Assignment #4: 排序、栈、队列和树

Updated 0005 GMT+8 March 11, 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

05902: 双端队列

http://cs101.openjudge.cn/practice/05902/

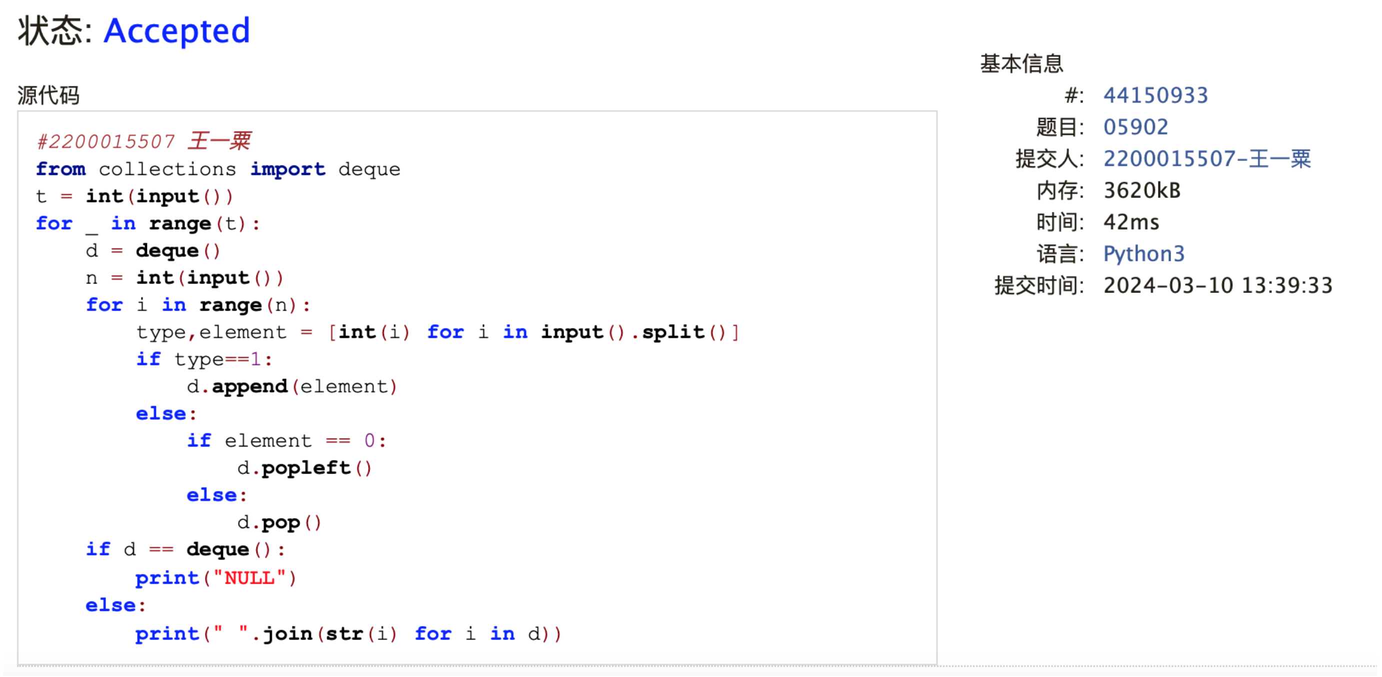
思路：题目很清晰，采用deque包很舒服。

耗时：15min

代码

```python
#2200015507 王一粟
from collections import deque
t = int(input())
for _ in range(t):
    d = deque()
    n = int(input())
    for i in range(n):
        type,element = [int(i) for i in input().split()]
        if type==1:
            d.append(element)
        else:
            if element == 0:
                d.popleft()
            else:
                d.pop()
    if d == deque():
        print("NULL")
    else:
        print(" ".join(str(i) for i in d))
```

代码运行截图 ==（至少包含有"Accepted"）==



状态: Accepted

源代码

```python
#2200015507 王一粟
from collections import deque
t = int(input())
for _ in range(t):
    d = deque()
    n = int(input())
    for i in range(n):
        type,element = [int(i) for i in input().split()]
        if type==1:
            d.append(element)
        else:
            if element == 0:
                d.popleft()
            else:
                d.pop()
    if d == deque():
        print("NULL")
    else:
        print(" ".join(str(i) for i in d))
```

基本信息

#:	44150933
题目:	05902
提交人:	2200015507-王一粟
内存:	3620kB
时间:	42ms
语言:	Python3
提交时间:	2024-03-10 13:39:33

02694: 波兰表达式

http://cs101.openjudge.cn/practice/02694/

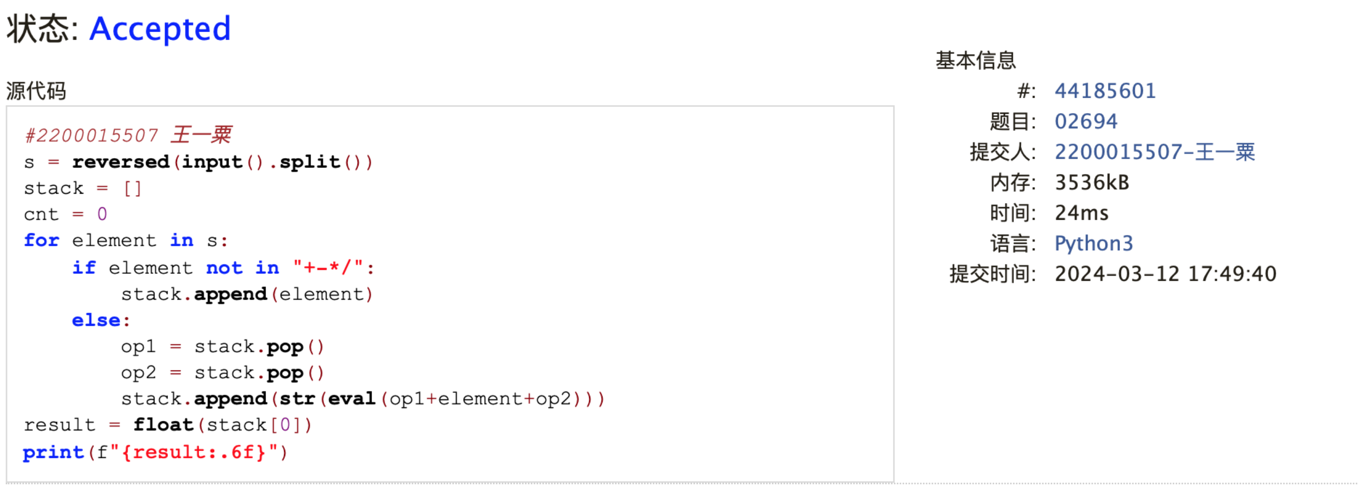
思路：用栈做，碰到操作符后就弹出2个操作数做运算。题解很厉害，我目前想不到。

耗时：20min

代码

```python
#2200015507 王一粟
s = reversed(input().split())
stack = []
cnt = 0
for element in s:
    if element not in "+-*/":
        stack.append(element)
    else:
        op1 = stack.pop()
        op2 = stack.pop()
        stack.append(str(eval(op1+element+op2)))
result = float(stack[0])
print(f"{result:.6f}")
```

代码运行截图 ==（至少包含有"Accepted"）==


状态: Accepted

源代码

```python
#2200015507 王一粟
s = reversed(input().split())
stack = []
cnt = 0
for element in s:
    if element not in "+-*/":
        stack.append(element)
    else:
        op1 = stack.pop()
        op2 = stack.pop()
        stack.append(str(eval(op1+element+op2)))
result = float(stack[0])
print(f"{result:.6f}")
```

基本信息

#:	44185601
题目:	02694
提交人:	2200015507-王一粟
内存:	3536kB
时间:	24ms
语言:	Python3
提交时间:	2024-03-12 17:49:40

24591: 中序表达式转后序表达式

http://cs101.openjudge.cn/practice/24591/

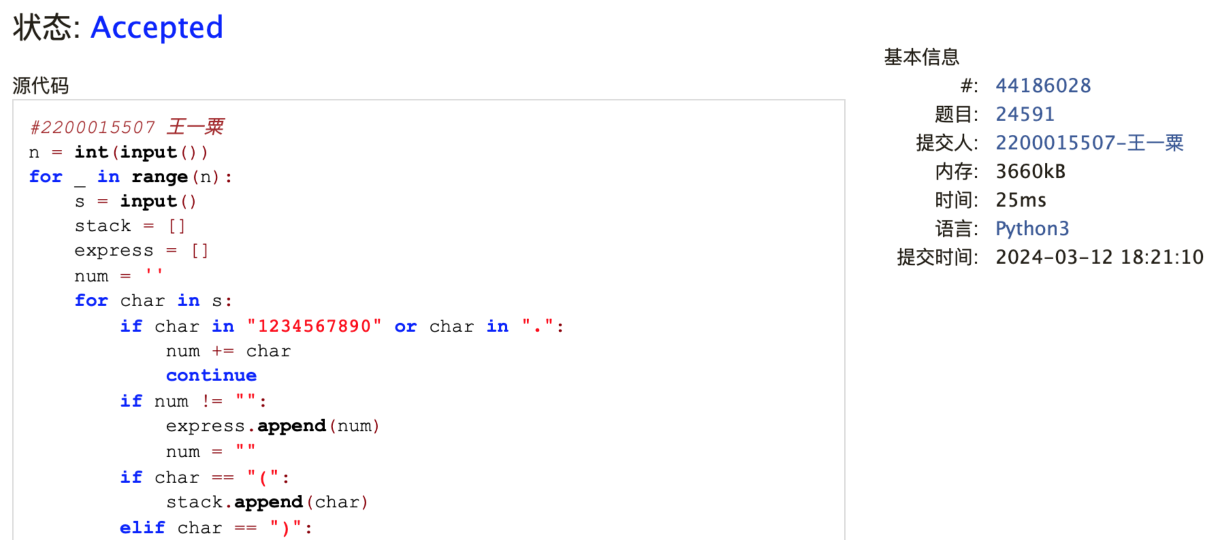
思路：经典题目。强调几件事情：while stack and stack[-1] == xxx的写法是没问题的，就算stack为空在while stack就会跳出，但颠倒顺序是有问题的；num的构建要注意做continue循环，当num不为空且当前char不为num是跳出；最后的expression在结束遍历后要先加num然后栈依次弹出。

耗时：30min

代码

```python
#2200015507 王一粟
n = int(input())
for _ in range(n):
    s = input()
    stack = []
    express = []
    num = ''
    for char in s:
        if char in "1234567890" or char in ".":
            num += char
            continue
        if num != "":
            express.append(num)
            num = ""
        if char == "(":
            stack.append(char)
        elif char == ")":
            while True:
                a = stack.pop()
                if a == "(":
                    break
                express.append(a)
        elif char in "+-":
            while stack:
                if stack[-1] in "+-*/":
                    express.append(stack.pop())
                else:
                    break
            stack.append(char)
        else:
            while stack:
                if stack[-1] in "*/":
                    express.append(stack.pop())
                else:
                    break
            stack.append(char)
    if num:
        express.append(num)
    while stack:
        express.append(stack.pop())
    print(" ".join(i for i in express))
```

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
n = int(input())
for _ in range(n):
    s = input()
    stack = []
    express = []
    num = ''
    for char in s:
        if char in "1234567890" or char in ".":
            num += char
            continue
        if num != "":
            express.append(num)
            num = ""
        if char == "(":
            stack.append(char)
        elif char == ")":
```

基本信息

#:	44186028
题目:	24591
提交人:	2200015507-王一粟
内存:	3660kB
时间:	25ms
语言:	Python3
提交时间:	2024-03-12 18:21:10

22068: 合法出栈序列

http://cs101.openjudge.cn/practice/22068/

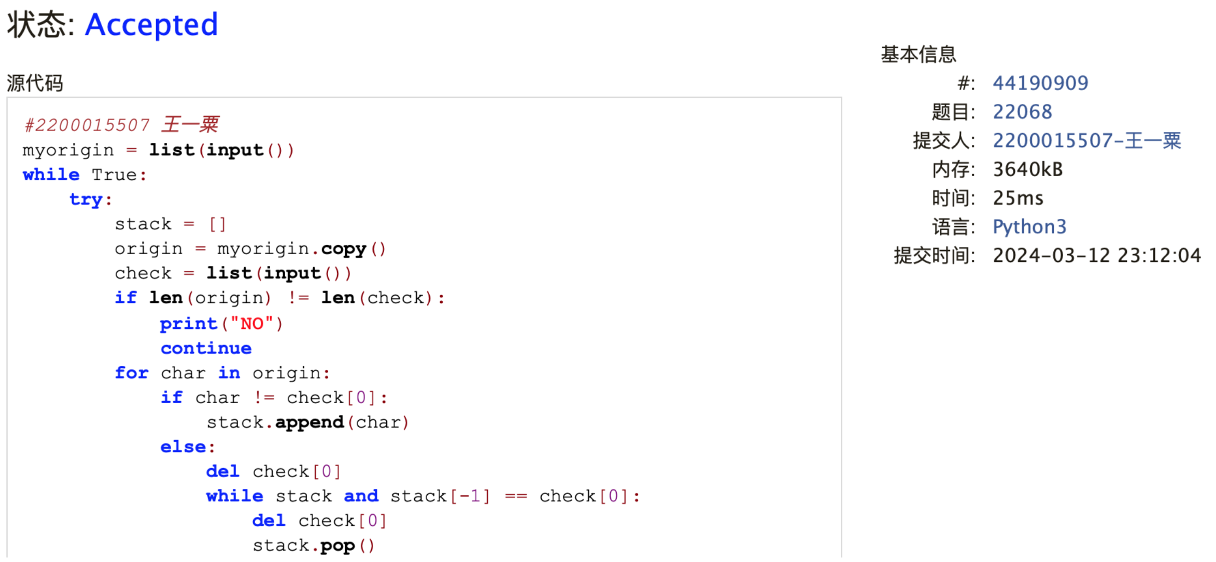
思路：根据可能的输出结果做校对。在排除了不等长度之后，若原串中取出的恰等于余下配对串种第一个，匹配完成，并判断栈顶是否可以继续配对；否则弹入栈中。

耗时：20min

代码

```python
#2200015507 王一粟
myorigin = list(input())
while True:
    try:
        stack = []
        origin = myorigin.copy()
        check = list(input())
        if len(origin) != len(check):
            print("NO")
            continue
        for char in origin:
            if char != check[0]:
                stack.append(char)
            else:
                del check[0]
                while stack and stack[-1] == check[0]:
                    del check[0]
                    stack.pop()
        if len(stack) == 0:
            print("YES")
        else:
            print("NO")
    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

```python
#2200015507 王一粟
myorigin = list(input())
while True:
    try:
        stack = []
        origin = myorigin.copy()
        check = list(input())
        if len(origin) != len(check):
            print("NO")
            continue
        for char in origin:
            if char != check[0]:
                stack.append(char)
            else:
                del check[0]
                while stack and stack[-1] == check[0]:
                    del check[0]
                    stack.pop()
```

基本信息

#:	44190909
题目:	22068
提交人:	2200015507-王一粟
内存:	3640kB
时间:	25ms
语言:	Python3
提交时间:	2024-03-12 23:12:04

06646: 二叉树的深度

http://cs101.openjudge.cn/practice/06646/

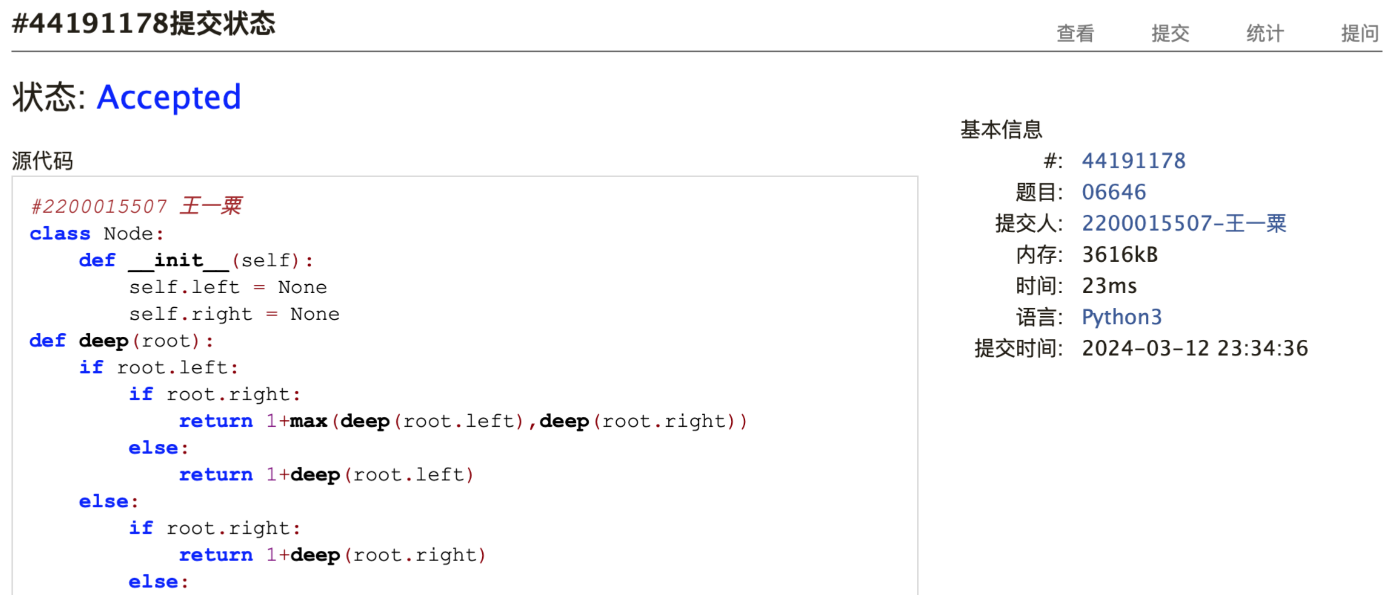
思路：先构建二叉树，用列表安装1-n个结点。然后做二叉树解析。最后通过max递归求出二叉树的最大层级。

耗时：20min

代码

```python
#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
n = int(input())
mylist = [Node() for i in range(n)]
for i in range(1,n+1):
    leftnum,rightnum = [int(k) for k in input().split()]
    if leftnum != -1:
        mylist[i - 1].left = mylist[leftnum - 1]
    if rightnum != -1:
        mylist[i - 1].right = mylist[rightnum - 1]
print(deep(mylist[0]))
```

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==


#44191178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):
        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:
```

基本信息

#:	44191178
题目:	06646
提交人:	2200015507-王一粟
内存:	3616kB
时间:	23ms
语言:	Python3
提交时间:	2024-03-12 23:34:36

02299: Ultra-QuickSort

http://cs101.openjudge.cn/practice/02299/

思路：第一次做接近于直接穷举，不断简化后也发现没有办法达到时间限制。基于hint采用了merge算法，在计算逆序数时候的代码还是总出错WA（主要在于发现左侧一个元素大于右侧后到底应该怎么计算逆序数，最后发现没必要分类讨论，直接加上左侧总列表长度减去当前索引数即可，规避少算或者重复运算），最后虽然答案正确，但依旧超时。最后发现是因为代码中使用了index函数复杂度太高，循环遍历专用enumerate后复杂度过关。不过教训是还是直接基于merge排序标准代码去修改为佳。

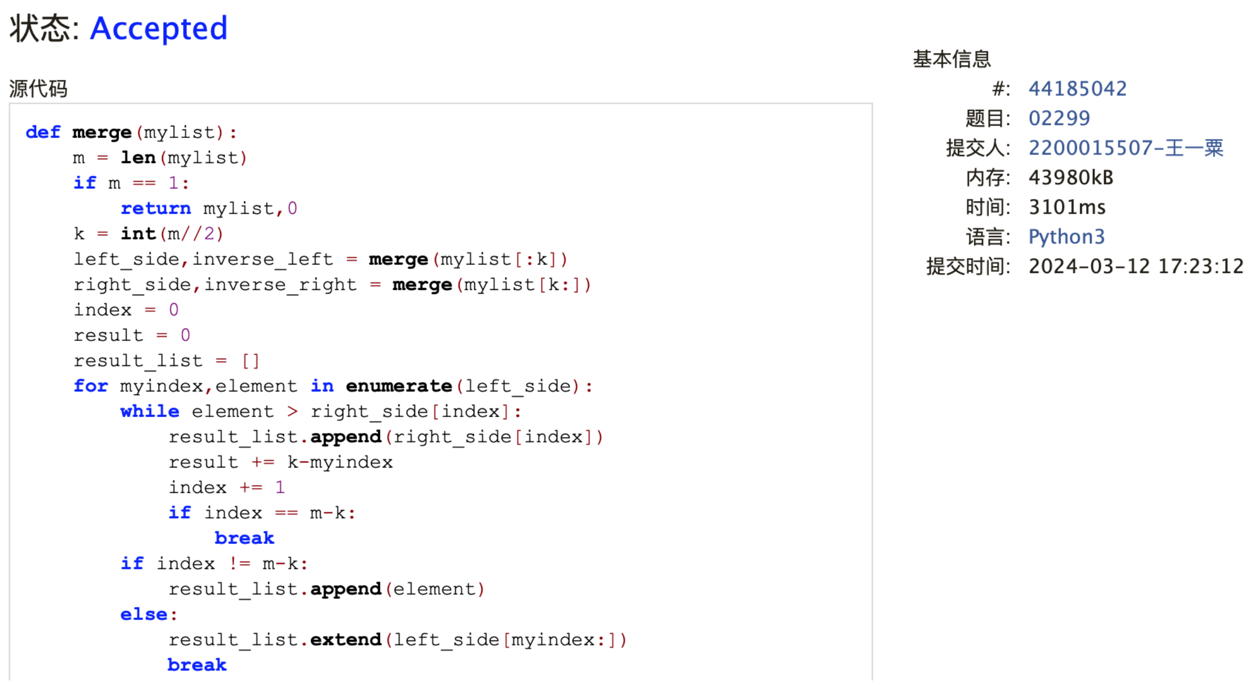
耗时：2.5h

代码

```python
# 2200015507 王一粟
def merge(mylist):
    m = len(mylist)
    if m == 1:
        return mylist,0
    k = int(m//2)
    left_side,inverse_left = merge(mylist[:k])
    right_side,inverse_right = merge(mylist[k:])
    index = 0
    result = 0
    result_list = []
    for myindex,element in enumerate(left_side):
        while element > right_side[index]:
            result_list.append(right_side[index])
            result += k-myindex
            index += 1
            if index == m-k:
                break
        if index != m-k:
            result_list.append(element)
        else:
            result_list.extend(left_side[myindex:])
            break
    if index != m-k:
        result_list.extend(right_side[index:])
    return result_list,inverse_left+inverse_right+result

while True:
    n = int(input())
    if n == 0:
        break
    mylist = []
    for _ in range(n):
        mylist.append(int(input()))
    result_list,inverse_num = merge(mylist)
    print(inverse_num)
```

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==



状态: Accepted

源代码

```
def merge(mylist):
    m = len(mylist)
    if m == 1:
        return mylist,0
    k = int(m//2)
    left_side,inverse_left = merge(mylist[:k])
    right_side,inverse_right = merge(mylist[k:])
    index = 0
    result = 0
    result_list = []
    for myindex,element in enumerate(left_side):
        while element > right_side[index]:
            result_list.append(right_side[index])
            result += k-myindex
            index += 1
            if index == m-k:
                break
        if index != m-k:
            result_list.append(element)
        else:
            result_list.extend(left_side[myindex:])
            break
```

基本信息

#:	44185042
题目:	02299
提交人:	2200015507-王一粟
内存:	43980kB
时间:	3101ms
语言:	Python3
提交时间:	2024-03-12 17:23:12

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。==

快排题目自己遇到了不小的麻烦，不过最后找到自己的错误和错误的原因还是很有成就感的，也感觉自己现在确实和厉害的同学们有一些差距。

除了快排之外其他题在每日选做都做过，这次重新做也发现了自己还是会遗漏和错误的细节，都在思路里面进行了标注。

基本恢复了去年学计概C的水平。但是优化算法和计概B的大家有差距，我补。

目前就是跟紧每日选做和老师课上的知识，本周争取补完老师递归的讲义。