

剑指Offer（三十八）：二叉树的深度

🕒 2018年1月16日 11:27:42 🗂 5 🌡 6,683 °C 🛠 编辑



一、前言

本系列文章为《剑指Offer》刷题笔记。

刷题平台：[牛客网](#)

书籍下载：[共享资源](#)

二、题目

输入一棵二叉树，求该树的深度。从根结点到叶结点依次经过的结点（含根、叶结点）形成树的一条路径，最长路径的长度为树的深度。

1、思路

这道题蛮简单的，求二叉树的深度。可以是递归的方法，属于DFS（深度优先搜索）；另一种方法是按照层次遍历，属于BFS（广度优先搜索）。

2、代码

C++:

DFS方法:

C+

```
1  /*
2  struct TreeNode {
3      int val;
4      struct TreeNode *left;
5      struct TreeNode *right;
6      TreeNode(int x) :
7          val(x), left(NULL), right(NULL) {
8      }
9  };*/
10 class Solution {
11 public:
12     int TreeDepth(TreeNode* pRoot)
13     {
14         if(pRoot == NULL){
15             return 0;
16         }
17         int left = TreeDepth(pRoot->left);
18         int right = TreeDepth(pRoot->right);
19         return (left > right) ? (left + 1) : (right + 1);
20     }
21 };
```

BFS方法:

C+

```
1  /*
2  struct TreeNode {
3      int val;
4      struct TreeNode *left;
5      struct TreeNode *right;
6      TreeNode(int x) :
7          val(x), left(NULL), right(NULL) {
8      }
9  };*/
10 class Solution {
11 public:
12     int TreeDepth(TreeNode* pRoot)
13     {
```

```
14     if(pRoot == NULL){
15         return 0;
16     }
17     queue<TreeNode*> que;
18     int depth = 0;
19     que.push(pRoot);
20     while(!que.empty()){
21         int size = que.size();
22         depth++;
23         for(int i = 0; i < size; i++){
24             TreeNode* node = que.front();
25             que.pop();
26             if(node->left){
27                 que.push(node->left);
28             }
29             if(node->right){
30                 que.push(node->right);
31             }
32         }
33     }
34     return depth;
35 }
36 };
```

感谢@小小毛提供的本地测试用例：

```
1 class TreeNode:
2     def __init__(self, x):
3         self.val = x
4         self.left = None
5         self.right = None
6
7 class Solution:
8     def TreeDepth(self, root):
9         # write code here
10        if root is None:
11            return 0
12        left=self.TreeDepth(root.left)
13        right=self.TreeDepth(root.right)
14        print(left,right)
15        return max(left,right)+1
16
17 if __name__=='__main__':
18     A1 = TreeNode(1)
19     A2 = TreeNode(2)
20     A3 = TreeNode(3)
21     A4 = TreeNode(4)
22     A5 = TreeNode(5)
23     A6 = TreeNode(6)
24
25     A1.left=A2
26     A1.right=A3
27     A2.left=A4
28     A2.right=A5
29     A4.left=A6
30
31     solution=Solution()
32     ans=solution.TreeDepth(A1)
33     print('ans=',ans)
```

Pytho



微信公众号

分享技术，乐享生活：微信公众号搜索

「JackCui-AI」关注一个在互联网摸爬滚
打的潜行者。

不在其位，不谋其政。--- 孔子