

Problem 437: Path Sum III

Problem Information

Difficulty: Medium

Acceptance Rate: 46.22%

Paid Only: No

Tags: Tree, Depth-First Search, Binary Tree

Problem Description

Given the `root` of a binary tree and an integer `targetSum`, return _the number of paths where the sum of the values along the path equals_ `targetSum`.

The path does not need to start or end at the root or a leaf, but it must go downwards (i.e., traveling only from parent nodes to child nodes).

Example 1:

Input: root = [10,5,-3,3,2,null,11,3,-2,null,1], targetSum = 8 **Output:** 3 **Explanation:**
The paths that sum to 8 are shown.

Example 2:

Input: root = [5,4,8,11,null,13,4,7,2,null,null,5,1], targetSum = 22 **Output:** 3

Constraints:

* The number of nodes in the tree is in the range `[0, 1000]`. * $-109 \leq \text{Node.val} \leq 109$ * $-1000 \leq \text{targetSum} \leq 1000$

Code Snippets

C++:

```
/**  
 * 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:  
    int pathSum(TreeNode* root, int targetSum) {  
  
    }  
};
```

Java:

```
/**  
 * Definition for a binary tree node.  
 * public class TreeNode {  
 *     int val;  
 *     TreeNode left;  
 *     TreeNode right;  
 *     TreeNode() {}  
 *     TreeNode(int val) { this.val = val; }  
 *     TreeNode(int val, TreeNode left, TreeNode right) {  
 *         this.val = val;  
 *         this.left = left;  
 *         this.right = right;  
 *     }  
 * }  
 */  
class Solution {  
    public int pathSum(TreeNode root, int targetSum) {  
  
    }  
}
```

Python3:

```
# Definition for a binary tree node.
# class TreeNode:
#     def __init__(self, val=0, left=None, right=None):
#         self.val = val
#         self.left = left
#         self.right = right
#     class Solution:
#         def pathSum(self, root: Optional[TreeNode], targetSum: int) -> int:
```