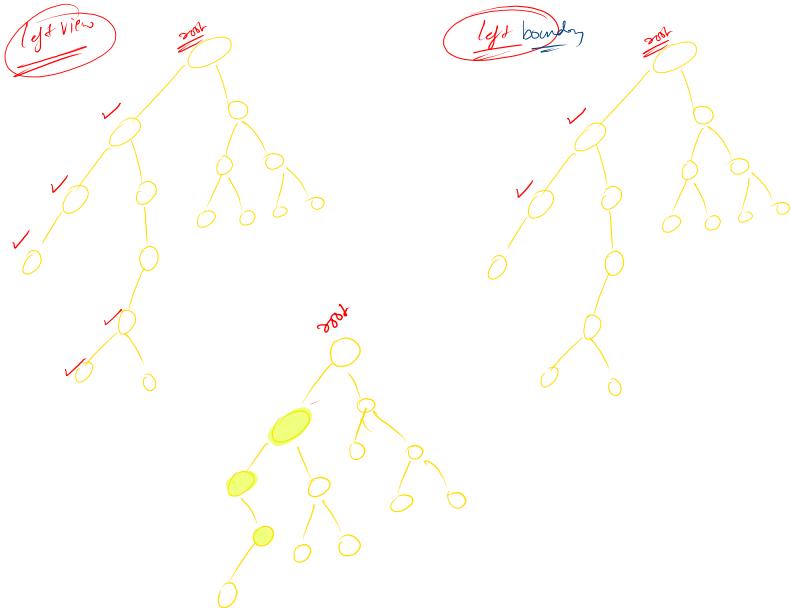
post → optimised / S.o.C.

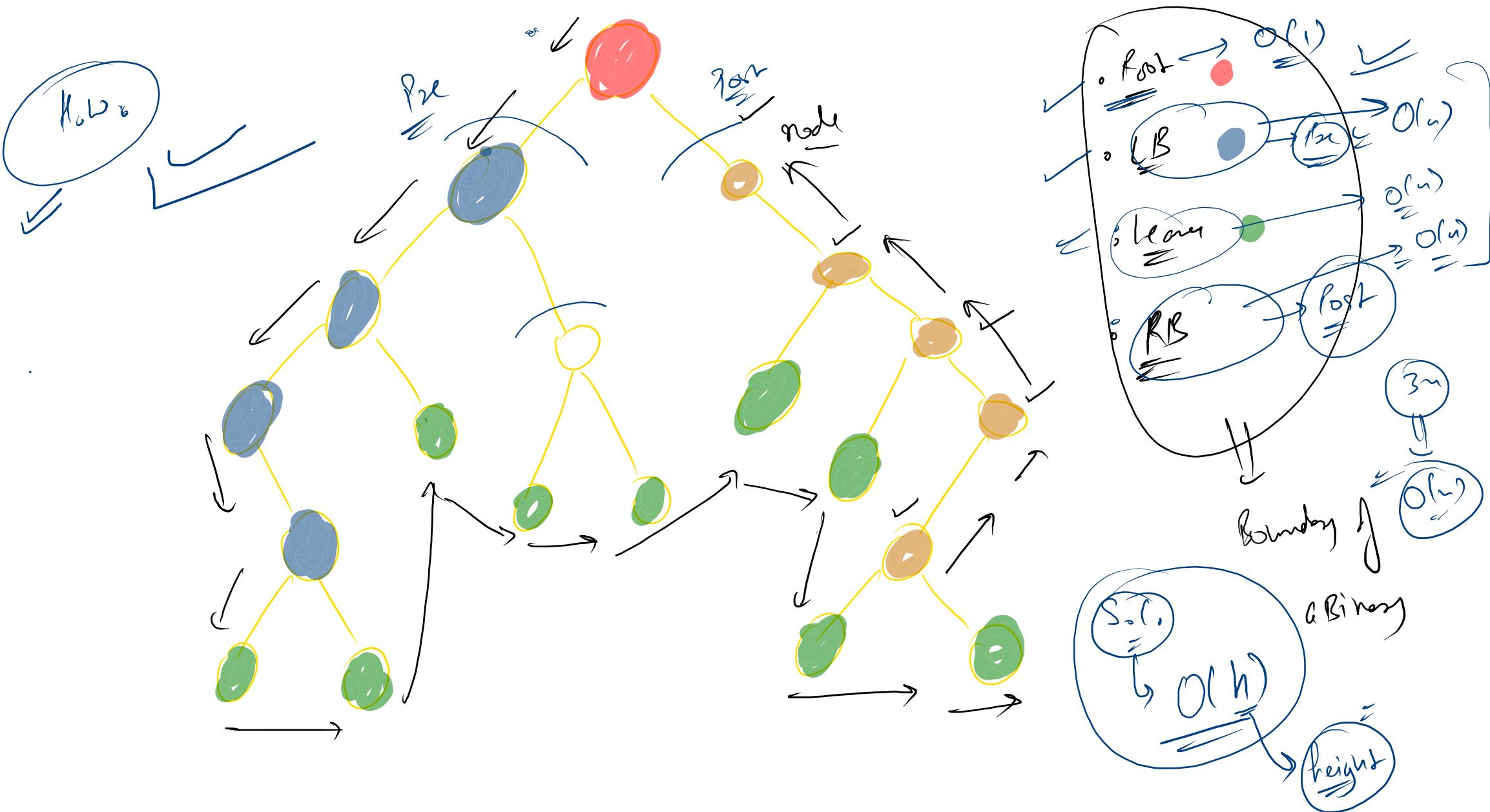




Pre	In	Post
$S.C. = O(h)$	$S.C. = O(n)$	$S.C. = O(n)$
$T.C. = O(n)$		
	$S.C.$	
	$T.C.$	
$S.C. \geq O(1)$	$S.C. \geq O(1)$	$S.C. \geq O(1)$
$T.C. \geq O(n)$	$T.C. \geq O(n)$	$T.C. \geq O(n)$
		No direct method

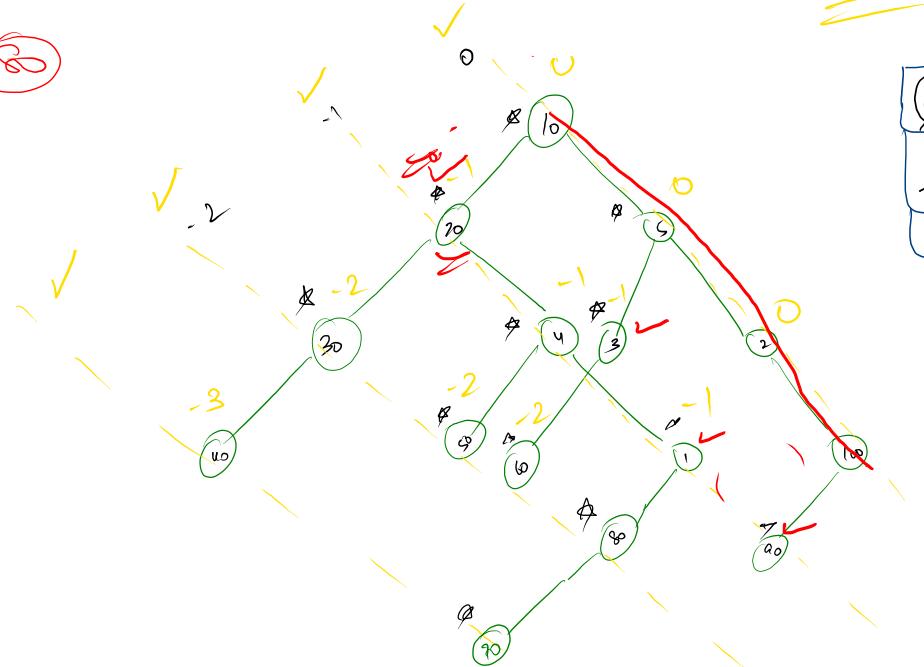






~~10 5 2 10~~

→ ~~20 4 1~~



Diagonal

(10, 0)	(20, -1)	(5, 0)	(30, -2)	(4, 1)
(3, -1)	(2, 0)	(4, -3)	(5, 2)	
(1, -1)	(6, -2)	(10, 0)	(8, -2)	(9, -1)

