437. Path Sum III

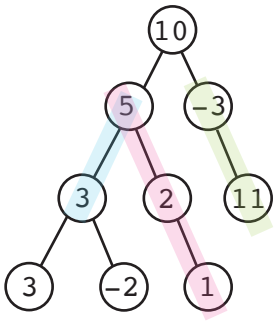
Medium

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).

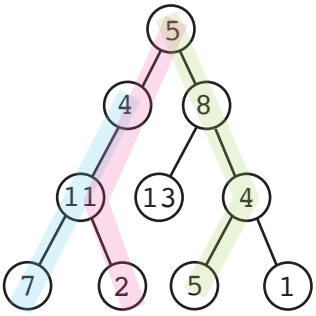
Example 1:

```
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
```




```python
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:
        self.values = defaultdict(int)
        self.values[0] = 1

        def dfs(node, total):
            count = 0
            if not node:
                return count
            total += node.val
            count = self.values[total - targetSum]

            self.values[total] += 1

            left = dfs(node.left, total)
            right = dfs(node.right, total)
            count += left + right

            self.values[total] -= 1
            return count

        answer = dfs(root, 0)
        return answer
```