https://course.acciojob.com/idle?question=7e89e1c0-0a8f-4e81-af44-b386b3293bce

- EASY
- Max Score: 30 Points

Child Sum Tree

Given the root node of a tree, check whether it holds the child sum property.

The child sum property means that each root node is equal to the sum of its child nodes.

Input Format:

You are given the pointer to the root of the binary tree.

Output Format:

For each test case print "YES" if the tree follows the child sum property, else print "NO"

Example 1:

Input:

```
a=[3,2,1]
```

Output:

```
YES
```

Example 2:

Input:

```
a = [1 2 3]
```

Output:

```
NO
```

Constraints:

1 <= n <= 10^5

Value of any node is less than 2^32

Topic Tags

- **Trees**

My code

```
// in java
import java.util.LinkedList;
import java.util.Queue;
import java.io.*;
import java.util.*;



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 {
        BufferedReader br =new BufferedReader(new
InputStreamReader(System.in));
        String s1 = br.readLine();
        Node root1 = buildTree(s1);
        Solution g = new Solution();
        g.childSumProperty(root1);
    }
```

```
}


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

class Solution {
        static int f=1;
        static int fun(Node n)
        {
                int t=0,p=0;
                if(n.left==null  && n.right==null)
                        return n.data;
                if(n.left!=null  )
                         t= fun(n.left);
                if(n.right!=null  )
                         p= fun(n.right);
                if((p+t)!=n.data)
                        f=0;
                return p+t;
        }
    public static void childSumProperty(Node root) {
                    fun(root);
              if(f==1)
```

```
                System.out.print("YES");
            else  System.out.print("NO");
    }
}
```