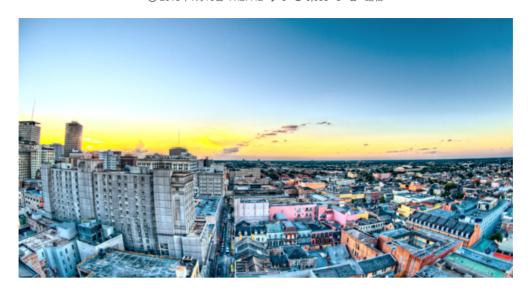
剑指Offer(三十八):二叉树的深度

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

BFS方法:

感谢@小小毛提供的本地测试用例:

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



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