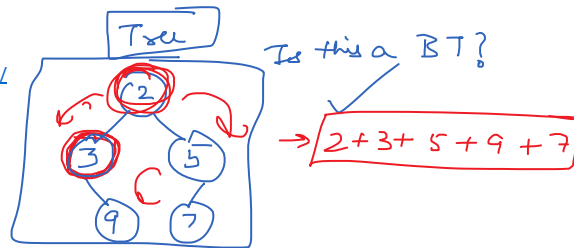


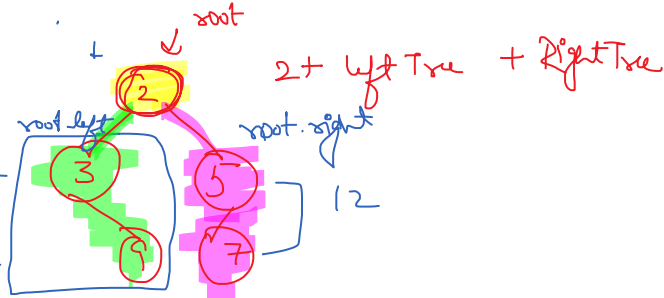
Link for reference: <https://thecodingsimplified.com/binary-tree/>

1. Create Binary Tree // Done
2. Sum of all nodes // Done
3. Get the total number of nodes // Done
4. Total number of Leaf nodes // Done
5. Height of a binary tree // Done
6. Print the elements at any particular level // Done



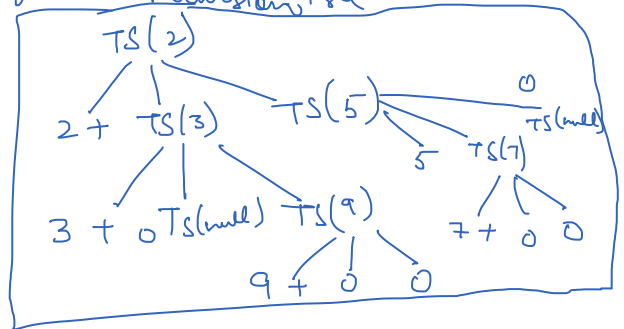
Sum of all Nodes

TreeSum(root)

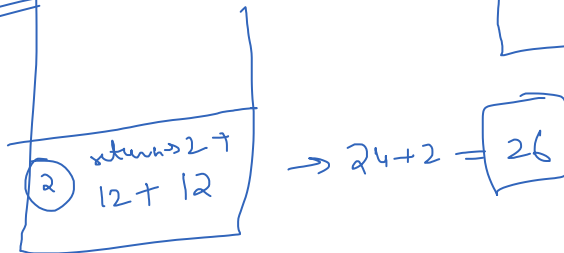


Node.data + LeftSubTreeSum + RightSubTree  
Khaali Wala Case

```
int TreeSum(Node root)
{
    if (root == null) return 0;
    return (root.data + TreeSum(root.left) + TreeSum(root.right));
}
```

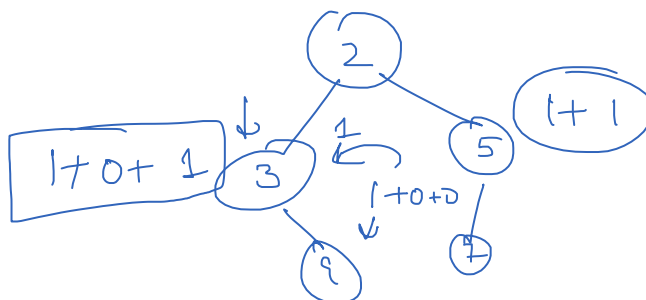


Stack



Get total No. of Nodes

$1 + 2 + 2 = 5$



```
int countNodes(Node root)
{
    if (root == null) return 0;
}
```

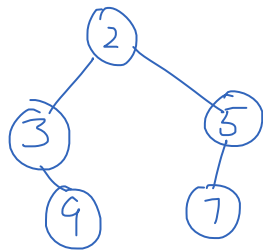
int countNodes -

```

{ if (root == null) return 0;
  return 1 + countNodes(root.left) + countNodes(root.right);
}

```

Total No of leaf Nodes

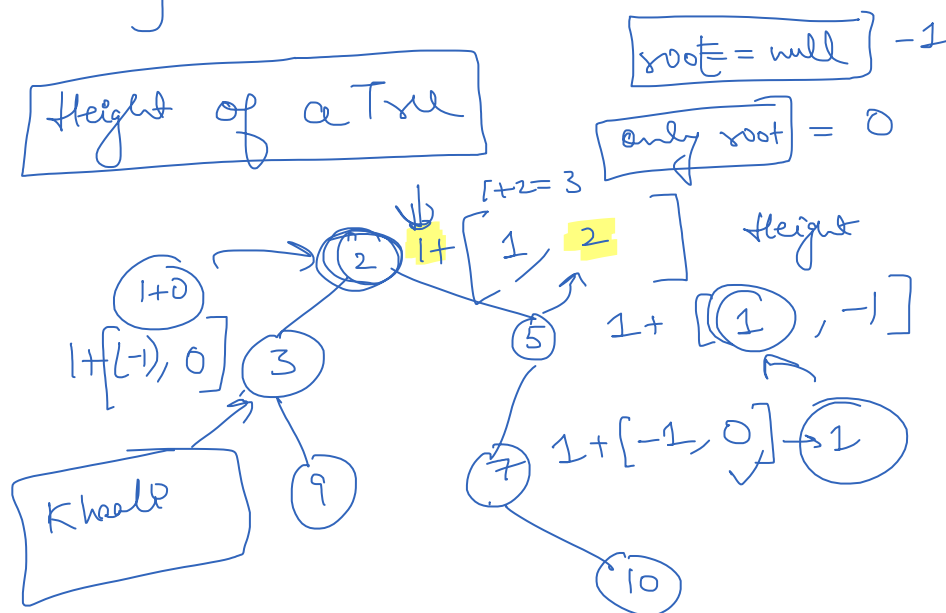


```

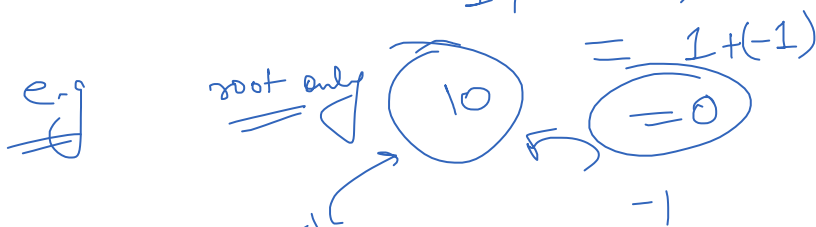
{ if (root == null) return 0;
  if (root.left == null && root.right == null) return 1;
  return L.S.T + R.S.T;
}

```

Height of a Tree



$1 + \max(L.S.T, R.S.T)$   $1 + \max(-1, -1)$

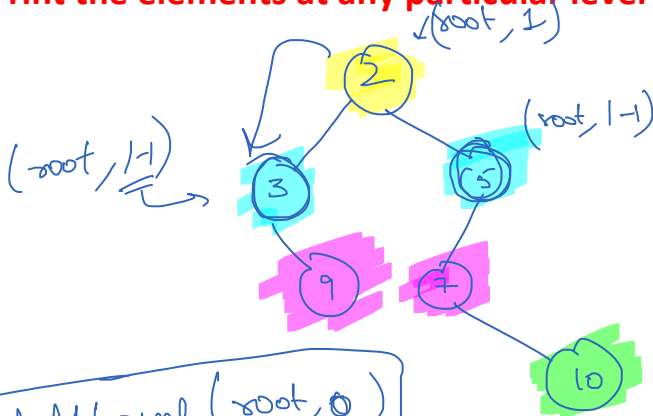


≡



## Print the elements at any particular level

[We are printing when level is 0]



0

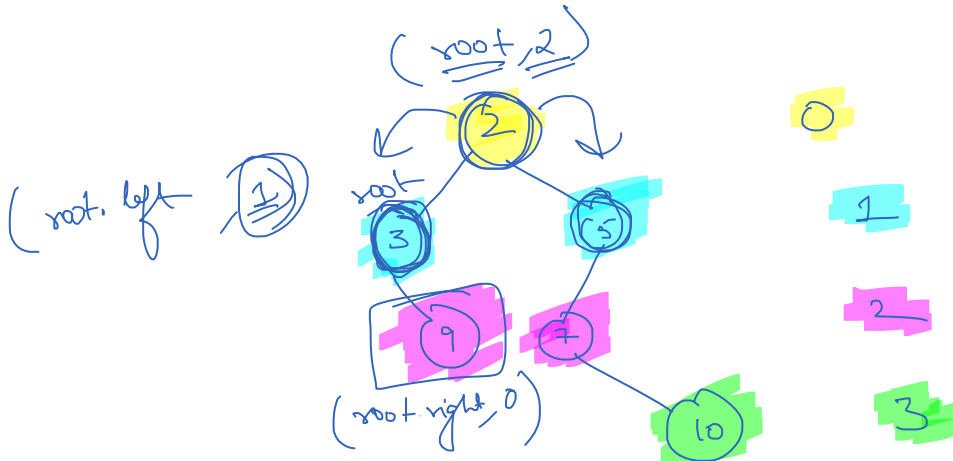
1

2

3

```
PrintAtLevel (root, 0)
{
    if (level == 0)
    {
        S.O.P (root.data + " ");
    }
}
```

PrintAtLevel (root, 1)



0

1

2

3