Assignment #5: "树"算：概念、表示、解析、遍历

Updated 2124 GMT+8 March 17, 2024

2024 spring, Complied by ==王一粟 经济学院==

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

==（请改为同学的操作系统、编程环境等）==

操作系统：macOS Ventura 13.4.1 (c)

Python编程环境：Spyder IDE 5.2.2, PyCharm 2023.1.4 (Professional Edition)

C/C++编程环境：Mac terminal vi (version 9.0.1424), g++/gcc (Apple clang version 14.0.3, clang-1403.0.22.14.1)

1. 题目

27638: 求二叉树的高度和叶子数目

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

思路：经典题目，递归处理

耗时：25min

代码

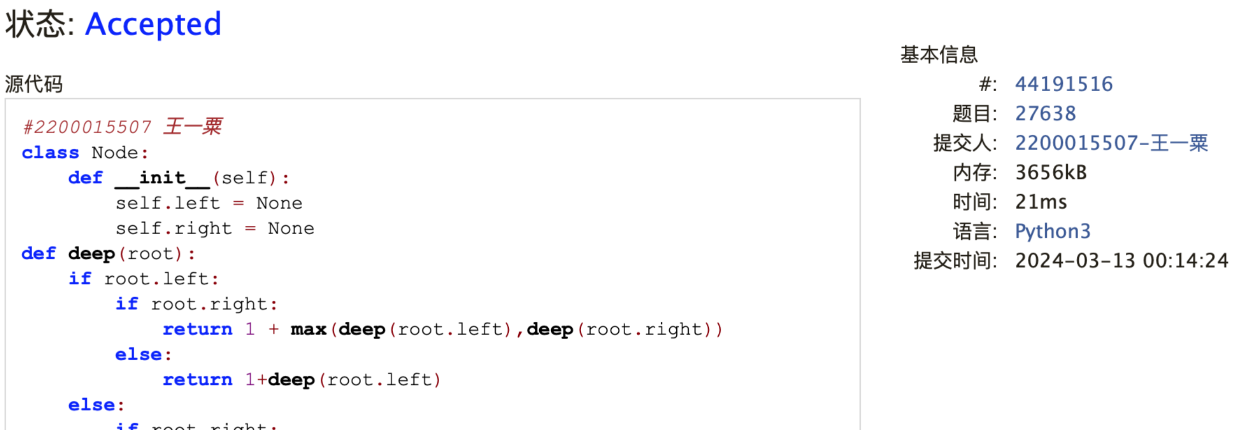
```
#2200015507 王一粟
class Node:
    def __init__(self):
        self.left = None
        self.right = None
def deep(root):
    if root.left:
        if root.right:
            return 1 + max(deep(root.left),deep(root.right))
        else:
            return 1+deep(root.left)
    else:
        if root.right:
            return 1+deep(root.right)
        else:
            return 1

def count(root):
    if root.left:
        if root.right:
            return count(root.left)+count(root.right)
        else:
            return count(root.left)
    else:
        if root.right:
            return count(root.right)
        else:
            return 1

n = int(input())
mylist = [Node() for i in range(n)]
isparent = [False]*n
for i in range(n):
    leftnum,rightnum = [int(i) for i in input().split()]
    if leftnum != -1:
        mylist[i].left = mylist[leftnum]
        isparent[leftnum] = True
    if rightnum != -1:
        mylist[i].right = mylist[rightnum]
        isparent[rightnum] = True

root_index = isparent.index(False)
height = deep(mylist[root_index]) - 1
leaf_num = count(mylist[root_index])
print(height,leaf_num)
```

代码运行截图 ==（至少包含有"Accepted"）==



状态: Accepted

源代码

```
#2200015507 王一粟
class Node:
    def __init__(self):
        self.left = None
        self.right = None
def deep(root):
    if root.left:
        if root.right:
            return 1 + max(deep(root.left),deep(root.right))
        else:
            return 1+deep(root.left)
    else:
        if root.right:
```

基本信息

#: 44191516
题目: 27638
提交人: 2200015507-王一粟
内存: 3656kB
时间: 21ms
语言: Python3
提交时间: 2024-03-13 00:14:24

24729: 括号嵌套树

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

思路：每个结点有两个标记，一个是其值，一个是其子结点的索引。用栈存储父结点。本题的难点在于敲定父结点的位置。

耗时：30min

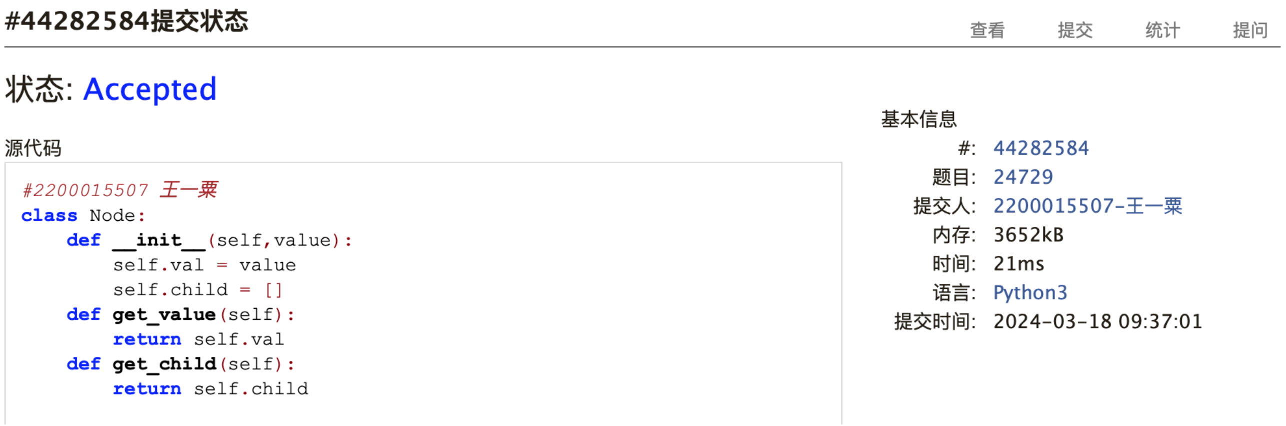
代码

```
#2200015507 王一粟
class Node:
    def __init__(self,value):
        self.val = value
        self.child = []
    def get_value(self):
        return self.val
    def get_child(self):
        return self.child

s = input()
root_node = Node(s[0])
stack = []
stack.append(root_node)
cnt = 0
for char in s[1:]:
    if char == "(":
        parent_node = stack[-1]
    elif char == ")":
        stack.pop()
    elif char == ",":
        stack.pop()
        parent_node = stack[-1]
    else:
        my_node = Node(char)
        parent_node.child.append(my_node)
        stack.append(my_node)
```

```
def pre_order(root):
    if root.get_child() == []:
        return [root.get_value()]
    else:
        result_list = [root.get_value()]
        for element in root.get_child():
            result_list.extend(pre_order(element))
        return result_list
def post_order(root):
    if root.get_child() == []:
        return [root.get_value()]
    else:
        result_list = []
        for element in root.get_child():
            result_list.extend(post_order(element))
        result_list.append(root.get_value())
        return result_list
print("".join(pre_order(root_node)))
print("".join(post_order(root_node)))
```

代码运行截图 ==（至少包含有"Accepted"）==



#44282584提交状态

状态: Accepted

源代码

```
#2200015507 王一粟
class Node:
    def __init__(self,value):
        self.val = value
        self.child = []
    def get_value(self):
        return self.val
    def get_child(self):
        return self.child
```

基本信息

#: 44282584
题目: 24729
提交人: 2200015507-王一粟
内存: 3652kB
时间: 21ms
语言: Python3
提交时间: 2024-03-18 09:37:01

02775: 文件结构“图”

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

思路：可以考虑用树的思想做处理。每个文件夹下都有文件的列表和文件夹的列表。对于每个文件夹，用列表存储其输出内容，再递归处理即可。

耗时：25min

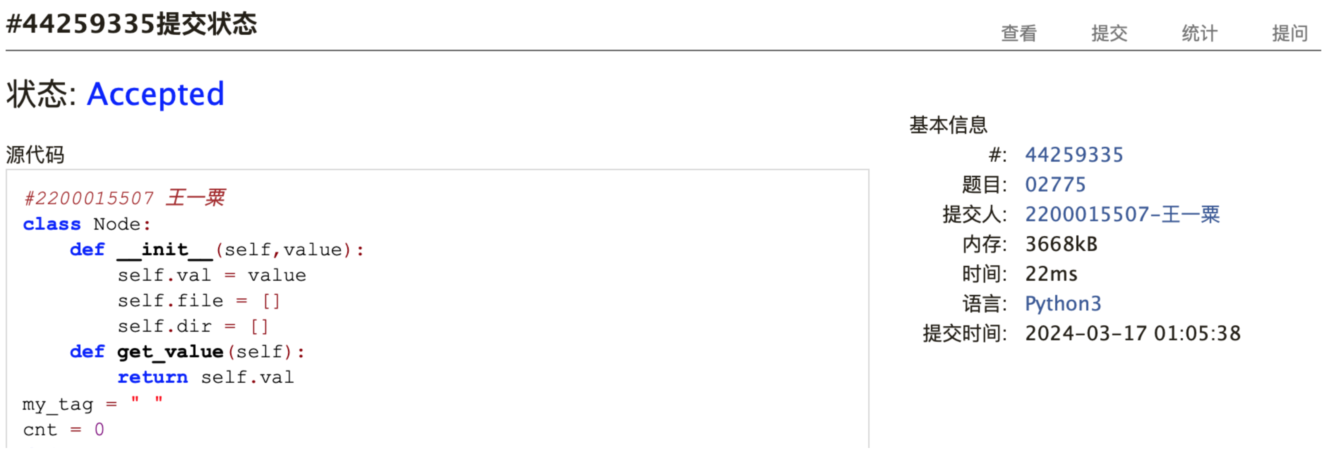
代码

```
#2200015507 王一粟
class Node:
    def __init__(self,value):
        self.val = value
        self.file = []
        self.dir = []
```

```python
    def get_value(self):
        return self.val
my_tag = " "
cnt = 0
def output(my_root):
    file_doc = sorted(my_root.file, key = lambda x:x.get_value())
    dir_doc = my_root.dir
    result_list = []
    for element in dir_doc:
        result_list.append("|"+5*my_tag+element.get_value())
        addi = ["|"+5*my_tag + i for i in output(element)]
        result_list.extend(addi)
    for element in file_doc:
        result_list.append(element.get_value())
    return result_list

while True:
    cnt += 1
    s = input()
    root_node = Node("root")
    stack = [root_node]
    if s == "#":
        break
    if s[0] == "f":
        current_node = Node(s)
        root_node.file.append(current_node)
    else:
        current_node = Node(s)
        root_node.dir.append(current_node)
        stack.append(current_node)
    while True:
        s = input()
        if s == "*":
            break
        if s[0] == "f":
            current_node = Node(s)
            stack[-1].file.append(current_node)
        elif s[0] == "d":
            current_node = Node(s)
            stack[-1].dir.append(current_node)
            stack.append(current_node)
        else:
            stack.pop()
    if cnt != 1:
        print()
    print(f"DATA SET {cnt}:")
    print("ROOT")
    end_list = output(root_node)
    for content in end_list:
        print(content)
```

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==



#44259335提交状态

状态: Accepted

源代码

```
#2200015507 王一粟
class Node:
    def __init__(self,value):
        self.val = value
        self.file = []
        self.dir = []
    def get_value(self):
        return self.val
my_tag = " "
cnt = 0
```

基本信息

#: 44259335
题目: 02775
提交人: 2200015507-王一粟
内存: 3668kB
时间: 22ms
语言: Python3
提交时间: 2024-03-17 01:05:38

25140: 根据后序表达式建立队列表达式

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

思路：经典题目。代码块两个方面构成，一个是通过后序表达式解析树，具体方式后续表达式求值逻辑相同，用栈处理操作数（和操作符），碰到操作符就确定为其子结点，操作符再返回栈中。队列表达式的输出恰为遍历结果的颠倒。

耗时：25min

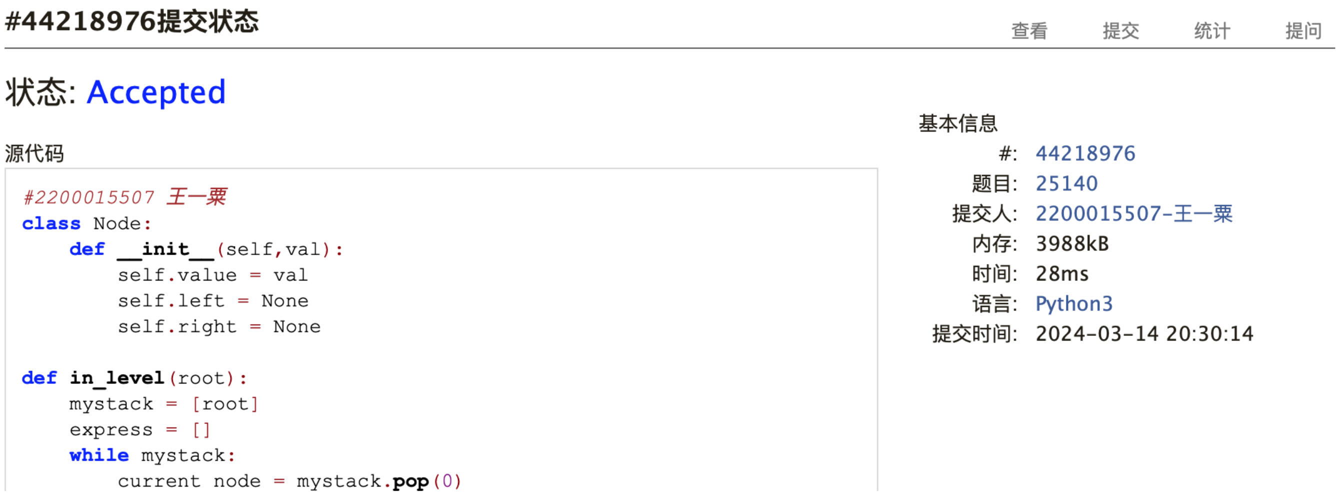
代码

```
#2200015507 王一粟
class Node:
    def __init__(self,val):
        self.value = val
        self.left = None
        self.right = None

def in_level(root):
    mystack = [root]
    express = []
    while mystack:
        current_node = mystack.pop(0)
        express.append(current_node.value)
        if current_node.left:
            mystack.append(current_node.left)
            mystack.append(current_node.right)
    return "".join(reversed(express))

n = int(input())
for _ in range(n):
    s = input()
    stack = []
    for char in s:
        if char.islower():
            my_node = Node(char)
            stack.append(my_node)
        else:
            my_node = Node(char)
            my_node.right = stack.pop()
            my_node.left = stack.pop()
            stack.append(my_node)
    root_node = stack.pop()
    print(in_level(root_node))
```

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==



#44218976提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
#2200015507 王一粟
class Node:
    def __init__(self,val):
        self.value = val
        self.left = None
        self.right = None

def in_level(root):
    mystack = [root]
    express = []
    while mystack:
        current_node = mystack.pop(0)
```

基本信息

#:	44218976
题目:	25140
提交人:	2200015507-王一粟
内存:	3988kB
时间:	28ms
语言:	Python3
提交时间:	2024-03-14 20:30:14

24750: 根据二叉树中后序序列建树

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

思路：给定中序序列和后序序列，可通过后序序列最后一个值敲定根结点，并据此确定中序序列的左右子树，再反推后序序列的左右子树，递归处理。学习到的一个地方在于对一个列表做切片，如果索引越界或者索引的长度为0，会直接返回空列表（但切片越界报错）

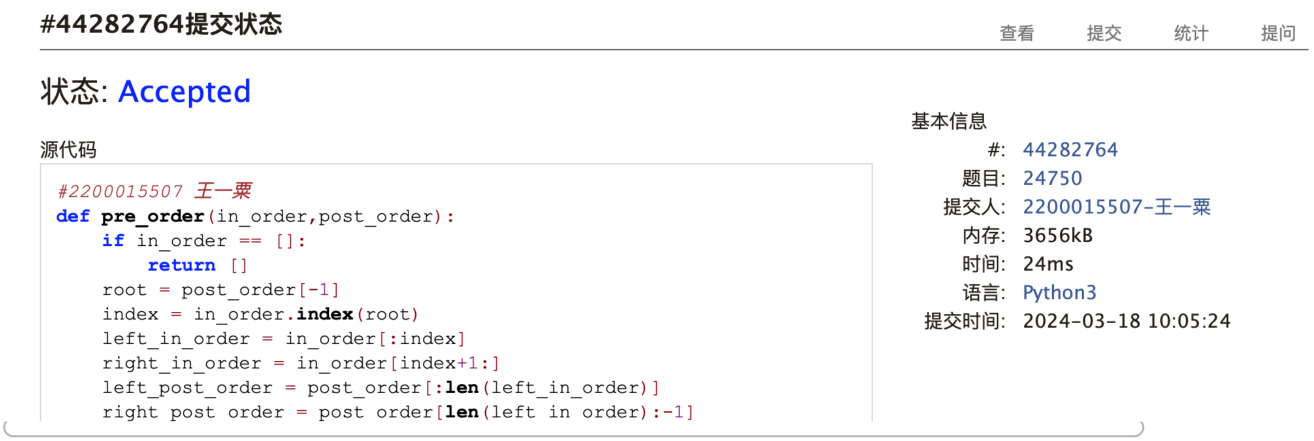
耗时：15min

代码

```
#2200015507 王一粟
def pre_order(in_order,post_order):
    if in_order == []:
        return []
    root = post_order[-1]
    index = in_order.index(root)
    left_in_order = in_order[:index]
    right_in_order = in_order[index+1:]
    left_post_order = post_order[:len(left_in_order)]
    right_post_order = post_order[len(left_in_order):-1]
    result_list = [root] + pre_order(left_in_order,left_post_order)+pre_order(right_in_order,
    return result_list
in_order = list(input())
```

```
post_order = list(input())
print("".join(pre_order(in_order,post_order)))
```

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==



#44282764提交状态

状态: Accepted

源代码

```
#2200015507 王一粟
def pre_order(in_order,post_order):
    if in_order == []:
        return []
    root = post_order[-1]
    index = in_order.index(root)
    left_in_order = in_order[:index]
    right_in_order = in_order[index+1:]
    left_post_order = post_order[:len(left_in_order)]
    right_post_order = post_order[len(left_in_order):-1]
```

基本信息

#:	44282764
题目:	24750
提交人:	2200015507-王一粟
内存:	3656kB
时间:	24ms
语言:	Python3
提交时间:	2024-03-18 10:05:24

22158: 根据二叉树前中序序列建树

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

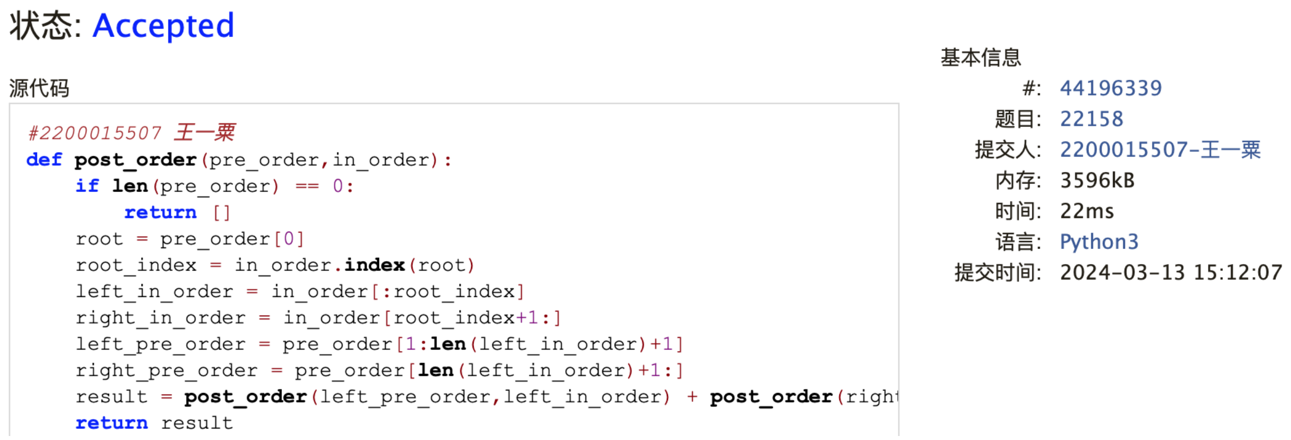
思路：给定前序序列和中序序列，可以通过前序序列的第一个值确定根结点，对应中序序列分开左子树和右子树，再根据子树长度确定前序序列中的左子树和右子树，再递归处理。

耗时：20min

代码

```
#2200015507 王一粟
def post_order(pre_order,in_order):
    if len(pre_order) == 0:
        return []
    root = pre_order[0]
    root_index = in_order.index(root)
    left_in_order = in_order[:root_index]
    right_in_order = in_order[root_index+1:]
    left_pre_order = pre_order[1:len(left_in_order)+1]
    right_pre_order = pre_order[len(left_in_order)+1:]
    result = post_order(left_pre_order,left_in_order) + post_order(right_pre_order,right_in_orde
    return result
while True:
    try:
        pre_order = list(input())
        in_order = list(input())
        result_list = post_order(pre_order,in_order)
        print("".join(result_list))
    except:
        break
```

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==



状态: Accepted

源代码

```
#2200015507 王一粟
def post_order(pre_order,in_order):
    if len(pre_order) == 0:
        return []
    root = pre_order[0]
    root_index = in_order.index(root)
    left_in_order = in_order[:root_index]
    right_in_order = in_order[root_index+1:]
    left_pre_order = pre_order[1:len(left_in_order)+1]
    right_pre_order = pre_order[len(left_in_order)+1:]
    result = post_order(left_pre_order,left_in_order) + post_order(right
    return result
```

基本信息

#:	44196339
题目:	22158
提交人:	2200015507-王一粟
内存:	3596kB
时间:	22ms
语言:	Python3
提交时间:	2024-03-13 15:12:07

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。==

这周有点忙（党校考试&几场实习面试），本计概C选手想学老师递归的，结果没完成，这周末补上。

树至少到目前为止讲过的内容感觉掌握的还算可以。每日选做一直在跟，没落下。