Assignment #5: "树"算：概念、表示、解析、遍历

Updated 2124 GMT+8 March 17, 2024

2024 spring, Complied by 赵策 数学科学学院

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

操作系统：Windows 11

Python编程环境：Visual Studio Code 1.86.2

1. 题目

27638: 求二叉树的高度和叶子数目

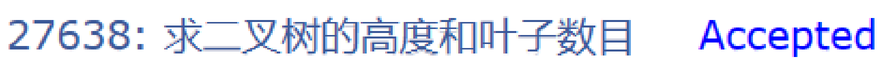
http://cs101.openjudge.cn/practice/27638/

思路：

代码

```
#
n=int(input())
dp=[0]*n
leaf_count=0
d={}
for _ in range(n):
    a,b=map(int,input().split())
    d[_]=a,b
    if [a,b]==[-1,-1]:
        leaf_count+=1

for i in range(n):
    a,b=d[i]
    for j in range(n):
        if i in d[j]:
            dp[i]=dp[j]+1
    for _ in [a,b]:
        if _!=-1:
            dp[_]=dp[i]+1

print(max(dp),leaf_count)
```

代码运行截图 （至少包含有"Accepted"）


27638: 求二叉树的高度和叶子数目 Accepted

24729: 括号嵌套树

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

思路：

代码

```
#
def parse_tree(s):
    pre=[]
    post=[]
    for char in s:
        post.append(char)
        if char.isupper():
            pre.append(char)
        elif char==')':
            post.pop()
            a=''
            while post[-1]!='(':
                if post[-1]!=',':
                    a+=post.pop()
                else:
                    post.pop()
            post.pop()
            a=post.pop()+a
            post+=a[::-1]
    return pre,post

s=input()
pre,post=parse_tree(s)
```

```python
print(''.join(pre))
print(''.join(post))
```

代码运行截图 （至少包含有"Accepted"）

24729：括号嵌套树 Accepted

02775: 文件结构“图”

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

思路：

代码

```python
#
```

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

```python
#
class Node:
    def __init__(self,value):
        self.value=value
        self.left=None
        self.right=None

def build_expression_tree(postfix_expression):
    stack=[]
    for token in postfix_expression:
        node=Node(token)
        if token.isupper():
            node.right=stack.pop()
            node.left=stack.pop()
        stack.append(node)
    return stack.pop()

def level_order_traversal(root):
    if root is None:
        return []
    queue=[]
    result=[]
    queue.append(root)
    while queue:
        node=queue.pop(0)
        result.append(node.value)
        if node.left:
            queue.append(node.left)
        if node.right:
            queue.append(node.right)
    return result

for _ in range(int(input())):
    postfix_expression=input()
    expression_tree=build_expression_tree(postfix_expression)
    queue_expression=level_order_traversal(expression_tree)
    queue_expression.reverse()  # 颠倒顺序
    queue_expression=''.join(queue_expression)
    print(queue_expression)
```

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表达式树 Accepted

24750: 根据二叉树中后序序列建树

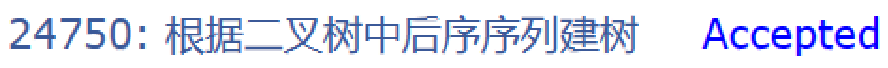
http://cs101.openjudge.cn/practice/24750/

思路：

代码

```
#
def build_preorder(inorder, postorder):
    if not inorder or not postorder:
        return ''

    root=postorder[-1]
    root_index=inorder.index(root)

    left_inorder=inorder[:root_index]
    right_inorder=inorder[root_index + 1:]

    left_postorder=postorder[:len(left_inorder)]
    right_postorder=postorder[len(left_inorder):-1]

    return root+build_preorder(left_inorder, left_postorder) +
build_preorder(right_inorder, right_postorder)

inorder=input()
postorder=input()
preorder=build_preorder(inorder, postorder)
print(preorder)
```

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   Accepted

22158: 根据二叉树前中序序列建树

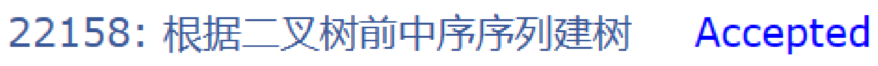
http://cs101.openjudge.cn/practice/22158/

思路：

代码

```
#
def build_tree(preorder,inorder):
    if not preorder or not inorder:
        return ''

    root=preorder[0]
    root_index=inorder.index(root)

    left_preorder=preorder[1:root_index+1]
    right_preorder=preorder[root_index+1:]

    left_inorder=inorder[:root_index]
    right_inorder=inorder[root_index+1:]

    left_tree=build_tree(left_preorder,left_inorder)
    right_tree=build_tree(right_preorder,right_inorder)

    return left_tree+right_tree+root

while True:
    try:
        preorder=input()
        inorder=input()
        postorder=build_tree(preorder, inorder)
        print(postorder)
    except EOFError:
        break
```

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 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。

本次作业难度较大，耗时较长，文件结构“图”一题尚未整理清楚