



love to lea...

谁说学通信的女孩当程序媛！

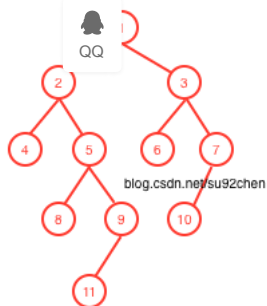


RSS订阅

# python二叉树遍历、求深度、已知前序中序 求树 求后序

2017年04月19日 16:15

阅读数：1349



前序遍历结果：1, 2, 4, 5, 8, 9, 11, 3, 6, 7, 10

中序遍历结果：4, 2, 8, 5, 11, 9, 1, 6, 3, 10, 7

后序遍历结果：4, 8, 11, 9, 5, 2, 6, 10, 7, 3, 1

根节点表示：

```
1 class Node(object):
2     def __init__(self, value=None, left=None, right=None):
3         self.value = value
4         self.left = left
5         self.right = right
6
7 if __name__ == "__main__":
8     tree = Node(1, Node(2, Node(4), Node(5, Node(8), Node(9, left=Node(11)))), Node(3, Node(6), Node(7, left=Node(10))))
```

## 1、深度优先遍历

### 1.1 递归

```
1 # 前序遍历（递归）
2 def pre_deep_func(root):
3     if root is None:
4         return
5     print(root.value) # print 放到下一行 就是中序遍历，放到最后 就是后序遍历
6     pre_deep_func(root.left)
7     pre_deep_func(root.right)
```

### 1.2 非递归

```
1 # 前序遍历（根左右）：模拟压栈过程
2 # 入栈之前读（根、左），这样出栈时再读右（也是右结点子节点们的根）
3 def pre_deep_func2(root):
4     a = []
5     while a or root:
6         while root:
7             print root.value
8             a.append(root)
9             root = root.left
10        h = a.pop()
```

python薪资多少？

大数据薪资多少？

AI薪资多少？

登录

注册

×

```

14 def mid_deep_func2(root):
15     a = []
16     while a or root:
17         if root:
18             a.append(root)
19             root = root.left
20         else:
21             root = a.pop()
22             int h.value
23             root = h.right
24 # 后序遍历 (左右根) : 模拟逆序(根右左)存入数组b, 然后再数组b逆序输出
25 # (根 与(根左右)类似, 入栈a前读(根、右), 出栈后指针变更再读左
26 def deep_func2(root):
27     a = []
28     while a or root:
29         if root:
30             b.append(root.value)
31             a.append(root)
32             root = root.right
33         else:
34             h = a.pop()
35             root = h.left
36     print b[::-1]

```

## 2、广度优先遍历

```

1 def level_func(root):
2     a = []
3     a.append(root)
4     while a:
5         head = a.pop(0)
6         print head.value
7         if head.left:
8             a.append(head.left)
9         if head.right:
10            a.append(head.right)

```

## 3、求深度

### 3.1 利用广度优先遍历

```

1 def get_level_func1(root):
2     a = []
3     b = []
4     a.append(root)
5     b.append(1)
6     while a:
7         head = a.pop(0)
8         p = b.pop(0)
9         if head.left:
10            a.append(head.left)
11            b.append(p+1)
12         if head.right:
13            a.append(head.right)
14            b.append(p+1)
15     return p

```

### 3.2 递归方法

```

1 def get_level_func2(root):
2     if not root:
3         return 0
4     left = right = 0
5     left = get_level_func2(root.left)
6     right = get_level_func2(root.right)
7     return max(left, right) + 1

```

## 4、已知二叉树前序中序, 还原二叉树

python薪资多少？

大数据薪资多少？

AI薪资多少？

登录

注册

```
1 def get_tree(pre, mid):
2     if len(pre) == 0:
3         turn None
4     if len(pre) == 1:
5         turn Node(pre[0])
6     root = Node(pre[0])
7     root.index = mid.index(pre[0])
8     root.left = get_tree(pre[1:root.index + 1], mid[:root.index])
9     root.right = get_tree(pre[root.index + 1:], mid[root.index + 1:])
10    return root
11
12 # head = get_tree([1, 2, 4, 5, 8, 9, 11, 3, 6, 7, 10], [4, 2, 8, 5, 11, 9, 1, 6, 3, 10, 7])
13 # pre_order_func(head)
```

5、已知前序中序，求后序

```
1 def get_after_deep(pre, mid, a):
2     if len(pre) == 1:
3         a.append(pre[0])
4         return
5     if len(pre) == 0:
6         return
7     root = pre[0]
8     root_index = mid.index(root)
9     get_after_deep(pre[1:root_index+1], mid[:root_index], a)
10    get_after_deep(pre[root_index+1:], mid[root_index+1:], a)
11    a.append(root)
12    return a
13
14 # res = get_after_deep([1, 2, 4, 5, 8, 9, 11, 3, 6, 7, 10], [4, 2, 8, 5, 11, 9, 1, 6, 3, 10, 7], [])
15 # res = [4, 8, 11, 9, 5, 2, 6, 10, 7, 3, 1]
```

版权声明：<绝对原创> 转发注明出处a亲 ~ ~ <https://blog.csdn.net/su92chen/article/details/70242822>

文章标签： 二叉树 遍历 python

个人分类： 算法



**【CSDN学院】Web全栈工程师最新攻略！**

你是组件开发学不会？框架核心学不会？小程序开发学不会？别犹豫了，看看我是怎么帮你学会的！

马上了解 >>

想对作者说点什么？ 我来说两句

树的先序，中序，后序遍历 | Python

一直觉得遍历先序，中序，后序遍历一棵树，总会记混。后来才发现用代码来记忆，然后反推遍历的结果更加靠谱。 class Solution(object): def inorderTraver...

u012560212 2017-06-13 13:59:36 阅读数：1273

二叉树的前序、中序、后序遍历的非递归方法 python

class TreeNode: def \_\_init\_\_(self,value=None,leftNode=None,rightNode=None): self.value =...

l153097889 2015-09-09 09:02:00 阅读数：3603

Python实现二叉树的深度 - CSDN博客

python实现二叉树的深度搜索 class TreeNode(object): def \_\_init\_\_(self, x...求深度用递归;求宽度用队列;然后把每层的宽度求出来,找出最大的就是二叉树的...

2018-7-3