# Assignment #5: "树"算：概念、表示、解析、遍历

Updated 2124 GMT+8 March 17, 2024

2024 spring, Complied by 田济维

**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**

（python pycharm）

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

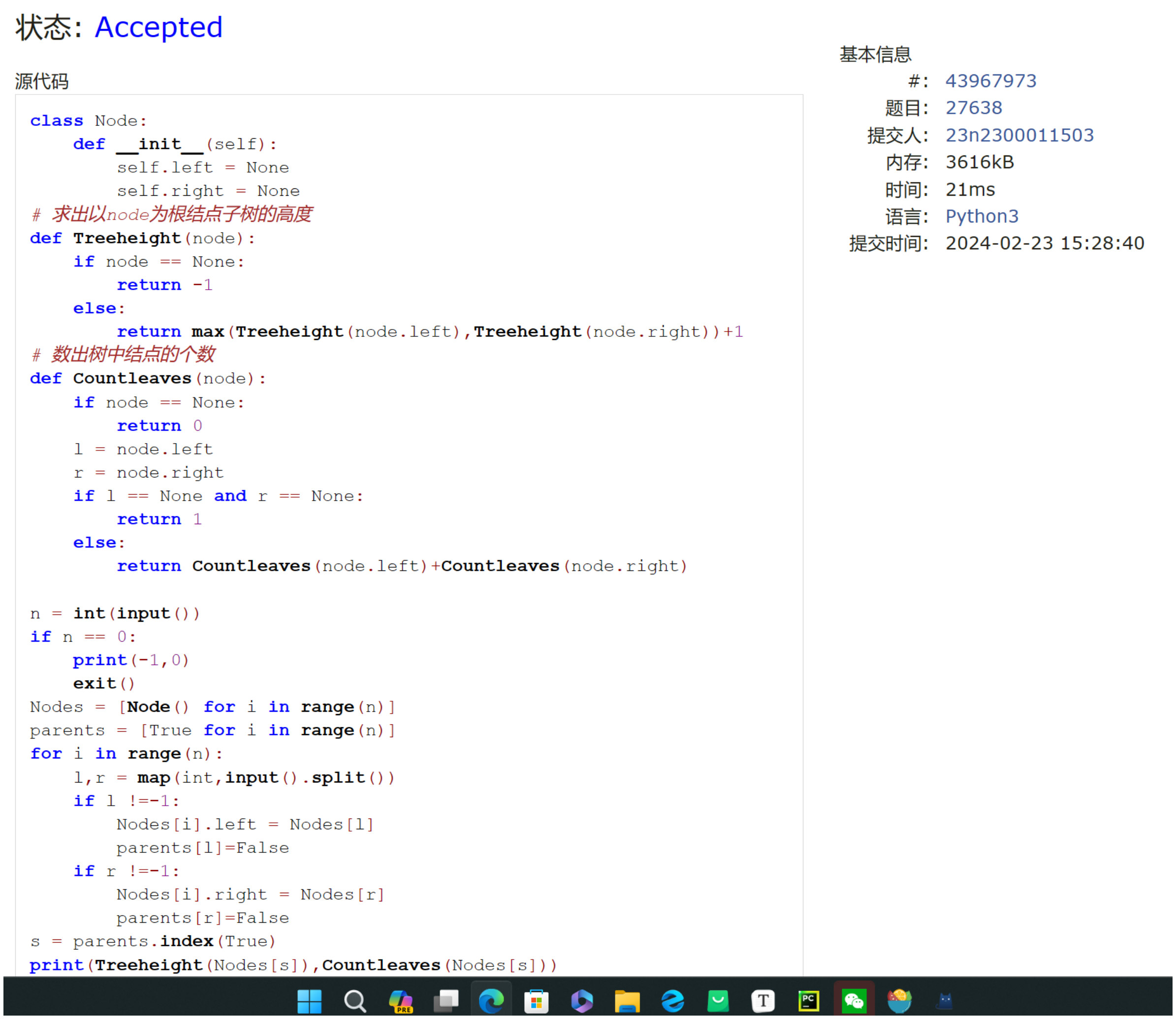
http://cs101.openjudge.cn/practice/27638/

思路：

代码

```python
#
class Node:
    def __init__(self):
        self.left = None
        self.right = None
# 求出以node为根结点子树的高度
def Treeheight(node):
    if node == None:
        return -1
    else:
        return max(Treeheight(node.left),Treeheight(node.right))+1
# 数出树中结点的个数
def Countleaves(node):
    if node == None:
        return 0
    l = node.left
    r = node.right
    if l == None and r == None:
        return 1
    else:
        return Countleaves(node.left)+Countleaves(node.right)

n = int(input())
if n == 0:
    print(-1,0)
    exit()
Nodes = [Node() for i in range(n)]
parents = [True for i in range(n)]
for i in range(n):
    l,r = map(int,input().split())
    if l !=-1:
        Nodes[i].left = Nodes[l]
        parents[l]=False
    if r !=-1:
        Nodes[i].right = Nodes[r]
        parents[r]=False
s = parents.index(True)
print(Treeheight(Nodes[s]),Countleaves(Nodes[s]))
```

代码运行截图 （至少包含有"Accepted"）


状态: Accepted

源代码

```python
class Node:
    def __init__(self):
        self.left = None
        self.right = None
# 求出以node为根结点子树的高度
def Treeheight(node):
    if node == None:
        return -1
    else:
        return max(Treeheight(node.left),Treeheight(node.right))+1
# 数出树中结点的个数
def Countleaves(node):
    if node == None:
        return 0
    l = node.left
    r = node.right
    if l == None and r == None:
        return 1
    else:
        return Countleaves(node.left)+Countleaves(node.right)

n = int(input())
if n == 0:
    print(-1,0)
    exit()
Nodes = [Node() for i in range(n)]
parents = [True for i in range(n)]
for i in range(n):
    l,r = map(int,input().split())
    if l !=-1:
        Nodes[i].left = Nodes[l]
        parents[l]=False
    if r !=-1:
        Nodes[i].right = Nodes[r]
        parents[r]=False
s = parents.index(True)
print(Treeheight(Nodes[s]),Countleaves(Nodes[s]))
```

基本信息

| | |
|---|---|
| #: | 43967973 |
| 题目: | 27638 |
| 提交人: | 23n2300011503 |
| 内存: | 3616kB |
| 时间: | 21ms |
| 语言: | Python3 |
| 提交时间: | 2024-02-23 15:28:40 |

## 24729: 括号嵌套树

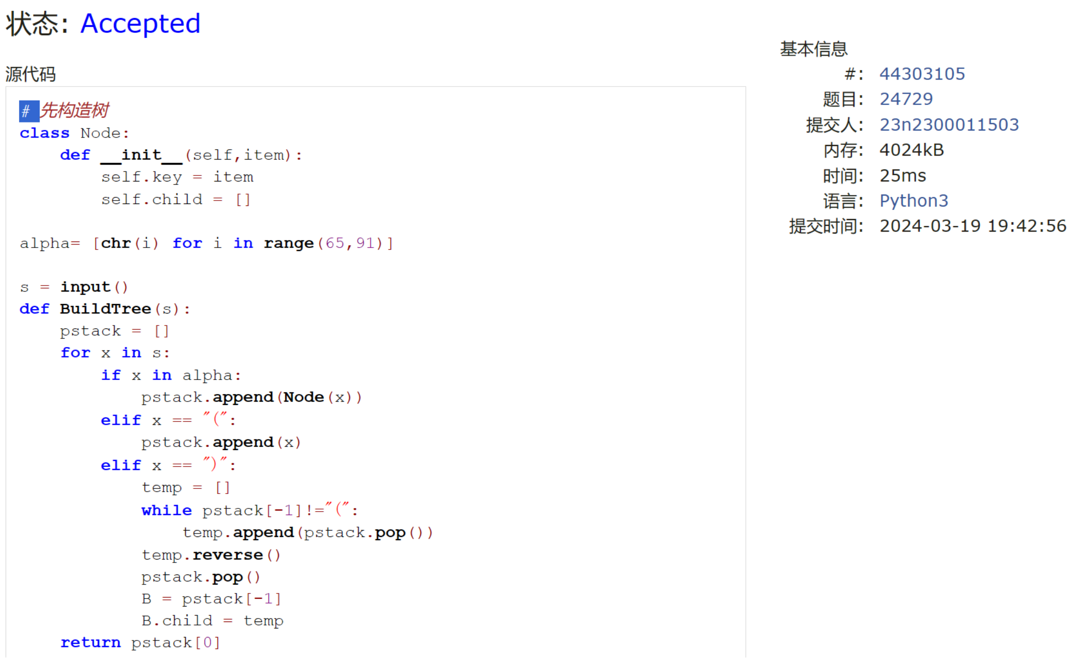
http://cs101.openjudge.cn/practice/24729/

思路:

代码

```python
#
# 先构造树
class Node:
    def __init__(self,item):
        self.key = item
        self.child = []

alpha= [chr(i) for i in range(65,91)]

s = input()
def BuildTree(s):
    pstack = []
    for x in s:
        if x in alpha:
            pstack.append(Node(x))
        elif x == "(":
            pstack.append(x)
        elif x == ")":
            temp = []
            while pstack[-1]!="(":
                temp.append(pstack.pop())
            temp.reverse()
            pstack.pop()
            B = pstack[-1]
            B.child = temp
    return pstack[0]
a = BuildTree(s)
def preorder(Tree):
    if Tree:
        print(Tree.key,end = "")
        for x in Tree.child:
            preorder(x)

def postorder(Tree):
    if Tree:
        for x in Tree.child:
            postorder(x)
        print(Tree.key,end = "")

preorder(a)
print("")
postorder(a)
```

代码运行截图 (至少包含有"Accepted")



状态: Accepted

源代码

```python
#先构造树
class Node:
    def __init__(self,item):
        self.key = item
        self.child = []

alpha= [chr(i) for i in range(65,91)]

s = input()
def BuildTree(s):
    pstack = []
    for x in s:
        if x in alpha:
            pstack.append(Node(x))
        elif x == "(":
            pstack.append(x)
        elif x == ")":
            temp = []
            while pstack[-1]!="(":
                temp.append(pstack.pop())
            temp.reverse()
            pstack.pop()
            B = pstack[-1]
            B.child = temp
    return pstack[0]
```

基本信息

| | |
|---|---|
| #: | 44303105 |
| 题目: | 24729 |
| 提交人: | 23n2300011503 |
| 内存: | 4024kB |
| 时间: | 25ms |
| 语言: | Python3 |
| 提交时间: | 2024-03-19 19:42:56 |

## 02775: 文件结构“图”

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

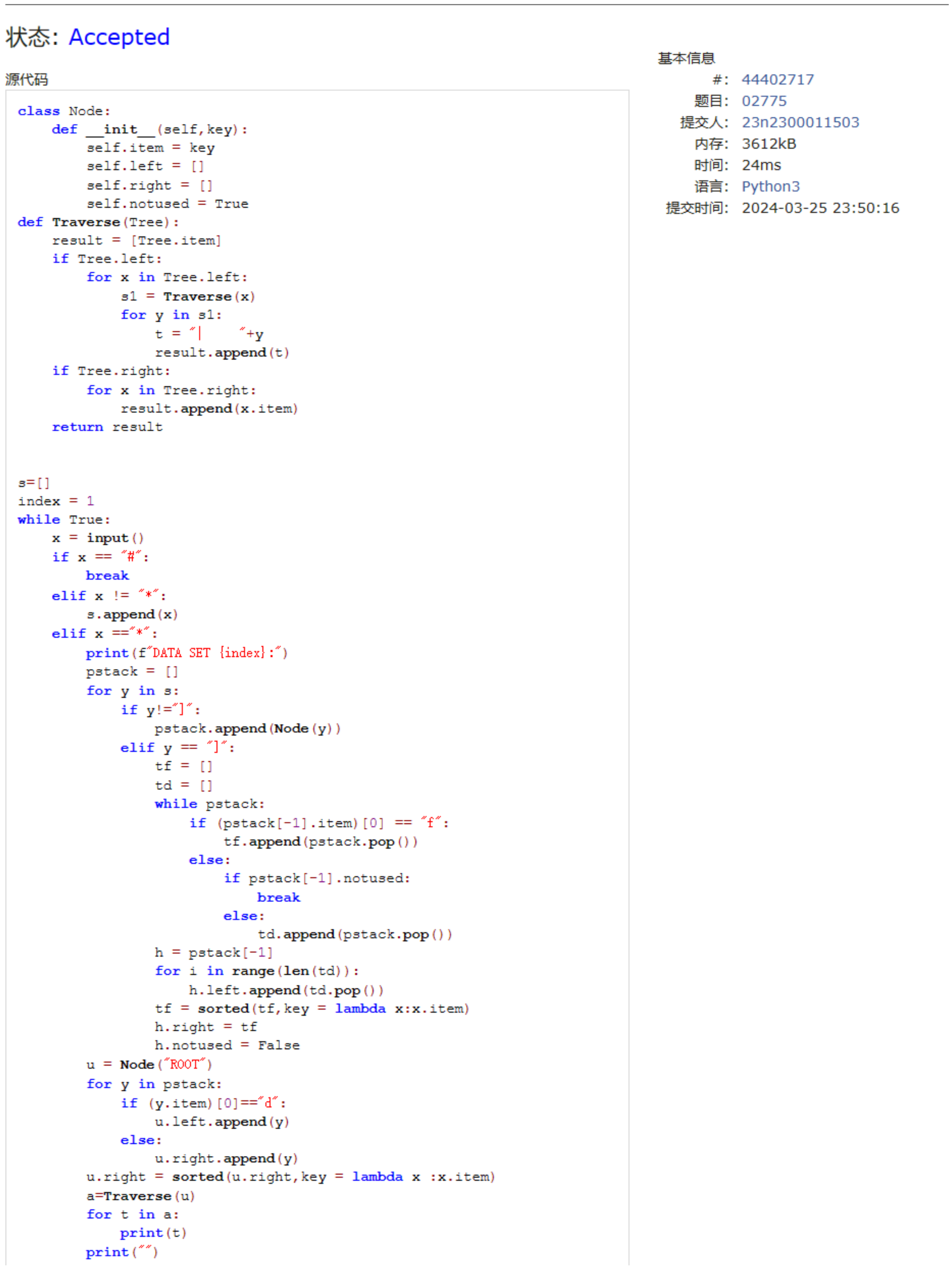
思路：

代码

```
#
class Node:
    def __init__(self,key):
        self.item = key
        self.left = []
        self.right = []
        self.notused = True
def Traverse(Tree):
    result = [Tree.item]
    if Tree.left:
        for x in Tree.left:
            s1 = Traverse(x)
            for y in s1:
                t = "|     "+y
                result.append(t)
    if Tree.right:
        for x in Tree.right:
            result.append(x.item)
    return result


s=[]
index = 1
while True:
    x = input()
    if x == "#":
        break
    elif x != "*":
        s.append(x)
    elif x =="*":
        print(f"DATA SET {index}:")
        pstack = []
        for y in s:
            if y!="]":
                pstack.append(Node(y))
            elif y == "]":
                tf = []
                td = []
                while pstack:
                    if (pstack[-1].item)[0] == "f":
                        tf.append(pstack.pop())
                    else:
                        if pstack[-1].notused:
                            break
```

```python
                        else:
                            td.append(pstack.pop())
                h = pstack[-1]
                for i in range(len(td)):
                    h.left.append(td.pop())
                tf = sorted(tf,key = lambda x:x.item)
                h.right = tf
                h.notused = False
        u = Node("ROOT")
        for y in pstack:
            if (y.item)[0]=="d":
                u.left.append(y)
            else:
                u.right.append(y)
        u.right = sorted(u.right,key = lambda x :x.item)
        a=Traverse(u)
        for t in a:
            print(t)
        print("")
        index+=1
        s.clear()
```

代码运行截图 （AC代码截图，至少包含有"Accepted"）


状态: Accepted

源代码

```
class Node:
    def __init__(self,key):
        self.item = key
        self.left = []
        self.right = []
        self.notused = True
def Traverse(Tree):
    result = [Tree.item]
    if Tree.left:
        for x in Tree.left:
            s1 = Traverse(x)
            for y in s1:
                t = "|     "+y
                result.append(t)
    if Tree.right:
        for x in Tree.right:
            result.append(x.item)
    return result


s=[]
index = 1
while True:
    x = input()
    if x == "#":
        break
    elif x != "*":
        s.append(x)
    elif x =="*":
        print(f"DATA SET {index}:")
        pstack = []
        for y in s:
            if y!="]":
                pstack.append(Node(y))
            elif y == "]":
                tf = []
                td = []
                while pstack:
                    if (pstack[-1].item)[0] == "f":
                        tf.append(pstack.pop())
                    else:
                        if pstack[-1].notused:
                            break
                        else:
                            td.append(pstack.pop())
                h = pstack[-1]
                for i in range(len(td)):
                    h.left.append(td.pop())
                tf = sorted(tf,key = lambda x:x.item)
                h.right = tf
                h.notused = False
        u = Node("ROOT")
        for y in pstack:
            if (y.item)[0]=="d":
                u.left.append(y)
            else:
                u.right.append(y)
        u.right = sorted(u.right,key = lambda x :x.item)
        a=Traverse(u)
        for t in a:
            print(t)
        print("")
```

基本信息

| | |
|---|---|
| #: | 44402717 |
| 题目: | 02775 |
| 提交人: | 23n2300011503 |
| 内存: | 3612kB |
| 时间: | 24ms |
| 语言: | Python3 |
| 提交时间: | 2024-03-25 23:50:16 |

## 25140: 根据后序表达式建立队列表达式

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

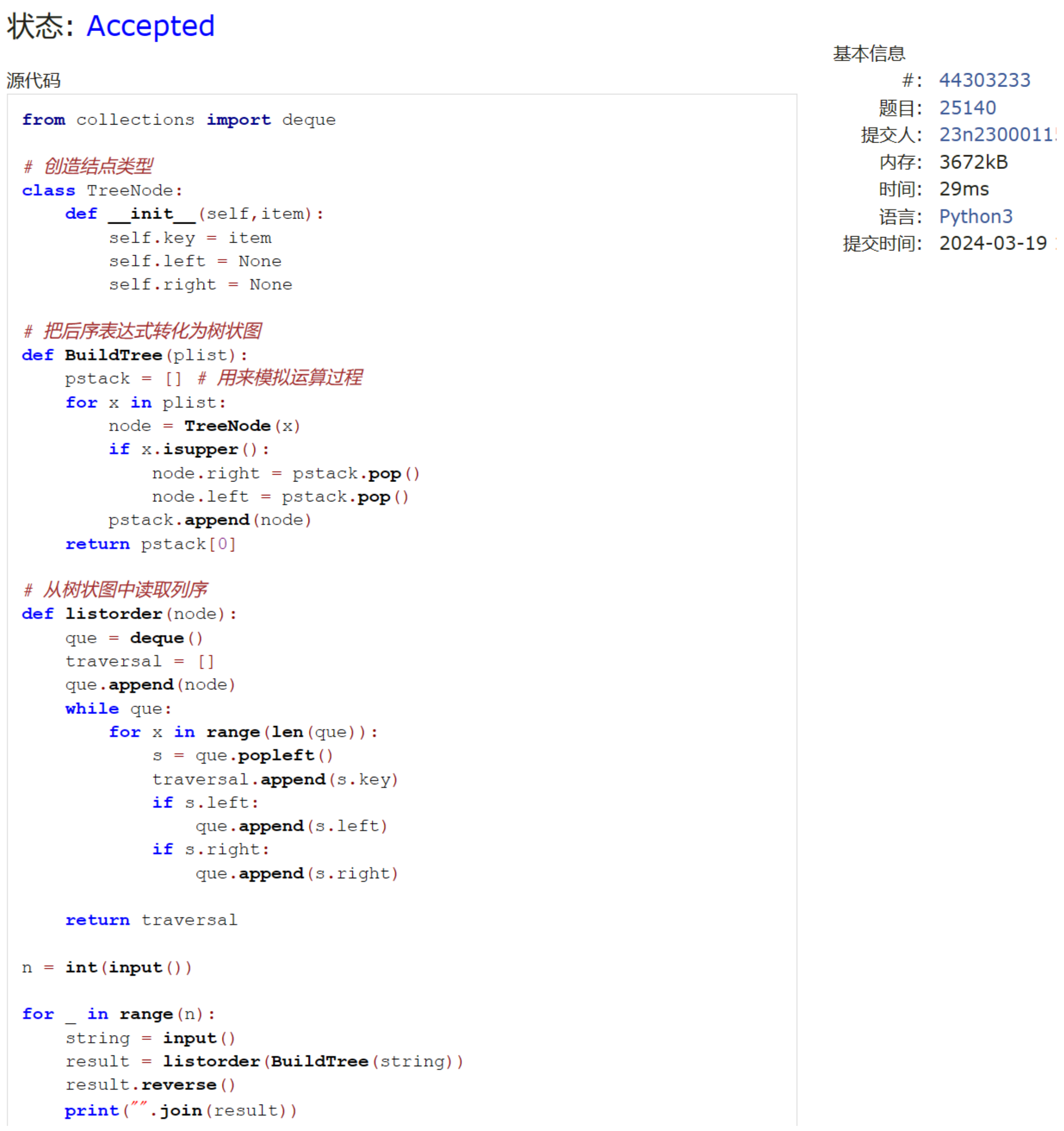
思路:

代码

```
# from collections import deque

```

```python
# 创造结点类型
class TreeNode:
    def __init__(self,item):
        self.key = item
        self.left = None
        self.right = None

# 把后序表达式转化为树状图
def BuildTree(plist):
    pstack = [] # 用来模拟运算过程
    for x in plist:
        node = TreeNode(x)
        if x.isupper():
            node.right = pstack.pop()
            node.left = pstack.pop()
        pstack.append(node)
    return pstack[0]

# 从树状图中读取列序
def listorder(node):
    que = deque()
    traversal = []
    que.append(node)
    while que:
        for x in range(len(que)):
            s = que.popleft()
            traversal.append(s.key)
            if s.left:
                que.append(s.left)
            if s.right:
                que.append(s.right)

    return traversal

n = int(input())

for _ in range(n):
    string = input()
    result = listorder(BuildTree(string))
    result.reverse()
    print("".join(result))



```

代码运行截图 （AC代码截图，至少包含有"Accepted"）



状态: Accepted

源代码

```python
from collections import deque

# 创造结点类型
class TreeNode:
    def __init__(self,item):
        self.key = item
        self.left = None
        self.right = None

# 把后序表达式转化为树状图
def BuildTree(plist):
    pstack = [] # 用来模拟运算过程
    for x in plist:
        node = TreeNode(x)
        if x.isupper():
            node.right = pstack.pop()
            node.left = pstack.pop()
        pstack.append(node)
    return pstack[0]

# 从树状图中读取列序
def listorder(node):
    que = deque()
    traversal = []
    que.append(node)
    while que:
        for x in range(len(que)):
            s = que.popleft()
            traversal.append(s.key)
            if s.left:
                que.append(s.left)
            if s.right:
                que.append(s.right)

    return traversal

n = int(input())

for _ in range(n):
    string = input()
    result = listorder(BuildTree(string))
    result.reverse()
    print("".join(result))
```

基本信息

| | |
|---|---|
| #: | 44303233 |
| 题目: | 25140 |
| 提交人: | 23n2300011 |
| 内存: | 3672kB |
| 时间: | 29ms |
| 语言: | Python3 |
| 提交时间: | 2024-03-19 |

## 24750: 根据二叉树中后序序列建树

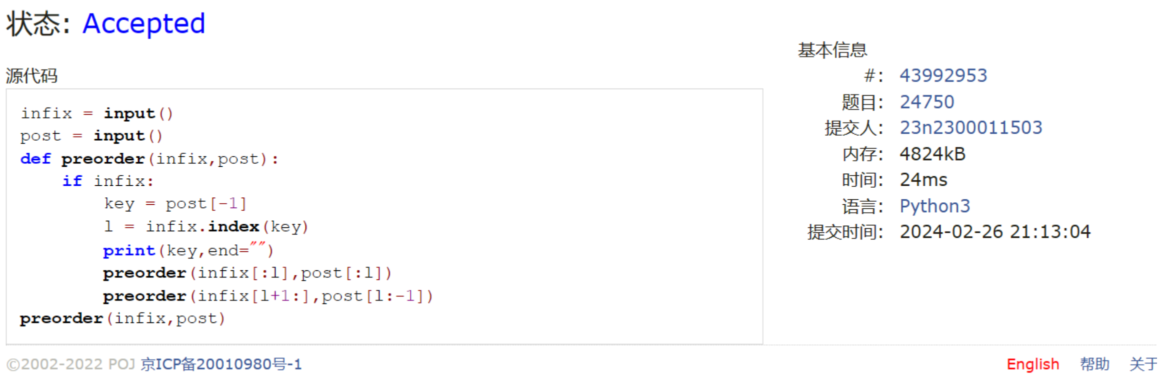
http://cs101.openjudge.cn/practice/24750/

思路:

代码

```python
#
infix = input()
post = input()
def preorder(infix,post):
    if infix:
        key = post[-1]
        l = infix.index(key)
        print(key,end="")
        preorder(infix[:l],post[:l])
        preorder(infix[l+1:],post[l:-1])
preorder(infix,post)

```

代码运行截图 （AC代码截图，至少包含有"Accepted"）



状态: Accepted

源代码

```python
infix = input()
post = input()
def preorder(infix,post):
    if infix:
        key = post[-1]
        l = infix.index(key)
        print(key,end="")
        preorder(infix[:l],post[:l])
        preorder(infix[l+1:],post[l:-1])
preorder(infix,post)
```

基本信息

| | |
|---|---|
| #: | 43992953 |
| 题目: | 24750 |
| 提交人: | 23n2300011503 |
| 内存: | 4824kB |
| 时间: | 24ms |
| 语言: | Python3 |
| 提交时间: | 2024-02-26 21:13:04 |

©2002-2022 POJ 京ICP备20010980号-1 English 帮助 关于

## 22158: 根据二叉树前中序序列建树

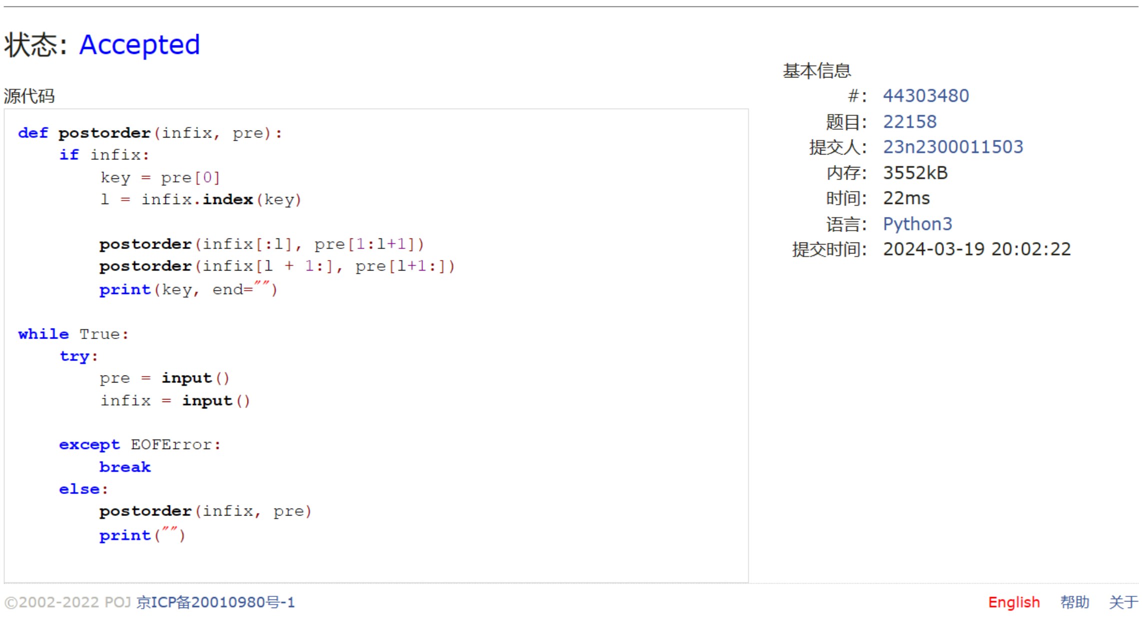
http://cs101.openjudge.cn/practice/22158/

思路:

代码

```python
def postorder(infix, pre):
    if infix:
        key = pre[0]
        l = infix.index(key)

        postorder(infix[:l], pre[1:l+1])
        postorder(infix[l + 1:], pre[l+1:])
        print(key, end="")

while True:
    try:
        pre = input()
        infix = input()

    except EOFError:
        break
    else:
        postorder(infix, pre)
        print("")


```

代码运行截图 （AC代码截图，至少包含有"Accepted"）



状态: Accepted

源代码

```
def postorder(infix, pre):
    if infix:
        key = pre[0]
        l = infix.index(key)

        postorder(infix[:l], pre[1:l+1])
        postorder(infix[l + 1:], pre[l+1:])
        print(key, end="")

while True:
    try:
        pre = input()
        infix = input()

    except EOFError:
        break
    else:
        postorder(infix, pre)
        print("")
```

基本信息

| | |
|---|---|
| #: | 44303480 |
| 题目: | 22158 |
| 提交人: | 23n2300011503 |
| 内存: | 3552kB |
| 时间: | 22ms |
| 语言: | Python3 |
| 提交时间: | 2024-03-19 20:02:22 |

©2002-2022 POJ 京ICP备20010980号-1 English 帮助 关于

## 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。

最让我难受并且有成就感的 是文件管理系统。