Children Sum Parent

Link:- <https://practice.geeksforgeeks.org/problems/children-sum-parent/1?utm_source=gfg&utm_medium=article&utm_campaign=bottom_sticky_on_article>

Given a Binary Tree. Check whether all of its nodes have the value equal to the sum of their child nodes.

**Example 1:**

**Input:**

10

/

10

**Output:** 1

**Explanation:** Here, every node is sum of

its left and right child.

**Example 2:**

**Input:**

1

/ \

  4 3

  / \

  5 N

**Output:** 0

**Explanation:** Here, 1 is the root node

and 4, 3 are its child nodes. 4 + 3 =

7 which is not equal to the value of

root node. Hence, this tree does not

satisfy the given conditions.

Non-Recursive Approach:-

class Node{

int data;

Node left,right;

Node(int key)

{

data = key;

left = right = null;

}

class Tree

{

//Function to check whether all nodes of a tree have the value

//equal to the sum of their child nodes.

public static int isSumProperty(Node root)

{

// add your code here

ArrayList<Node> ar=new ArrayList<>();

ar.add(root);

while(ar.size()!=0){

Node temp=ar.remove(0);

if(!(temp.left==null && temp.right==null)){

int sum=0;

Node l=null;

Node r=null;

if(temp.left!=null){

sum=sum+temp.left.data;

l=temp.left;

}

if(temp.right!=null){

sum=sum+temp.right.data;

r=temp.right;

ar.add(0,r);

}

if(sum!=temp.data)

return 0;

if(l!=null)

ar.add(0,l);

}

}

return 1;

}

}

Recursive Approach:-

//User function Template for Java

/\*Complete the function below

Node is as follows:

class Node{

int data;

Node left,right;

Node(int key)

{

data = key;

left = right = null;

}

}

\*/

class Tree

{

//Function to check whether all nodes of a tree have the value

//equal to the sum of their child nodes.

//static int res=1;

public static int isSumProperty(Node node)

{

// add your code here

int sum=0;

if(node==null)

return 1;

if(node.left==null & node.right==null)

return 1;

else{

int l=0,r=0;

if(node.left!=null){

sum=sum+node.left.data;

// return isSumProperty(node.left);

}

if(node.right!=null){

sum=sum+node.right.data;

// return isSumProperty(node.right);

}

if(sum==node.data)

return (isSumProperty(node.left)&isSumProperty(node.right));

}

return 0;

}

}