

# [每日 LeetCode] 700. Search in a Binary Search Tree

作者: [Hanseltu](#)

原文链接: <https://ld246.com/article/1558021019144>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

原文链接 [\[每日LeetCode\] 700. Search in a Binary Search Tree](#)

### Description:

Given the root node of a binary search tree (BST) and a value. You need to find the node in the BST that the node's value equals the given value. Return the subtree rooted with that node. If such node doesn't exist, you should return NULL.

For example,

Given the tree:



And the value to search: 2

You should return this subtree:



In the example above, if we want to search the value 5, since there is no node with value 5, we should return NULL.

Note that an empty tree is represented by NULL, therefore you would see the expected output (serialized tree format) as [], not null.

---

思路：本题要求在二叉搜索树中找出某个元素，并返回以此点为头结点的二叉树。比较简单的题，递归即可，遇到相等的值就返回其结点。

---

### C++代码

```
/**
 * Definition for a binary tree node.
 * struct TreeNode {
 *     int val;
 *     TreeNode *left;
 *     TreeNode *right;
 *     TreeNode(int x) : val(x), left(NULL), right(NULL) {}
 * };
 */
class Solution {
public:
    TreeNode* searchBST(TreeNode* root, int val) {
        if (!root)
            return NULL;
        int value = root->val;
```

```
    if (value == val)
        return root;
    else if (val < value)
        return searchBST(root->left, val);
    else if (val > value)
        return searchBST(root->right, val);
    return NULL;
}
};
```

---

运行时间: 48ms

运行内存: 29.3M