

# Two Sum IV – Input is a BST

## Leet Code:

</> Code

C v Auto


≡ 🔖 {} ↺ ↻

```
1  /**
2   * Definition for a binary tree node.
3   * struct TreeNode {
4   *     int val;
5   *     struct TreeNode *left;
6   *     struct TreeNode *right;
7   * };
8   */
9  void inorder(struct TreeNode* root, int* arr, int* size) {
10     if (root == NULL) return;
11     inorder(root->left, arr, size);
12     arr[(*size)++] = root->val;
13     inorder(root->right, arr, size);
14 }
15
16 bool findTarget(struct TreeNode* root, int k) {
17     if (root == NULL) return false;
18
19     int arr[10000]; // BST max nodes
20     int size = 0;
21
22     // Step 1: Get inorder traversal (sorted)
23     inorder(root, arr, &size);
24
25     // Step 2: Two pointer technique
26     int left = 0, right = size - 1;
27
28     while (left < right) {
29         int sum = arr[left] + arr[right];
30
31         if (sum == k)
32             return true;
33         else if (sum < k)
34             left++;
35         else
36             right--;
37     }
38
39     return false;
40 }
41
```

## Output:

### Test Case 1:

☒ Testcase | [Test Result](#)

**Accepted** Runtime: 0 ms 

☒ Case 1 ☒ Case 2

Input

root =  
[5,3,6,2,4,null,7]

k =  
9

Output


true

Expected

true

### Test Case 2:

☒ Testcase | [Test Result](#)

**Accepted** Runtime: 0 ms 

☒ Case 1 ☒ Case 2

Input

root =  
[5,3,6,2,4,null,7]

k =  
28

Output

false

Expected

false