Problem 113: Path Sum II

Problem Information

Difficulty: Medium
Acceptance Rate: 61.33%
Paid Only: No
Tags: Backtracking, Tree, Depth-First Search, Binary Tree

Problem Description

Given the `root` of a binary tree and an integer `targetSum`, return _all**root-to-leaf** paths where the sum of the node values in the path equals _`targetSum` _. Each path should be returned as a list of the node**values** , not node references_.

A **root-to-leaf** path is a path starting from the root and ending at any leaf node. A **leaf** is a node with no children.

Example 1:

Input: root = [5,4,8,11,null,13,4,7,2,null,null,5,1], targetSum = 22 **Output:** [[5,4,11,2],[5,8,4,5]] **Explanation:** There are two paths whose sum equals targetSum: 5 + 4 + 11 + 2 = 22 5 + 8 + 4 + 5 = 22

Example 2:

Input: root = [1,2,3], targetSum = 5 **Output:** []

Example 3:

Input: root = [1,2], targetSum = 0 **Output:** []

Constraints:

* The number of nodes in the tree is in the range `[0, 5000]`. * `-1000 <= Node.val <= 1000` * `-1000 <= targetSum <= 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:
vector<vector<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 List<List<Integer>> 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) ->
List[List[int]]:
```