

11 找出二叉树中最大值的点

24计算机考研成员一战成硕!



题目描述:

找出二叉树中最大值的节点

🕒 倒计时

1、知识点及难度



题解人: 多动症男孩

难度: 中等

知识点:

1. 使用深度优先搜索算法遍历二叉树, 记录最大值节点的值和对应的节点指针。
2. 通过递归调用实现深度优先搜索, 并在遍历过程中更新最大值和最大值节点。
 - 时间复杂度: $O(n)$, n 为节点数
 - 空间复杂度: $O(n)$

2、算法题

思路

1. 初始化最大值变量为负无穷大, 最大值节点指针为空。
2. 从根节点开始进行深度优先搜索:
 - 如果当前节点的值大于最大值, 则更新最大值和最大值节点指针。
 - 递归调用遍历左子树。
 - 递归调用遍历右子树。
3. 返回最大值节点指针。

基本实现-C++

```
1 #include <iostream>
2 #include <limits>
3
4 using namespace std;
5
6 // 二叉树节点结构
7 struct TreeNode {
8     int val;
9     TreeNode* left;
10    TreeNode* right;
11 };
12
13 // 深度有限搜索函数
14 void DFS(TreeNode* node, TreeNode* &maxNode, int& maxVal) {
15     if (!node) return;
16
17     // 寻找最大值节点
18     if (node->val > maxVal){
19         maxVal = node->val;
20         maxNode = node;
21     }
22
23     // 递归左右子树
24     DFS(node->left, maxNode, maxVal);
25     DFS(node->right, maxNode, maxVal);
26 }
27
28 // 找出二叉树中的最大值的节点
29 TreeNode* findMaxNode(TreeNode* root) {
30     if(!root) return nullptr;
31     TreeNode* maxNode = nullptr; // 最大值节点
32     int maxVal = numeric_limits<int>::min(); // 初始化为最小值
33
34     DFS(root, maxNode, maxVal);
35     return maxNode;
36 }
```

基本实现C

```
1 #include <stdio.h>
2 #include <limits.h>
```

```

3
4 // 二叉树节点结构
5 struct TreeNode {
6     int val;
7     TreeNode* left;
8     TreeNode* right;
9 };
10
11 // 深度有限搜索函数
12 void DFS(TreeNode* node, TreeNode** maxNode, int* maxVal) {
13     if (!node) return;
14
15     // 寻找最大值节点
16     if (node->val > *maxVal){
17         *maxVal = node->val;
18         *maxNode = node;
19     }
20
21     // 递归左右子树
22     DFS(node->left, maxNode, maxVal);
23     DFS(node->right, maxNode, maxVal);
24 }
25
26 // 找出二叉树中的最大值的节点
27 TreeNode* findMaxNode(struct TreeNode* root) {
28     if(!root) return NULL;
29     struct TreeNode* maxNode = NULL; // 最大值节点
30     int maxVal = INT_MIN; // 初始化为最小值
31
32     DFS(root, &maxNode, &maxVal);
33     return maxNode;
34 }

```

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。