

01 求二叉树的最大深度

24计算机考研成员一战成硕!



题目描述:

求二叉树的最大深度

🕒 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. 使用递归的方式来求解二叉树的最大深度, 将问题分解为子问题。
2. 在递归过程中要注意递归结束的条件, 即空树的深度为0。

时间复杂度: $O(n)$, 其中 n 为二叉树的节点个数。

空间复杂度: $O(n)$, 最坏情况下二叉树退化成了链表, 则递归栈的深度为 n 。

2、算法题

思路

1. 如果树为空, 则深度为0。
2. 否则, 通过递归求解左子树和右子树的深度, 取较大值。
3. <https://juejin.cn/post/7211936663797907516> (需要可上机代码的同学, 可以看这个)

基本实现-C++

```
1 #include <iostream>
```

```

2 using namespace std;
3
4 // 定义二叉树节点
5 typedef struct BiTNode {
6     Element data; // 数据域
7     BiTNode *left, *right; // 左右节点
8 }BiTNode, *BiTree;
9
10 // 计算二叉树深度
11 int TreeDepth(BiTree T) {
12     if(!T) return 0; // 空树深度为0
13     // 递归计算左右子树的深度
14     int LDepth = TreeDepth(T->left);
15     int RDepth = TreeDepth(T->right);
16     return LDepth > RDepth ? LDepth + 1 : RDepth + 1;
17 }

```

基本实现C

```

1 #include <stdio.h>
2 #include <stdlib.h>
3
4 // 定义二叉书节点
5 typedef struct BiTNode {
6     Element data; // 数据域
7     BiTNode *left, *right; // 左右节点
8 }BiTNode, *BiTree;
9
10 // 计算二叉树深度
11 int TreeDepth(BiTree T) {
12     if(!T) return 0; // 空树深度为0
13     // 递归计算左右子树的深度
14     int LDepth = TreeDepth(T->left);
15     int RDepth = TreeDepth(T->right);
16     return LDepth > RDepth ? LDepth + 1 : RDepth + 1;
17 }

```

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。