

# Problem 653: Two Sum IV - Input is a BST

## Problem Information

**Difficulty:** Easy

**Acceptance Rate:** 62.77%

**Paid Only:** No

**Tags:** Hash Table, Two Pointers, Tree, Depth-First Search, Breadth-First Search, Binary Search Tree, Binary Tree

## Problem Description

Given the `root` of a binary search tree and an integer `k`, return `true` if there exist two elements in the BST such that their sum is equal to `k`, or `false` otherwise.

**Example 1:**



**Input:** `root = [5,3,6,2,4,null,7]`, `k = 9` **Output:** `true`

**Example 2:**



**Input:** `root = [5,3,6,2,4,null,7]`, `k = 28` **Output:** `false`

**Constraints:**

\* The number of nodes in the tree is in the range `[1, 104]`. \* `-104 <= Node.val <= 104` \* `root` is guaranteed to be a **valid** binary search tree. \* `-105 <= k <= 105`

## 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 findTarget(TreeNode* root, int k) {

    }
};

```

## 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 findTarget(TreeNode root, int k) {

    }
}

```

### 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 findTarget(self, root: Optional[TreeNode], k: int) -> bool:
```