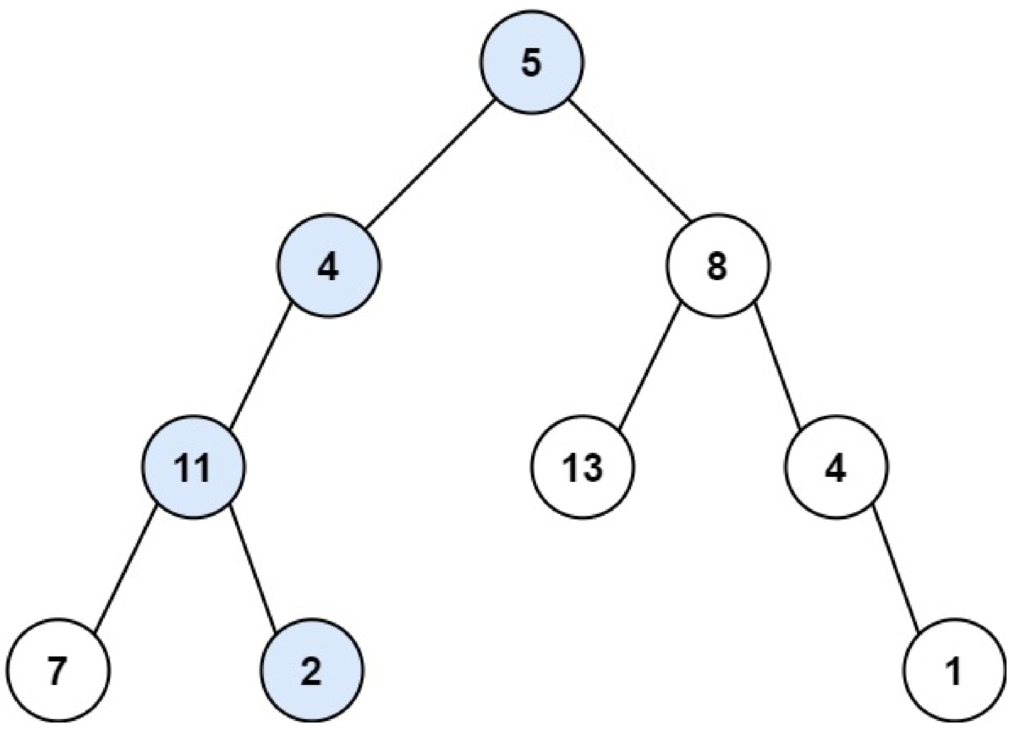
112. Path Sum

Easy 5992 809 Add to List Share

Given the `root` of a binary tree and an integer `targetSum`, return `true` if the tree has a **root-to-leaf** path such that adding up all the values along the path equals `targetSum`.

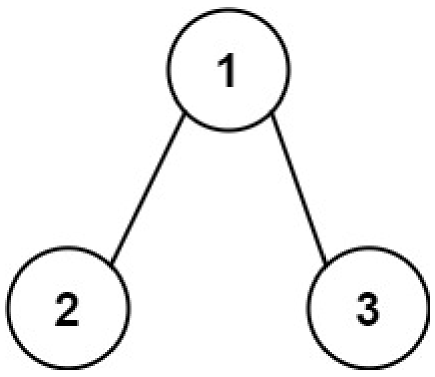
A **leaf** is a node with no children.

Example 1:



```
Input: root = [5,4,8,11,null,13,4,7,2,null,null,null,1], targetSum = 22
Output: true
Explanation: The root-to-leaf path with the target sum is shown.
```

Example 2:



```
Input: root = [1,2,3], targetSum = 5
Output: false
Explanation: There two root-to-leaf paths in the tree:
(1 --> 2): The sum is 3.
(1 --> 3): The sum is 4.
There is no root-to-leaf path with sum = 5.
```

Example 3:

```
Input: root = [], targetSum = 0
Output: false
Explanation: Since the tree is empty, there are no root-to-leaf paths.
```

Constraints:

- The number of nodes in the tree is in the range `[0, 5000]`.
- `-1000 <= Node.val <= 1000`
- `-1000 <= targetSum <= 1000`

```cpp
/**
 * Definition for a binary tree node.
 * struct TreeNode {
 *     int val;
 *     TreeNode *left;
 *     TreeNode *right;
 *     TreeNode() : val(0), left(nullptr), right(nullptr) {}
 *     TreeNode(int x) : val(x), left(nullptr), right(nullptr) {}
 *     TreeNode(int x, TreeNode *left, TreeNode *right) : val(x), left(left), right(right) {}
 * };
 */
class Solution {
public:
    bool hasPathSum(TreeNode* root, int sum) {
        if(root == nullptr){
            return false;
        }
        return has(root, 0, sum);
    }
    bool has(TreeNode* root, int cur, int sum){
        if(root->left == nullptr && root->right == nullptr){
            return sum == cur + root->val;
        }
        cur += root->val;
        bool res = false;
        if(root->left != nullptr){
            res = has(root->left, cur, sum);
        }
        if(res) return true;

        if(root->right != nullptr){
            res = has(root->right, cur, sum);
        }

        return res;
    }
};
```

Testcase Run Code Result Debugger

Accepted Runtime: 3 ms

Your input
```
[5,4,8,11,null,13,4,7,2,nul
22
```

Output
```
true
```
Diff

Expected
```
true
```