Practice(https://practice.geeksforgeeks.org/home/) 172 (/leaderboard)

Problem Editorial Submissions Doubt Support C++ (g++ 5.4) Test against custom input

```
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 ≤ 10^4

View Bookmarked Problems (https://practice.geeksforgeeks.org/explore/?problemType=bookmark)

Company Tags

Topic Tags

Related Courses

We are replacing the old Disqus forum with the new Discussions section given below. Click here (https://practice.geeksforgeeks.org/comments/sum-tree/1/?rel=https://practice.geeksforgeeks.org/problems/sum-tree/1) to view old Disqus comments.

```
{
    public:
    int solve(Node* root, bool* ans){
        if(root == NULL)
            return 0;
        if(root->left == NULL and root->right == NULL)
            return root->data;

        int l = solve(root->left, ans);
        int r = solve(root->right, ans);
        if(root->data != (l + r))
            *ans = *ans and false;
        else *ans = *ans and true;

        return (root->data + l + r);
    }
    bool isSumTree(Node* root)
    {
         // Your code here
         int sum = 0;
         bool ans = true;
         int height = solve(root, &ans);
         return ans;

```

Discussions (521 Threads)

+1

aryanmaurya314 5 hours ago

```
class Solution
{
 private:
  pair<bool, int> isSumTreeFast(Node* root){
   // base case
   if(root == NULL){
     pair<bool, int> p = make_pair(true, 0);
     return p;
   }
   // manage leaf nodes
   if(root->left == NULL && root->right == NULL)
     pair<bool, int> p = make_pair(true, root->dat
     return p;
   }

   pair<bool, int> left = isSumTreeFast(root->left
   pair<bool, int> right = isSumTreeFast(root->rig

   bool condition = root->data == left.second + rig

   bool leftAns = left.first;
   bool rightAns = right.first;

   pair<bool, int> ans;

   if(leftAns && rightAns && condition){
     ans.first = true;
     ans.second = 2*root->data;
   }
   else{
     ans.first = false;
   }
   return ans;
  }
 public:
 bool isSumTree(Node* root)
 {
    return isSumTreeFast(root).first;
 }
};
```

Reply Open Externally

Output Window

Problem Solved Successfully

You get marks only for the first correct submission if you solve the problem without viewing the full solution.

Test Cases Passed: **223** / 223

Total Time Taken: **0.03**/1.04

Correct Submission Count: **3**

Attempts No.: **3**

Compile & Run Submit