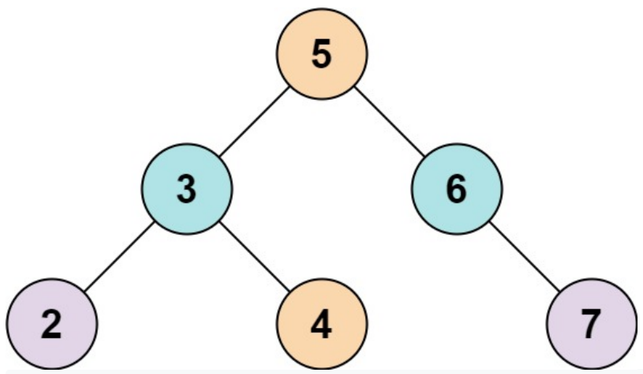
653. Two Sum IV - Input is a BST

Easy 4938 222 Add to List Share

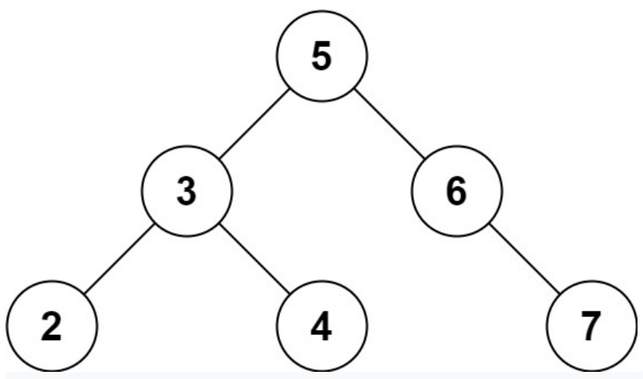
Given the `root` of a Binary Search Tree and a target number `k`, return `true` *if there exist two elements in the BST such that their sum is equal to the given target*.

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
```

Java Autocomplete

```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) {

        Stack<TreeNode> stackL = new Stack<TreeNode>();
        Stack<TreeNode> stackR = new Stack<TreeNode>();

        for(TreeNode cur = root; cur != null; cur = cur.left)
            stackL.push(cur);

        for(TreeNode cur = root; cur != null; cur = cur.right)
            stackR.push(cur);

        while(stackL.size() != 0 && stackR.size() != 0 && stackL.peek() != stackR.peek()){

            int tmpSum = stackL.peek().val + stackR.peek().val;

            if(tmpSum == k)
                return true;

            else if(tmpSum < k)
                for(TreeNode cur = stackL.pop().right; cur != null; cur = cur.left)
                    stackL.push(cur);
            else
                for(TreeNode cur = stackR.pop().left; cur != null; cur = cur.right)
                    stackR.push(cur);
        }

        return false;
    }
}
```