

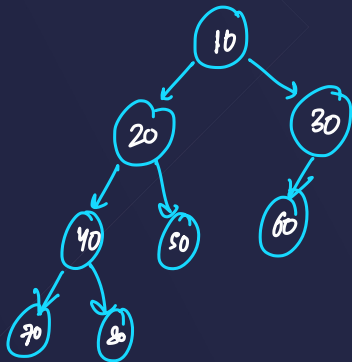


# Trees

1) Non-linear

2) Root, left, right

LST, RST



Binary tree  
is Recursion

3) Size, Sum, Max, Height/Level

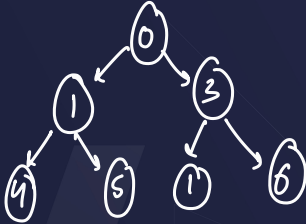
$$\text{size}(\text{tree}) = 1 + \text{size}(\text{LST}) + \text{size}(\text{RST})$$

```
public static int productExceptZero(Node root){
    if(root==null || root.val==0) return 1;
    int x = root.val;
    if(x==0) x = 1;
    return x * productExceptZero(root.left) * productExceptZero(root.right);
}
```

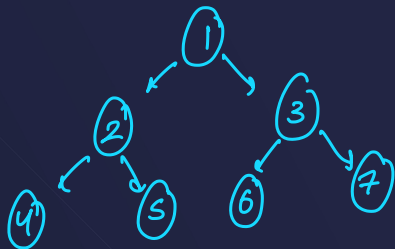
Genuine  
Doubt

$$pro(1) = 1 * pro(0) * pro(3)$$

$$pro(0) = 1 * pro(4) * pro(5)$$



0, 1, 3, 4, 5, 1, 6, 3

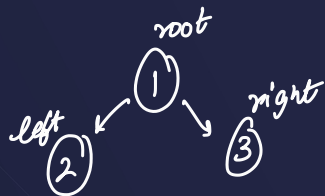


```

print (root){
    if (root == null) return;
    sout (root.val);
    print (root.left);
    print (root.right);
}
  
```

1 2 4 5 3 6 7

COLLEGE  
WALLAH



preorder  $\rightarrow$  Root Left Right

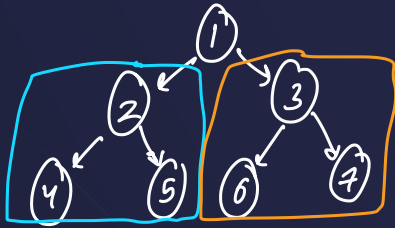
1 2 3

Inorder  $\rightarrow$  Left Root Right

2 1 3

PostOrder  $\rightarrow$  Left Right Root

2 3 1

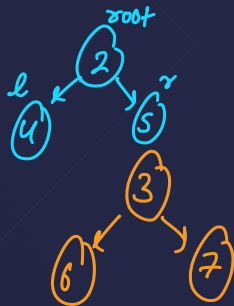


Preorder : Root Left Right

1 (245) (367)

1 2 4 5 3 6 7

Inorder: Left Root Right



4 2 5

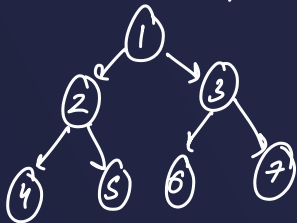
6 3 7

(2 4 5) 1 (3 6 7)

4 2 5 1 6 3 7

COLLEGE  
WALLAH

PostOrder : Left Right Root



(2 4 5) (3 6 7) 1

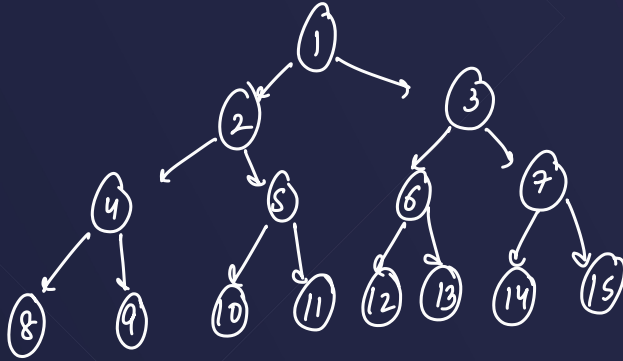
4 5 2 6 7 3 1

4 5 2 6 7 3 1



Homework :

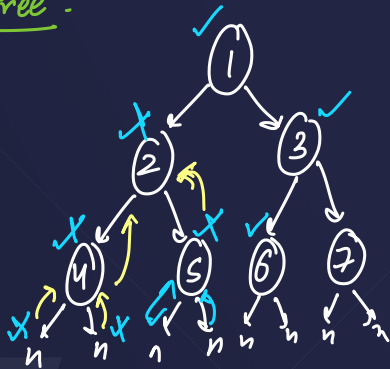
Pre, In, Post



8 4 9 2 10 5 11 1 12 6 13 3 14 7 15

Euler's tree :

✓✗

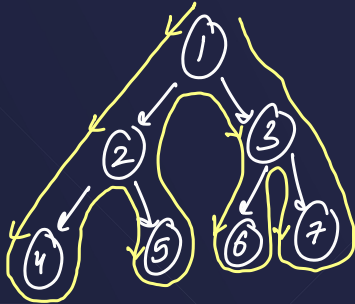


```
preorder(root)
    if (root == null) re
    sent (root.val)
    preorder(left)
    preorder(right)
```

3

1 2 4 5 3 6 7

# Euler's Tree :

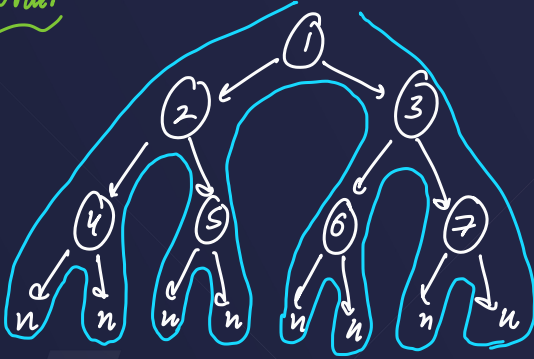


Pre = 1 2 4 5 3 6 7

in: 4 2 5 1 6 3 7

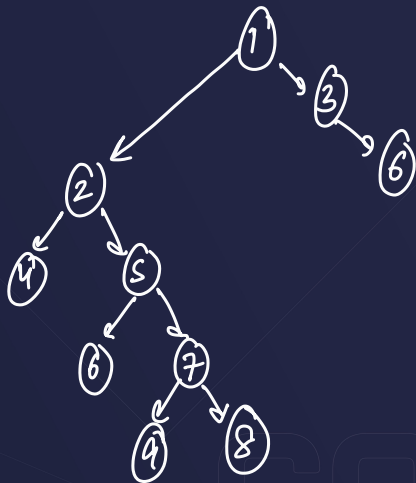
COLLEGE  
WALLAH

Post Order



4 5 2 6 7 3 1

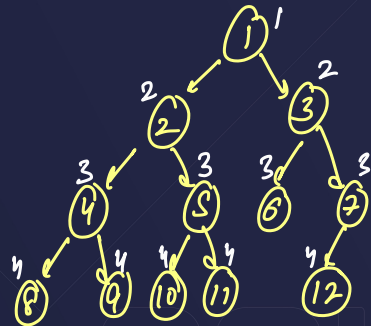
COLLEGE  
WALLAH



# Breadth First Search (BFS)

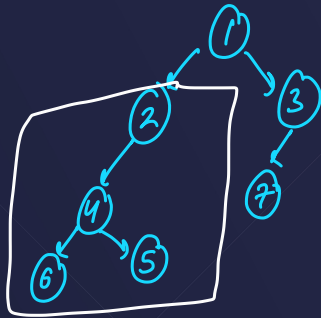
## Level Order Traversal

1  
2 3  
4 5 6 7  
8 9 10 11 12



- 1) It can be done via recursion → by printing each level separately
- 2) It can be done using a queue

# Balanced Binary Tree (LeetCode 110)



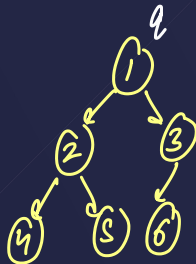
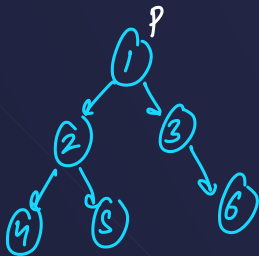
NO



COLLEGE  
WALLAH

Same Tree

Node p, Node q



pre : 1 2 4 5 3 6

pre: 1 2 4 5 3 6

Goodnight ♡