**Sum Tree**

Given a Binary Tree. Return **true** if, for every node **X** in the tree other than the leaves, its value is equal to the sum of its left subtree's value and its right subtree's value. Else return **false**.

An empty tree is also a Sum Tree as the sum of an empty tree can be considered to be 0. A leaf node is also considered a Sum Tree.

**Example 1:**

**Input:**

3

/ \

1 2

**Output:** 1

**Explanation:**

The sum of left subtree and right subtree is

1 + 2 = 3, which is the value of the root node.

Therefore,the given binary tree is a **sum tree**.

**Example 2:**

**Input:**

10

/ \

20 30

/ \

10 10

**Output:** 0

**Explanation:**

The given tree is not a sum tree.

For the root node, sum of elements

in left subtree is 40 and sum of elements

in right subtree is 30. Root element = 10

which is not equal to 30+40.

**Your Task:**  
You don't need to read input or print anything. Complete the function **isSumTree()**which takes **root**node as input parameter and returns true if the tree is a SumTree else it returns false.

**Expected Time Complexity:**O(N)  
**Expected Auxiliary Space:** O(Height of the Tree)

**Constraints:**  
1 ≤ number of nodes ≤ 104