Assignment #5: "树"算：概念、表示、解析、遍历

Updated 2124 GMT+8 March 17, 2024

2024 spring, Complied by Xinjie Song, Phy

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 22H2

Python编程环境：PyCharm 2023.2 (Community Edition)

C/C++编程环境：g++ (x86_64-win32-seh-rev0, Built by MinGW-W64 project) 8.1.0

1. 题目

27638: 求二叉树的高度和叶子数目

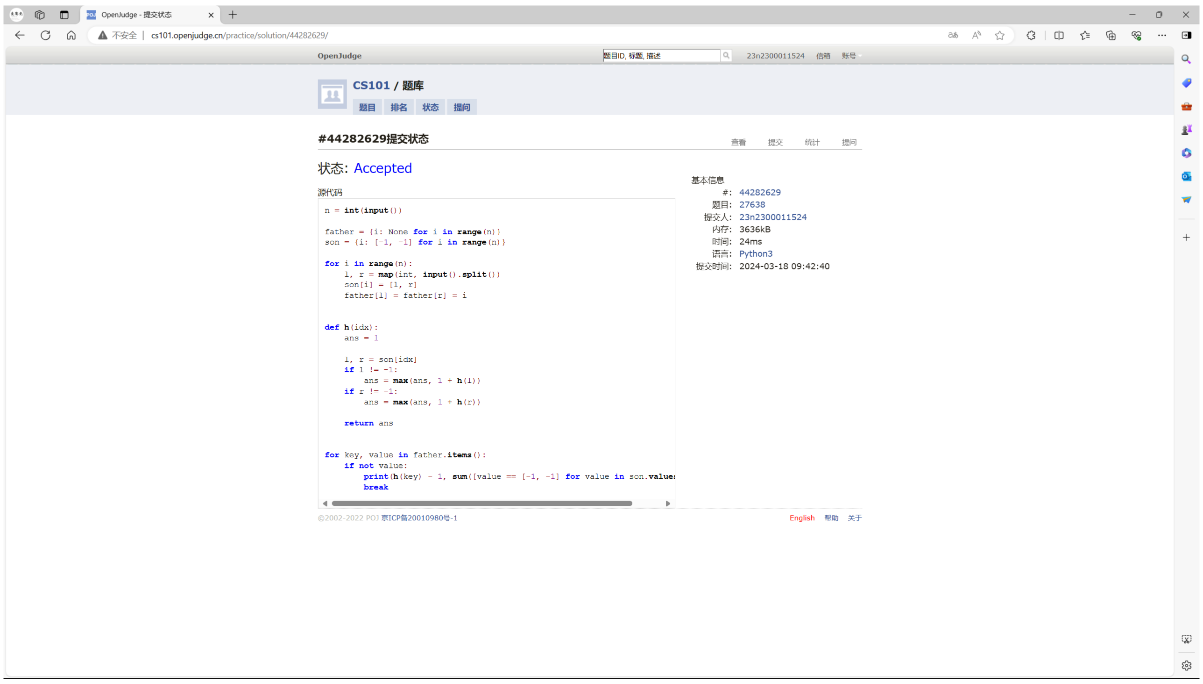
http://cs101.openjudge.cn/practice/27638/

思路：递归

代码

```python
n = int(input())

father = {i: None for i in range(n)}
son = {i: [-1, -1] for i in range(n)}

for i in range(n):
    l, r = map(int, input().split())
    son[i] = [l, r]
    father[l] = father[r] = i


def h(idx):
    ans = 1

    l, r = son[idx]
    if l != -1:
        ans = max(ans, 1 + h(l))
    if r != -1:
        ans = max(ans, 1 + h(r))

    return ans


for key, value in father.items():
    if not value:
        print(h(key) - 1, sum([value == [-1, -1] for value in son.values()]))
        break
```

代码运行截图


cs101.openjudge.cn/practice/solution/44282629/

OpenJudge   23n2300011524   信箱   账号

CS101 / 题库

#44282629提交状态

状态: Accepted

源代码

```python
n = int(input())

father = {i: None for i in range(n)}
son = {i: [-1, -1] for i in range(n)}

for i in range(n):
    l, r = map(int, input().split())
    son[i] = [l, r]
    father[l] = father[r] = i


def h(idx):
    ans = 1

    l, r = son[idx]
    if l != -1:
        ans = max(ans, 1 + h(l))
    if r != -1:
        ans = max(ans, 1 + h(r))

    return ans


for key, value in father.items():
    if not value:
        print(h(key) - 1, sum([value == [-1, -1] for value in son.value
        break
```

基本信息

#:	44282629
题目:	27638
提交人:	23n2300011524
内存:	3636kB
时间:	24ms
语言:	Python3
提交时间:	2024-03-18 09:42:40

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

24729: 括号嵌套树

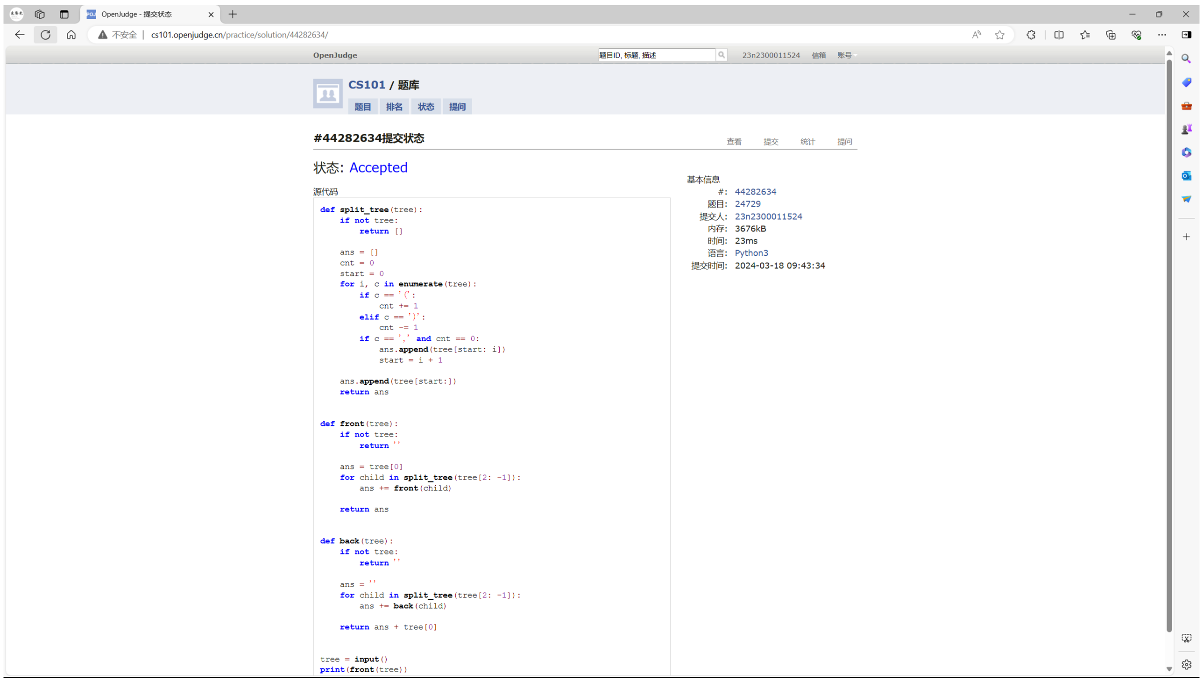
http://cs101.openjudge.cn/practice/24729/

思路：递归

代码

```python
def split_tree(tree):
    if not tree:
        return []

    ans = []
    cnt = 0
    start = 0
    for i, c in enumerate(tree):
        if c == '(':
            cnt += 1
        elif c == ')':
            cnt -= 1
        if c == ',' and cnt == 0:
            ans.append(tree[start: i])
            start = i + 1

    ans.append(tree[start:])
    return ans


def front(tree):
    if not tree:
        return ''

    ans = tree[0]
    for child in split_tree(tree[2: -1]):
        ans += front(child)

    return ans


def back(tree):
    if not tree:
        return ''

    ans = ''
    for child in split_tree(tree[2: -1]):
        ans += back(child)

    return ans + tree[0]


tree = input()
print(front(tree))
print(back(tree))
```

代码运行截图


#44282634提交状态

状态: Accepted

源代码

```python
def split_tree(tree):
    if not tree:
        return []

    ans = []
    cnt = 0
    start = 0
    for i, c in enumerate(tree):
        if c == '(':
            cnt += 1
        elif c == ')':
            cnt -= 1
        if c == ',' and cnt == 0:
            ans.append(tree[start: i])
            start = i + 1

    ans.append(tree[start:])
    return ans


def front(tree):
    if not tree:
        return ''

    ans = tree[0]
    for child in split_tree(tree[2: -1]):
        ans += front(child)

    return ans


def back(tree):
    if not tree:
        return ''

    ans = ''
    for child in split_tree(tree[2: -1]):
        ans += back(child)

    return ans + tree[0]


tree = input()
print(front(tree))
```

基本信息
#: 44282634
题目: 24729
提交人: 23n2300011524
内存: 3676kB
时间: 23ms
语言: Python3
提交时间: 2024-03-18 09:43:34

02775: 文件结构“图”

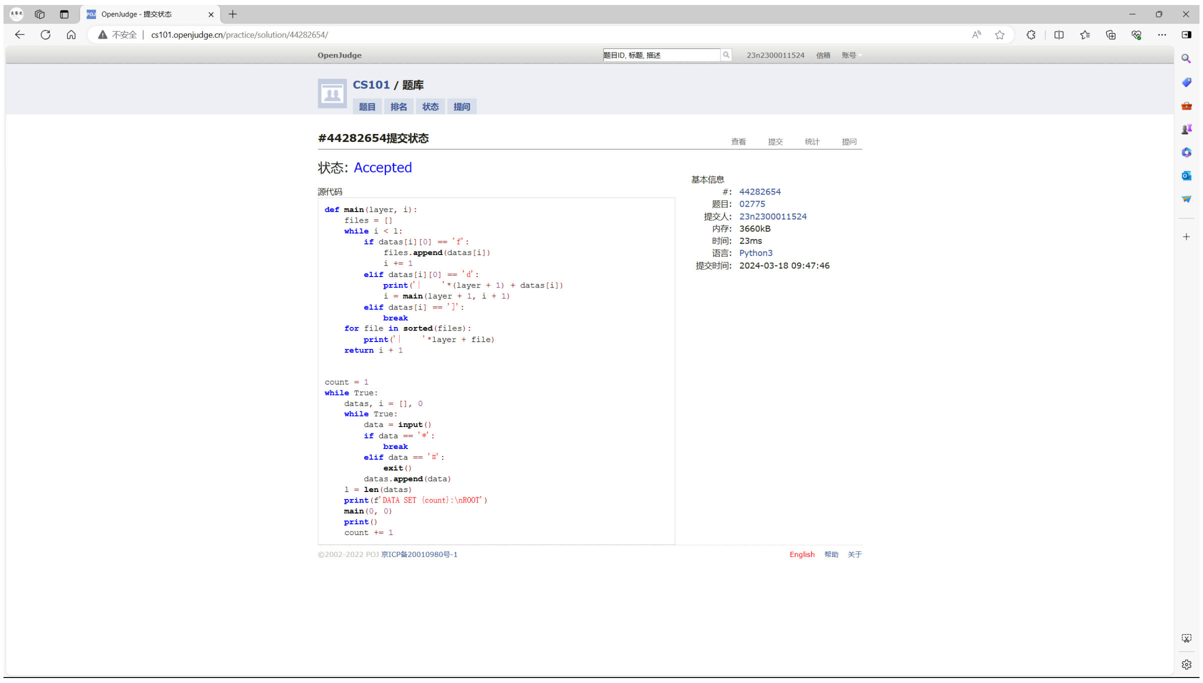
http://cs101.openjudge.cn/practice/02775/

思路：递归

代码

```python
def main(layer, i):
    files = []
    while i < l:
        if datas[i][0] == 'f':
            files.append(datas[i])
            i += 1
        elif datas[i][0] == 'd':
            print('|     '*(layer + 1) + datas[i])
            i = main(layer + 1, i + 1)
        elif datas[i] == ']':
            break
    for file in sorted(files):
        print('|     '*layer + file)
    return i + 1


count = 1
while True:
    datas, i = [], 0
    while True:
        data = input()
        if data == '*':
            break
        elif data == '#':
            exit()
        datas.append(data)
    l = len(datas)
    print(f'DATA SET {count}:\nROOT')
    main(0, 0)
    print()
    count += 1
```

代码运行截图



#44282654提交状态

状态: Accepted

源代码

```python
def main(layer, i):
    files = []
    while i < l:
        if datas[i][0] == 'f':
            files.append(datas[i])
            i += 1
        elif datas[i][0] == 'd':
            print('|     '*(layer + 1) + datas[i])
            i = main(layer + 1, i + 1)
        elif datas[i] == ']':
            break
    for file in sorted(files):
        print('|     '*layer + file)
    return i + 1


count = 1
while True:
    datas, i = [], 0
    while True:
        data = input()
        if data == '*':
            break
        elif data == '#':
            exit()
        datas.append(data)
    l = len(datas)
    print(f'DATA SET {count}:\nROOT')
    main(0, 0)
    print()
    count += 1
```

基本信息

#: 44282654
题目: 02775
提交人: 23n2300011524
内存: 3660kB
时间: 23ms
语言: Python3
提交时间: 2024-03-18 09:47:46

©2002-2022 POJ 京ICP备20010980号-1

25140: 根据后序表达式建立队列表达式

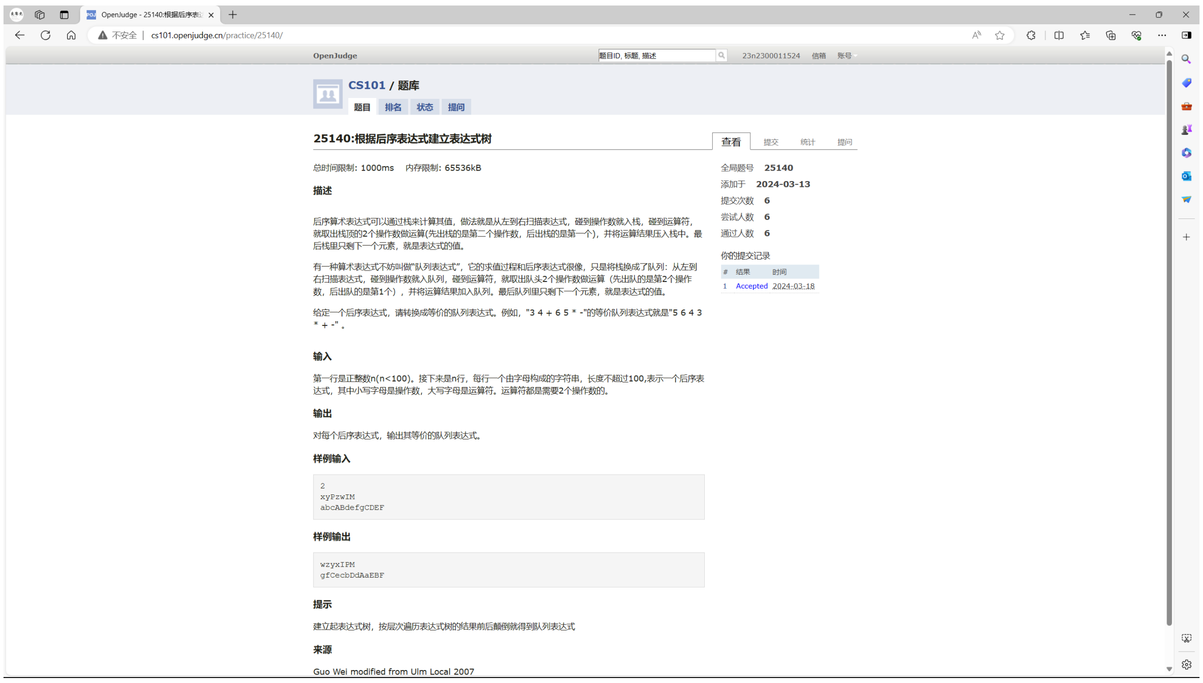
http://cs101.openjudge.cn/practice/25140/

思路：同提示

代码

```python
s = ''
idx = 0


class Tree:
    def __init__(self):
        self.value = None
        self.left_tree = None
        self.right_tree = None

    def layers(self):
        ans = []
        if self.left_tree:
            ans = self.left_tree.layers()
        if self.right_tree:
            t = self.right_tree.layers()
            for i in range(min(len(t), len(ans))):
                ans[i] += t[i]
            ans += t[len(ans):]

        return [self.value] + ans

    def initializing(self):
        global s, idx
        self.value = s[idx]
        if s[idx].islower():
            idx -= 1
            return

        idx -= 1
        self.right_tree = Tree()
        self.right_tree.initializing()
        self.left_tree = Tree()
        self.left_tree.initializing()

        return


for _ in range(int(input())):
    s = input()
    idx = len(s) - 1
    tree = Tree()
    tree.initializing()
    ans = ''.join(tree.layers())
    print(''.join([ans[i] for i in range(len(ans) - 1, -1, -1)]))
```

代码运行截图



25140:根据后序表达式建立表达式树

总时间限制: 1000ms　内存限制: 65536kB

全局题号 25140
添加于 2024-03-13
提交次数 6
尝试人数 6
通过人数 6

你的提交记录

#	结果	时间
1	Accepted	2024-03-18

描述

后序算术表达式可以通过栈来计算其值，做法就是从左到右扫描表达式，碰到操作数就入栈，碰到运算符，就取出栈顶的2个操作数做运算(先出栈的是第二个操作数，后出栈的是第一个)，并将运算结果压入栈中。最后栈里只剩下一个元素，就是表达式的值。

有一种算术表达式不妨叫做“队列表达式”，它的求值过程和后序表达式很像，只是将栈换成了队列：从左到右扫描表达式，碰到操作数就入队列，碰到运算符，就取出队头2个操作数做运算（先出队的是第2个操作数，后出队的是第1个），并将运算结果加入队列。最后队列里只剩下一个元素，就是表达式的值。

给定一个后序表达式，请转换成等价的队列表达式。例如，"3 4 + 6 5 * -"的等价队列表达式就是"5 6 4 3 * + -" 。

输入

第一行是正整数n(n<100)。接下来是n行，每行一个由字母构成的字符串，长度不超过100,表示一个后序表达式，其中小写字母是操作数，大写字母是运算符。运算符都是需要2个操作数的。

输出

对每个后序表达式，输出其等价的队列表达式。

样例输入

```
2
xyPzwIM
abcABdefgCDEF
```

样例输出

```
wzyxIPM
gfCecbDdAaEBF
```

提示

建立起表达式树，按层次遍历表达式树的结果前后颠倒就得到队列表达式

来源

Guo Wei modified from Ulm Local 2007

24750: 根据二叉树中后序序列建树

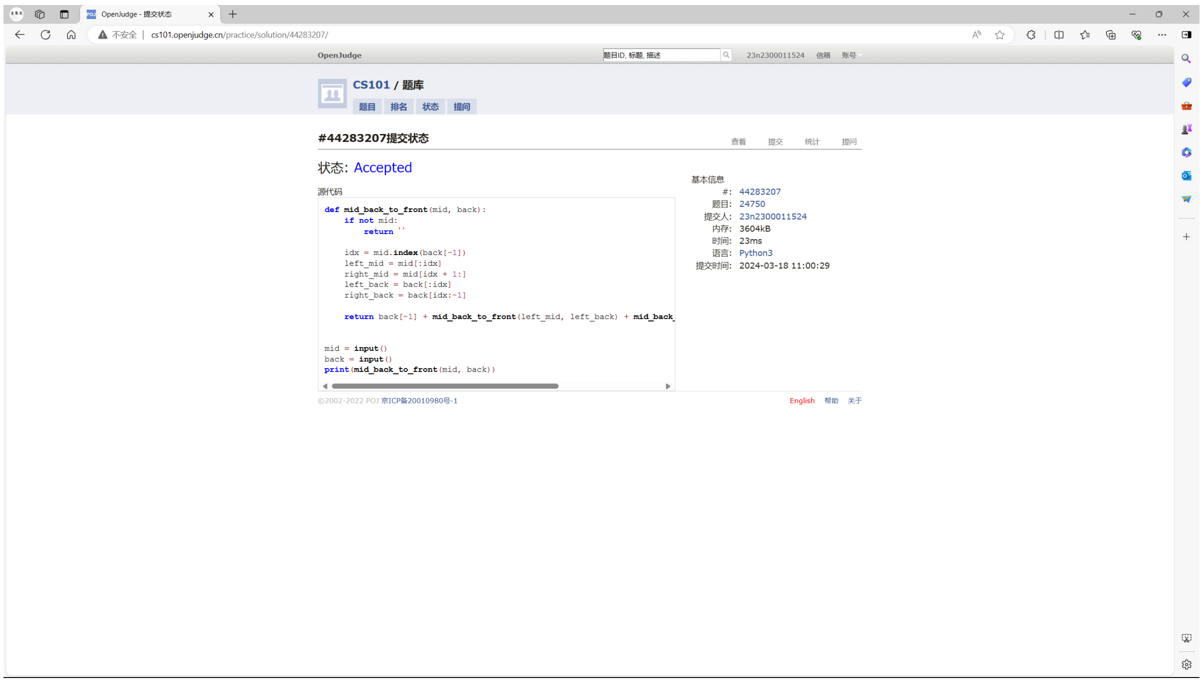
http://cs101.openjudge.cn/practice/24750/

思路：根据后续表达式最后一位确定根节点，继而确定左右子树，递归即可

代码

```python
def mid_back_to_front(mid, back):
    if not mid:
        return ''

    idx = mid.index(back[-1])
    left_mid = mid[:idx]
    right_mid = mid[idx + 1:]
    left_back = back[:idx]
    right_back = back[idx:-1]

    return back[-1] + mid_back_to_front(left_mid, left_back) +
mid_back_to_front(right_mid, right_back)


mid = input()
back = input()
print(mid_back_to_front(mid, back))
```

代码运行截图



#44283207提交状态

状态: Accepted

源代码

```
def mid_back_to_front(mid, back):
    if not mid:
        return ''

    idx = mid.index(back[-1])
    left_mid = mid[:idx]
    right_mid = mid[idx + 1:]
    left_back = back[:idx]
    right_back = back[idx:-1]

    return back[-1] + mid_back_to_front(left_mid, left_back) + mid_back_


mid = input()
back = input()
print(mid_back_to_front(mid, back))
```

基本信息

#:	44283207
题目:	24750
提交人:	23n2300011524
内存:	3604kB
时间:	23ms
语言:	Python3
提交时间:	2024-03-18 11:00:29

©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

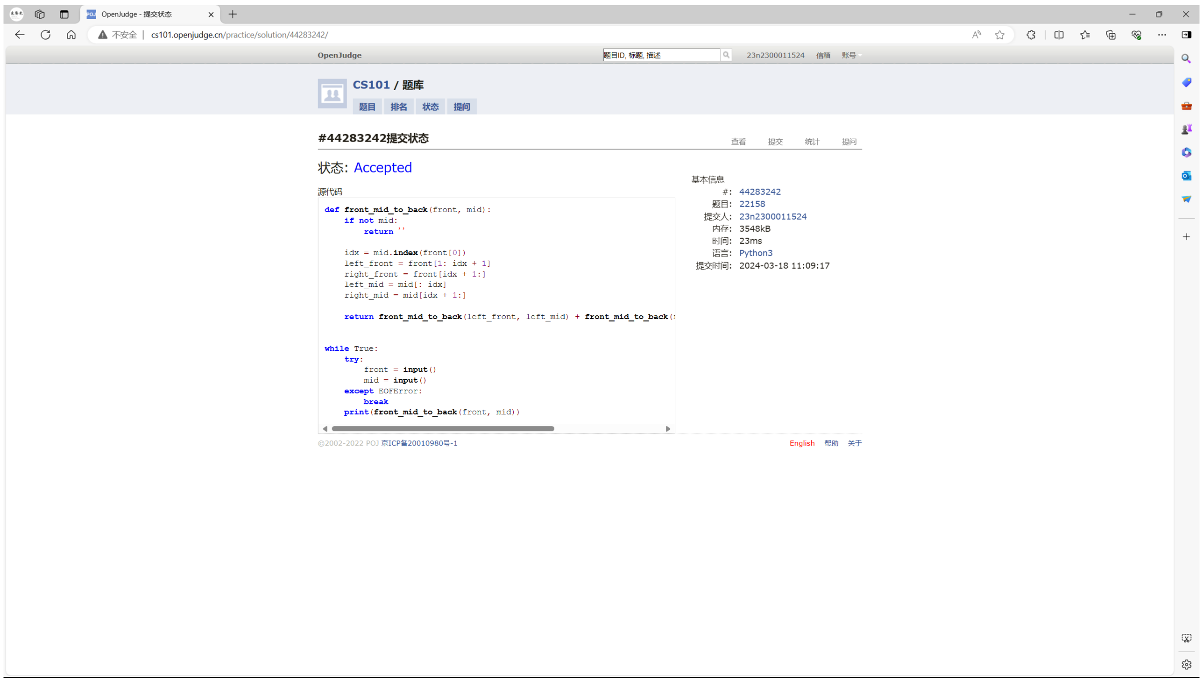
http://cs101.openjudge.cn/practice/22158/

思路：同上

代码

```python
def front_mid_to_back(front, mid):
    if not mid:
        return ''

    idx = mid.index(front[0])
    left_front = front[1: idx + 1]
    right_front = front[idx + 1:]
    left_mid = mid[: idx]
    right_mid = mid[idx + 1:]

    return front_mid_to_back(left_front, left_mid) + 
front_mid_to_back(right_front, right_mid) + front[0]


while True:
    try:
        front = input()
        mid = input()
    except EOFError:
        break
    print(front_mid_to_back(front, mid))
```

代码运行截图



#44283242提交状态

状态: Accepted

源代码

```python
def front_mid_to_back(front, mid):
    if not mid:
        return ''

    idx = mid.index(front[0])
    left_front = front[1: idx + 1]
    right_front = front[idx + 1:]
    left_mid = mid[: idx]
    right_mid = mid[idx + 1:]

    return front_mid_to_back(left_front, left_mid) + front_mid_to_back(


while True:
    try:
        front = input()
        mid = input()
    except EOFError:
        break
    print(front_mid_to_back(front, mid))
```

基本信息

#:	44283242
题目:	22158
提交人:	23n2300011524
内存:	3548kB
时间:	23ms
语言:	Python3
提交时间:	2024-03-18 11:09:17

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

2. 学习总结和收获

递归永不为奴

前序后序首尾拆分中序

中序拆完再拆前序后序