

# Assignment #5: "树"算：概念、表示、解析、遍历

Updated 2124 GMT+8 March 17, 2024

2024 spring, Compiled by 杨乐山 物理学院 2100011502

## 说明：

1) The complete process to learn DSA from scratch can be broken into 4 parts:

Learn about Time complexities, learn the basics of individual Data Structures, learn the basics of Algorithms, and practice Problems.

2) 请把每个题目解题思路（可选），源码Python, 或者C++（已经在Codeforces/Openjudge上AC），截图（包含Accepted），填写到下面作业模版中（推荐使用 typora <https://typoraio.cn>，或者用word）。AC 或者没有AC，都请标上每个题目大致花费时间。

3) 提交时候先提交pdf文件，再把md或者doc文件上传到右侧“作业评论”。Canvas需要有同学清晰头像、提交文件有pdf、“作业评论”区有上传的md或者doc附件。

4) 如果不能在截止前提交作业，请写明原因。

## 编程环境

操作系统：Windows 11 专业版 23H2 22631.3296

Python编程环境：PyCharm 2023.3.5 (Professional Edition)

## 1. 题目

### 27638: 求二叉树的高度和叶子数目

<http://cs101.openjudge.cn/practice/27638/>

思路：正在快马加鞭地赶作业QAQ

代码

```
class TreeNode:
    def __init__(self, val):
        self.val = val
        self.left = None
        self.right = None

def build_tree(nodes):
    if not nodes:
        return None
    tree = [TreeNode(i) for i in range(len(nodes))]
    for i, (left, right) in enumerate(nodes):
```

```

        if left != -1:
            tree[i].left = tree[left]
        if right != -1:
            tree[i].right = tree[right]
    return tree[0]

def tree_height_and_leaves(root):
    if not root:
        return 0, 0

    stack = [(root, 1)]
    max_height = 0
    leaves_count = 0

    while stack:
        node, height = stack.pop()
        max_height = max(max_height, height)

        if not node.left and not node.right:
            leaves_count += 1

        if node.left:
            stack.append((node.left, height + 1))
        if node.right:
            stack.append((node.right, height + 1))

    return max_height, leaves_count+1

# 读取输入
n = int(input())
nodes = [tuple(map(int, input().split())) for _ in range(n)]

# 构建二叉树
root = build_tree(nodes)

# 计算二叉树高度和叶子节点个数
height, leaves = tree_height_and_leaves(root)

# 输出结果
print(height, leaves)

```

代码运行截图 (至少包含有"Accepted")

## 24729: 括号嵌套树

<http://cs101.openjudge.cn/practice/24729/>

思路:

代码

```
#
```

代码运行截图 (至少包含有"Accepted")

## 02775: 文件结构“图”

<http://cs101.openjudge.cn/practice/02775/>

思路:

代码

```
#
```

代码运行截图 (AC代码截图, 至少包含有"Accepted")

## 25140: 根据后序表达式建立队列表达式

<http://cs101.openjudge.cn/practice/25140/>

思路:

代码

```
#
```

代码运行截图 (AC代码截图, 至少包含有"Accepted")

## 24750: 根据二叉树中后序序列建树

<http://cs101.openjudge.cn/practice/24750/>

思路:

代码

```
#
```

代码运行截图 (AC代码截图, 至少包含有"Accepted")

## 22158: 根据二叉树前中序序列建树

<http://cs101.openjudge.cn/practice/22158/>

思路:

代码

```
#
```

代码运行截图 (AC代码截图, 至少包含有"Accepted")

## 2. 学习总结和收获

---

如果作业题目简单，有否额外练习题目，比如：OJ“2024spring每日选做”、CF、LeetCode、洛谷等网站题目。