

Description Solution Discuss (999+) Submissions

C++ Autocomplete

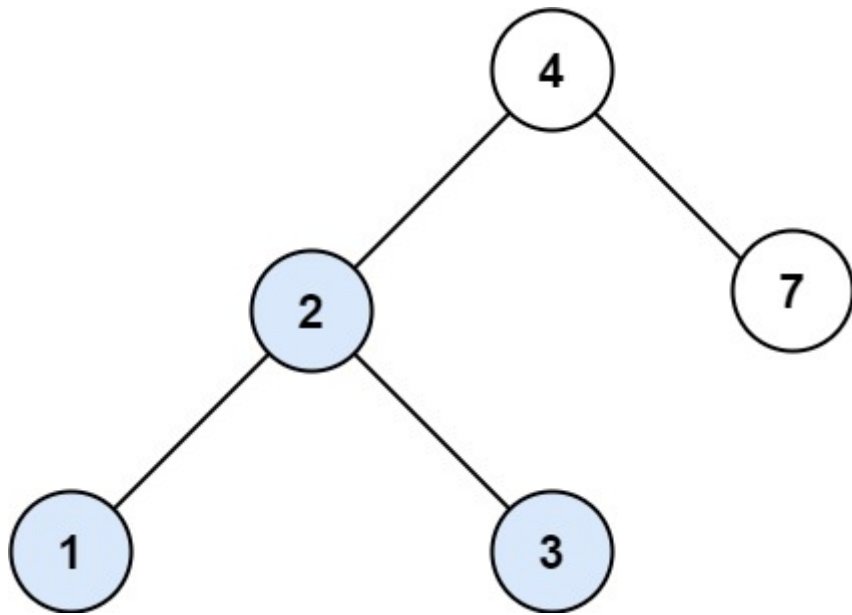
700. Search in a Binary Search Tree

Easy 3642 156 Add to List Share

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

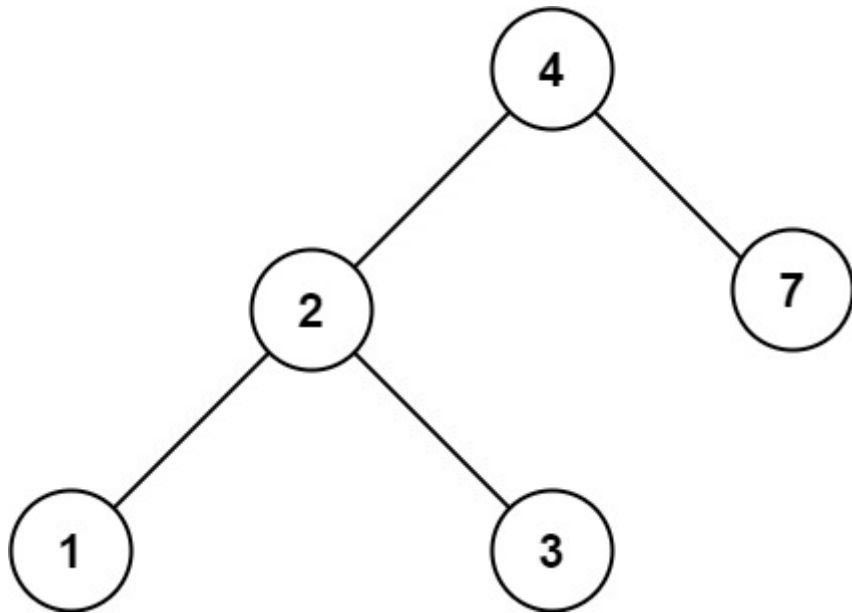
Find the node in the BST that the node's value equals `val` and return the subtree rooted with that node. If such a node does not exist, return `null`.

Example 1:



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

Example 2:



Input: root = [4,2,7,1,3], val = 5
Output: []

Constraints:

- The number of nodes in the tree is in the range `[1, 5000]`.
- `1 <= Node.val <= 107`
- `root` is a binary search tree.
- `1 <= val <= 107`

Accepted 520,902 Submissions 678,537

Seen this question in a real interview before? Yes No

Companies

Related Topics

```
1  /**
2   * Definition for a binary tree
3   * struct TreeNode {
4   *     int val;
5   *     TreeNode *left;
6   *     TreeNode *right;
7   *     TreeNode() : val(0),
8   *     left(nullptr), right(nullptr) {}
9   *     TreeNode(int x) : val(x),
10    *     left(nullptr), right(nullptr) {}
11    *     TreeNode(int x, TreeNode
12    *left, TreeNode *right) :
13    *val(x), left(left), right(right)
14    {}
15    * };
16    */
17    class Solution {
18    public:
19        TreeNode*
20        searchBST(TreeNode* root, int
21        value) {
22            if(root == nullptr)
23                return root;
24            if(value == root->val)
25                return root;
26            if(value < root->val)
27                return
28                searchBST(root->left, value);
29            return searchBST(root->
30            right, value);
31        }
32    };
33
```

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]