

<https://course.acciojob.com/idle?question=57d8deff-acbe-407a-b504-6fe20f94770e>

● EASY

● Max Score: 30 Points

●

Sum of all Nodes

Give an algorithm for finding the sum of all elements in a binary tree.

Input Format

Given the root of the tree.

Output Format

Return the sum of all nodes.

EXAMPLE 1

Input:

```
10 4 6 N N N N
```

Output:

```
20
```

EXPLANATION

```
10
```

```
/  \
```

```
4    6
```

The Sum of nodes is 20

EXAMPLE 2

Input

100 N N

Output:

100

EXPLANATION

Sum of nodes is 100.

CONSTRAINTS

The number of nodes in the tree is in the range [0, 5000].

$1 \leq \text{Node.val} \leq 10000$

Topic Tags

- Trees

My code

```
// in java
import java.util.LinkedList;
```

```
import java.util.Queue;
import java.io.*;
import java.util.*;
```

```
class Node{
    int data;
    Node left;
    Node right;
    Node(int data){
        this.data = data;
        left=null;
        right=null;
    }
}
```

```
class Main {
    static Node buildTree(String str){
        if(str.length()==0 || str.charAt(0)=='N'){
            return null;
        }
        String ip[] = str.split(" ");
        Node root = new Node(Integer.parseInt(ip[0]));
        Queue<Node> queue = new LinkedList<>();
        queue.add(root);
        int i = 1;
        while(queue.size()>0 && i < ip.length) {
            Node currNode = queue.peek();
            queue.remove();
            String currVal = ip[i];
            if(!currVal.equals("N")) {
                currNode.left = new Node(Integer.parseInt(currVal));
```

```

        queue.add(currNode.left);
    }
    i++;
    if(i >= ip.length)
        break;
    currVal = ip[i];
    if(!currVal.equals("N")) {
        currNode.right = new Node(Integer.parseInt(currVal));
        queue.add(currNode.right);
    }
    i++;
}
return root;
}

```

```

public static void main (String[] args) throws IOException{
    Scanner sc = new Scanner(System.in);
    String s = sc.nextLine();
    Node root = buildTree(s);
    Solution tree = new Solution();
    int result=tree.addBT(root);
    System.out.print(result);
}
}

```

```

class Solution{
    public int addBT(Node root)
    {
        //Write code here
    }
}

```

```
        if(root==null)
            return 0;
        return root.data+addBT(root.left)+addBT(root.right);
    }
}
```