Description Solution Discuss (999+) Submissions

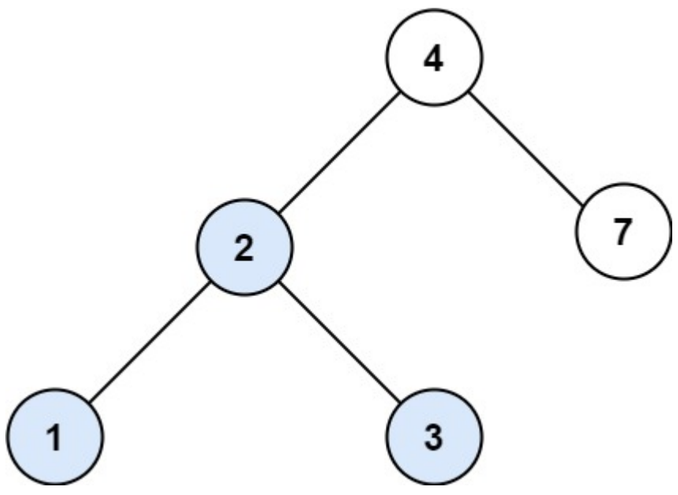
700. Search in a Binary Search Tree

Easy 2395 139 Add to List Share

You are given the `root` of a binary search tree (BST) and an integer `val`.

Find the node in the BST that the node's value equals `val` and return the subtree rooted with that node. If such a node does not exist, return `null`.

Example 1:



Input: root = [4,2,7,1,3], val = 2
Output: [2,1,3]

Example 2:

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 TreeNode searchBST(TreeNode root, int val) {
        if(root == null)
            return null;
        else if(root.val > val)
            return searchBST(root.left, val);
        else if(root.val < val)
            return searchBST(root.right, val);
        else
            return root;
    }
}
```

Testcase Run Code Result Debugger

Accepted Runtime: 0 ms

Your input
[4,2,7,1,3]
2

Output [2,1,3] Diff

Expected [2,1,3]