

07 判断二叉树是否为二叉排序树

24计算机考研成员一战成硕！



题目描述：

判断二叉树是否为二叉排序树

🕒 倒计时

1、知识点及难度



题解人：多动症男孩

难度：中下

知识点：

1. 二叉排序树的中序遍历结果应为升序排列，利用这个性质进行判断。
2. 在遍历过程中记录前一个节点的值，与当前节点的值进行比较，判断是否满足二叉排序树的条件。
 - 时间复杂度: $O(n)$
 - 空间复杂度: $O(n)$

2、算法题

思路

1. 对二叉树进行中序遍历，同时记录遍历的前一个节点的值。
2. 检查当前遍历的节点值是否大于前一个节点的值，如果大于，则继续遍历；如果小于或等于，则说明不满足二叉排序树的条件，返回 false。
3. 遍历完成后，如果没有发现不满足条件的节点，说明二叉树是二叉排序树，返回 true。

基本实现-C++

```
1 #include <iostream>
2 using namespace std;
3
4 // 辅助函数，中序遍历二叉树并检查是否属于升序序列
5 bool InOrder(TreeNode* root, int& prev) {
6     if(!root) return true;
7
8     // 遍历左子树
9     if(!InOrder(root->left,prev)) return false;
10
11     // 检查当前节点值是否大于前一个节点值
12     if (root->val <= prev) return false;
13     prev = root->val;
14
15     // 遍历右子树
16     return InOrder(root->right, prev);
17 }
18
19 // 判断是否为二叉排序树
20 bool isBST(TreeNode* root) {
21     if (root) {
22         int prev = INT_MIN; // 初始化为最小值
23     }else return false; // 空树
24 }
```

基本实现C

```
1 #include <stdbool.h>
2 #include <stdio.h>
3 #include <stdlib.h>
4
5 // 辅助函数，中序遍历二叉树并检查是否属于升序序列（参考王道风格，这里&属于混用）
6 bool InOrder(TreeNode* root, int& prev) {
7     if(!root) return true;
8
9     // 遍历左子树
10    if(!InOrder(root->left,prev)) return false;
11
12    // 检查当前节点值是否大于前一个节点值
13    if (root->val <= prev) return false;
14    prev = root->val;
15}
```

```
16    // 遍历右子树
17    return InOrder(root->right, prev);
18 }
19
20 // 判断是否为二叉排序树
21 bool isBST(TreeNode* root) {
22     if (root) {
23         int prev = INT_MIN; // 初始化为最小值
24         Inorder(root, prev);
25     } else return false; // 空树
26 }
```

3、总结



总结栏

蓝蓝B站首页: [蓝蓝希望你上岸呀B站首页](#)

蓝蓝公众号: [算法训练营9分计划](#)

蓝蓝知识星球介绍: [👁 关于知识星球的权益](#)

如何在星球打卡记录:

- 计算机考研数据结构算法专项day[1/60]:
- 学习内容: 最好能发出自己写的图片
- 遇到的问题: 如果无就不用写了
- 小结: 这部分一周写一次即可。