Problem 1214: Two Sum BSTs

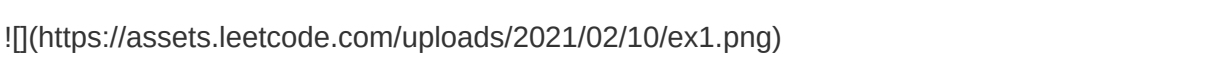
Problem Information

Difficulty: Medium
Acceptance Rate: 67.47%
Paid Only: Yes
Tags: Two Pointers, Binary Search, Stack, Tree, Depth-First Search, Binary Search Tree, Binary Tree

Problem Description

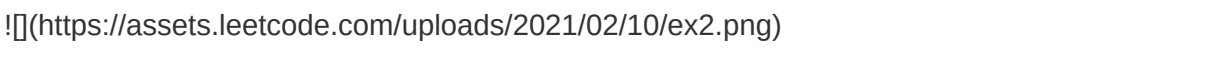
Given the roots of two binary search trees, `root1` and `root2`, return `true` if and only if there is a node in the first tree and a node in the second tree whose values sum up to a given integer `target`.

Example 1:



Input: root1 = [2,1,4], root2 = [1,0,3], target = 5 **Output:** true **Explanation:** 2 and 3 sum up to 5.

Example 2:



Input: root1 = [0,-10,10], root2 = [5,1,7,0,2], target = 18 **Output:** false

Constraints:

* The number of nodes in each tree is in the range `[1, 5000]`. * `-109 <= Node.val, target <= 109`

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:
bool twoSumBSTs(TreeNode* root1, TreeNode* root2, int target) {

}
};
```

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 boolean twoSumBSTs(TreeNode root1, TreeNode root2, int target) {

}
}
```

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 twoSumBSTs(self, root1: Optional[TreeNode], root2: Optional[TreeNode],
target: int) -> bool:
```