Assignment #4: 排序、栈、队列和树

Updated 0005 GMT+8 March 11, 2024

2024 spring, Complied by ==苏王捷 工学院==

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：Windows11

Python编程环境：Spyder IDE 5.5.0

1. 题目

05902: 双端队列

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

思路：用指针模拟从前出队

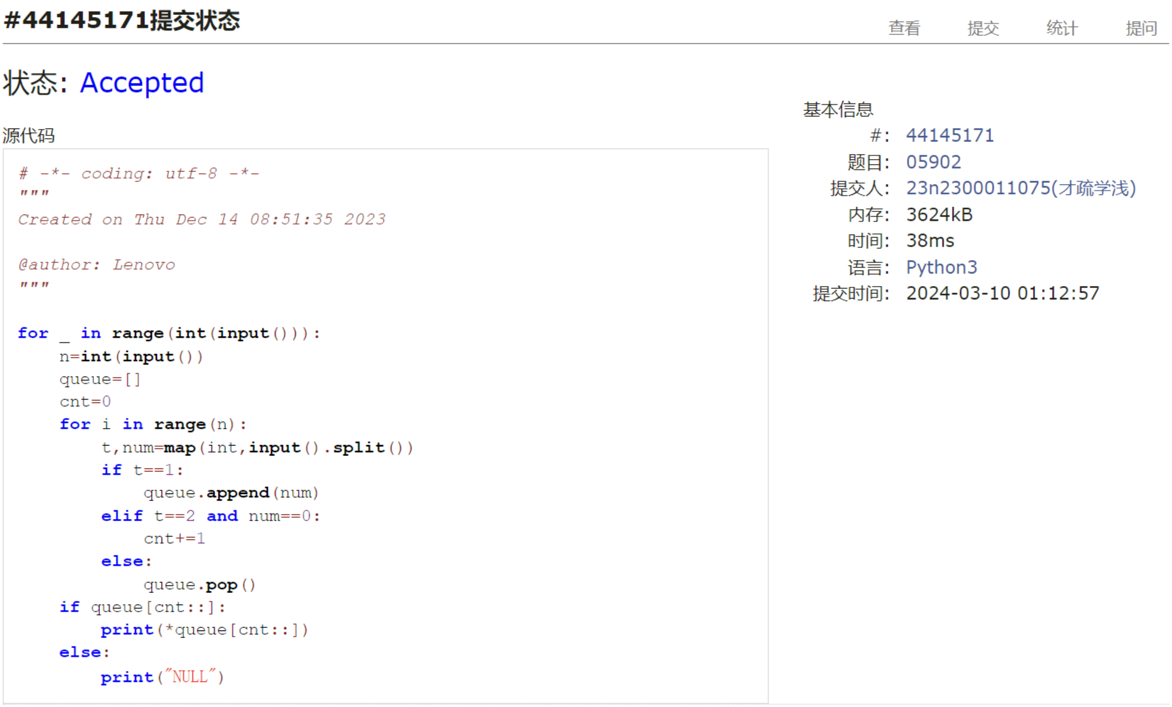
代码

```
# # -*- coding: utf-8 -*-
"""
Created on Thu Dec 14 08:51:35 2023

@author: Lenovo
"""

for _ in range(int(input())):
    n=int(input())
    queue=[]
    cnt=0
    for i in range(n):
        t,num=map(int,input().split())
        if t==1:
            queue.append(num)
        elif t==2 and num==0:
            cnt+=1
        else:
            queue.pop()
    if queue[cnt::]:
        print(*queue[cnt::])
    else:
        print("NULL")
```

代码运行截图 ==（至少包含有"Accepted"）==



#44145171提交状态

状态: Accepted

源代码

```
# -*- coding: utf-8 -*-
"""
Created on Thu Dec 14 08:51:35 2023

@author: Lenovo
"""

for _ in range(int(input())):
    n=int(input())
    queue=[]
    cnt=0
    for i in range(n):
        t,num=map(int,input().split())
        if t==1:
            queue.append(num)
        elif t==2 and num==0:
            cnt+=1
        else:
            queue.pop()
    if queue[cnt::]:
        print(*queue[cnt::])
    else:
        print("NULL")
```

基本信息

#:	44145171
题目:	05902
提交人:	23n2300011075(才疏学浅)
内存:	3624kB
时间:	38ms
语言:	Python3
提交时间:	2024-03-10 01:12:57

02694: 波兰表达式

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

思路：递归

代码

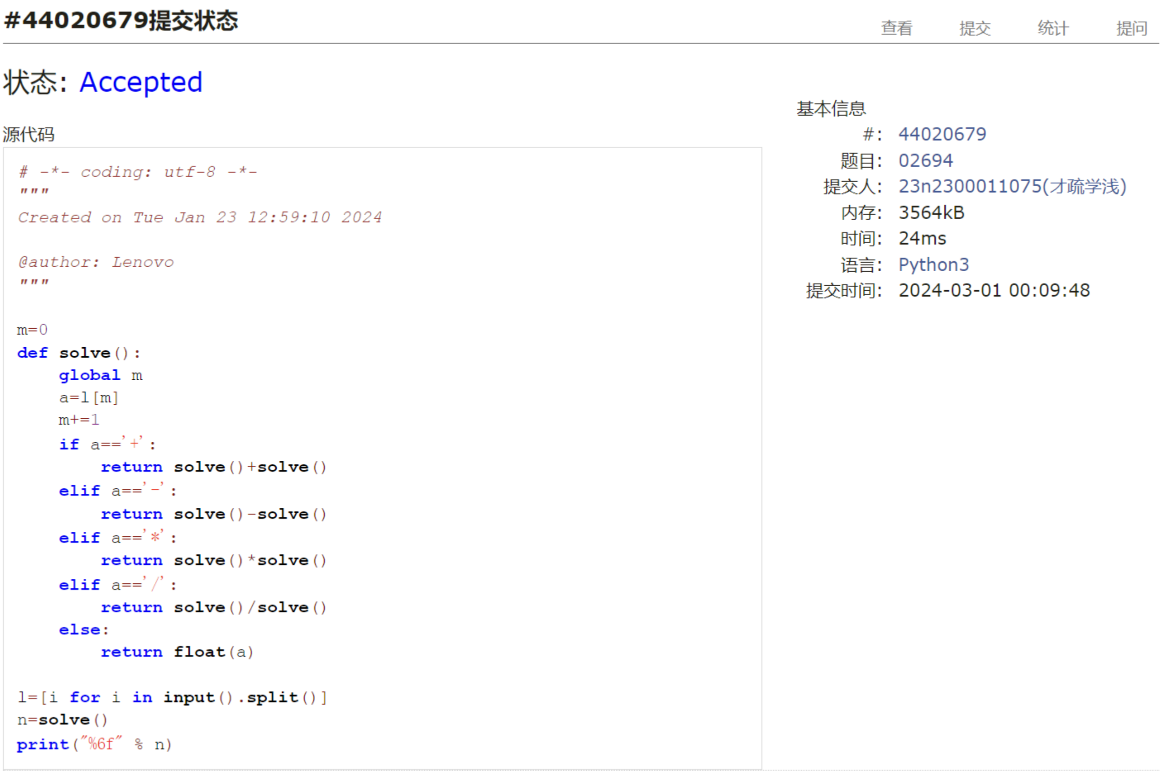
```
# # -*- coding: utf-8 -*-
"""
Created on Tue Jan 23 12:59:10 2024
```

```
@author: Lenovo
"""

m=0
def solve():
    global m
    a=l[m]
    m+=1
    if a=='+':
        return solve()+solve()
    elif a=='-':
        return solve()-solve()
    elif a=='*':
        return solve()*solve()
    elif a=='/':
        return solve()/solve()
    else:
        return float(a)

l=[i for i in input().split()]
n=solve()
print("%6f" % n)
```

代码运行截图 ==（至少包含有"Accepted"）==



#44020679提交状态

状态: Accepted

源代码

```
# -*- coding: utf-8 -*-
"""
Created on Tue Jan 23 12:59:10 2024

@author: Lenovo
"""

m=0
def solve():
    global m
    a=l[m]
    m+=1
    if a=='+':
        return solve()+solve()
    elif a=='-':
        return solve()-solve()
    elif a=='*':
        return solve()*solve()
    elif a=='/':
        return solve()/solve()
    else:
        return float(a)

l=[i for i in input().split()]
n=solve()
print("%6f" % n)
```

基本信息

#:	44020679
题目:	02694
提交人:	23n2300011075(才疏学浅)
内存:	3564kB
时间:	24ms
语言:	Python3
提交时间:	2024-03-01 00:09:48

24591: 中序表达式转后序表达式

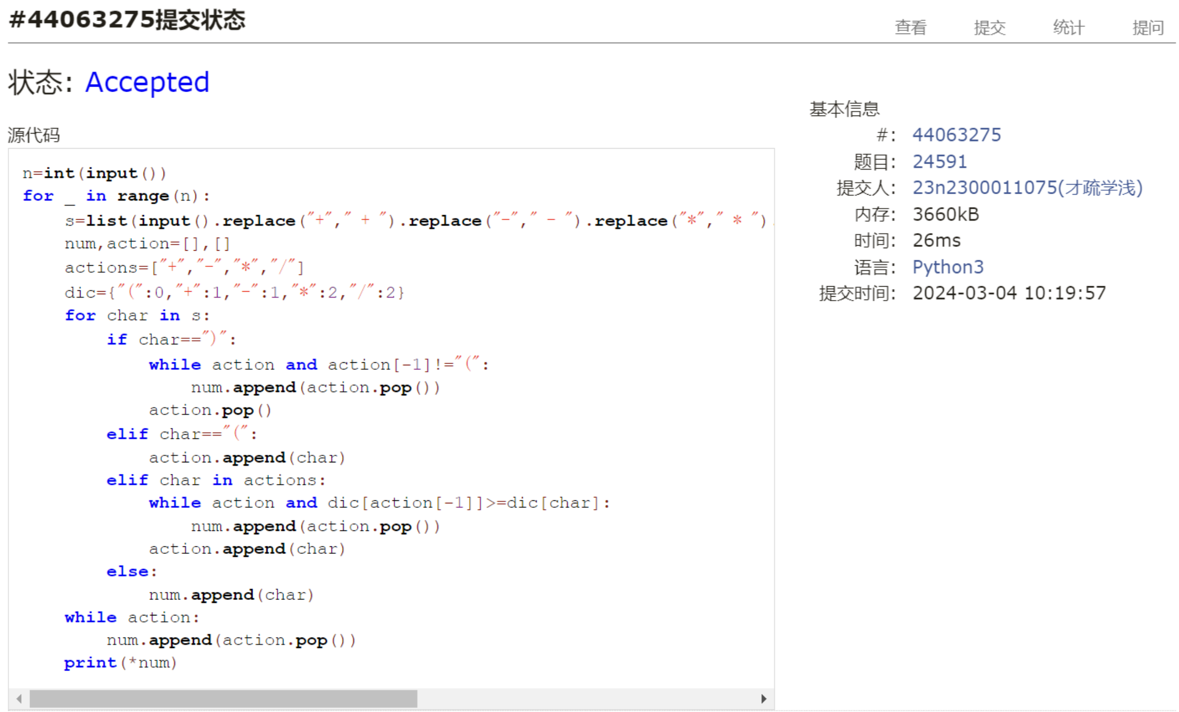
http://cs101.openjudge.cn/practice/24591/

思路：用栈模拟，对运算符级别分类

代码

```python
#
n=int(input())
for _ in range(n):
    s=list(input().replace("+"," + ").replace("-"," - ").replace("*"," * ").replace("/"," / ").replace("("," ( ").replace(")"," ) ").split())
    num,action=[],[]
    actions=["+","-","*","/"]
    dic={"(":0,"+":1,"-":1,"*":2,"/":2}
    for char in s:
        if char==")":
            while action and action[-1]!="(":
                num.append(action.pop())
            action.pop()
        elif char=="(":
            action.append(char)
        elif char in actions:
            while action and dic[action[-1]]>=dic[char]:
                num.append(action.pop())
            action.append(char)
        else:
            num.append(char)
    while action:
        num.append(action.pop())
    print(*num)
```

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==



#44063275提交状态

状态: Accepted

源代码

```
n=int(input())
for _ in range(n):
    s=list(input().replace("+"," + ").replace("-"," - ").replace("*"," * ")
    num,action=[],[]
    actions=["+","-","*","/"]
    dic={"(":0,"+":1,"-":1,"*":2,"/":2}
    for char in s:
        if char==")":
            while action and action[-1]!="(":
                num.append(action.pop())
            action.pop()
        elif char=="(":
            action.append(char)
        elif char in actions:
            while action and dic[action[-1]]>=dic[char]:
                num.append(action.pop())
            action.append(char)
        else:
            num.append(char)
    while action:
        num.append(action.pop())
    print(*num)
```

基本信息

#:	44063275
题目:	24591
提交人:	23n2300011075(才疏学浅)
内存:	3660kB
时间:	26ms
语言:	Python3
提交时间:	2024-03-04 10:19:57

22068: 合法出栈序列

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

思路：用栈模拟

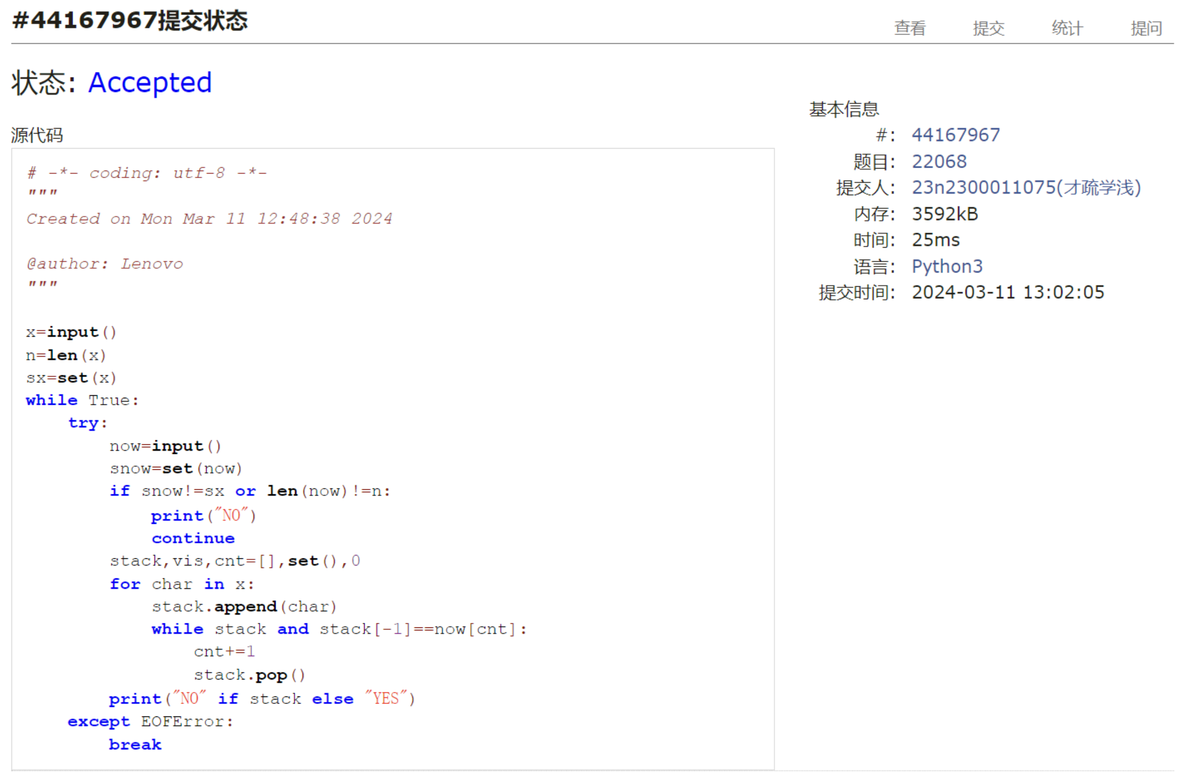
代码

```
# # -*- coding: utf-8 -*-
"""
Created on Mon Mar 11 12:48:38 2024

@author: Lenovo
"""

x=input()
n=len(x)
sx=set(x)
while True:
    try:
        now=input()
        snow=set(now)
        if snow!=sx or len(now)!=n:
            print("NO")
            continue
        stack,vis,cnt=[],set(),0
        for char in x:
            stack.append(char)
            while stack and stack[-1]==now[cnt]:
                cnt+=1
                stack.pop()
        print("NO" if stack else "YES")
    except EOFError:
        break
```

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==



#44167967提交状态

状态: Accepted

源代码

```
# -*- coding: utf-8 -*-
"""
Created on Mon Mar 11 12:48:38 2024

@author: Lenovo
"""

x=input()
n=len(x)
sx=set(x)
while True:
    try:
        now=input()
        snow=set(now)
        if snow!=sx or len(now)!=n:
            print("NO")
            continue
        stack,vis,cnt=[],set(),0
        for char in x:
            stack.append(char)
            while stack and stack[-1]==now[cnt]:
                cnt+=1
                stack.pop()
        print("NO" if stack else "YES")
    except EOFError:
        break
```

基本信息

#:	44167967
题目:	22068
提交人:	23n2300011075(才疏学浅)
内存:	3592kB
时间:	25ms
语言:	Python3
提交时间:	2024-03-11 13:02:05

06646: 二叉树的深度

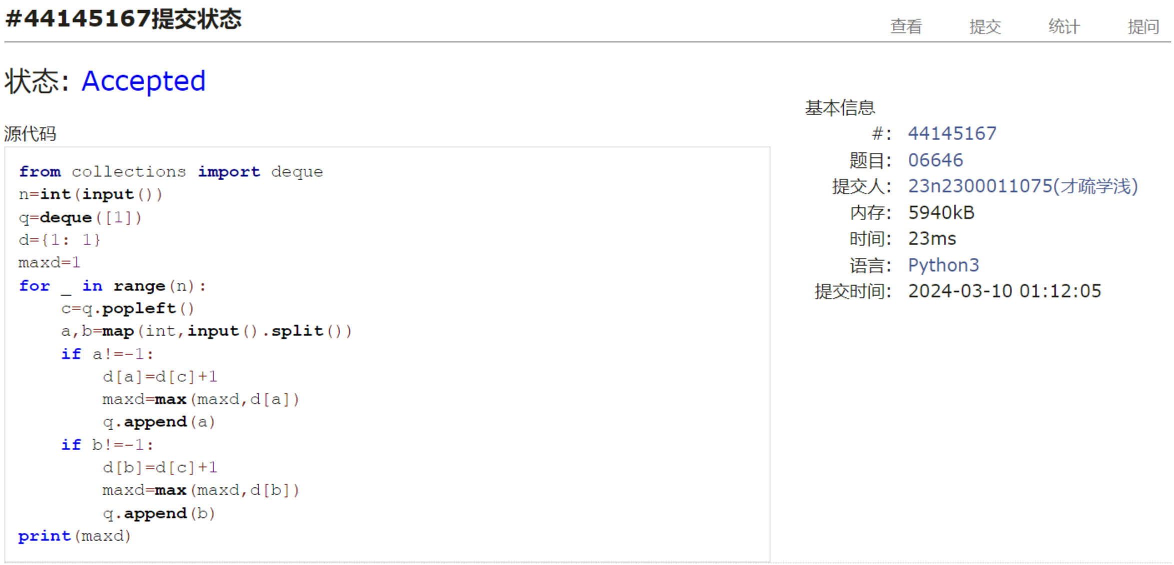
http://cs101.openjudge.cn/practice/06646/

思路：按顺序则可用队列模拟

代码

```
#
from collections import deque
n=int(input())
q=deque([1])
d={1: 1}
maxd=1
for _ in range(n):
    c=q.popleft()
    a,b=map(int,input().split())
    if a!=-1:
        d[a]=d[c]+1
        maxd=max(maxd,d[a])
        q.append(a)
    if b!=-1:
        d[b]=d[c]+1
        maxd=max(maxd,d[b])
        q.append(b)
print(maxd)
```

代码运行截图 ==（AC代码截图，至少包含有"Accepted"） ==



#44145167提交状态

状态: Accepted

源代码

```
from collections import deque
n=int(input())
q=deque([1])
d={1: 1}
maxd=1
for _ in range(n):
    c=q.popleft()
    a,b=map(int,input().split())
    if a!=-1:
        d[a]=d[c]+1
        maxd=max(maxd,d[a])
        q.append(a)
    if b!=-1:
        d[b]=d[c]+1
        maxd=max(maxd,d[b])
        q.append(b)
print(maxd)
```

基本信息

#:	44145167
题目:	06646
提交人:	23n2300011075(才疏学浅)
内存:	5940kB
时间:	23ms
语言:	Python3
提交时间:	2024-03-10 01:12:05

02299: Ultra-QuickSort

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

思路：归并排序 or bisect

代码

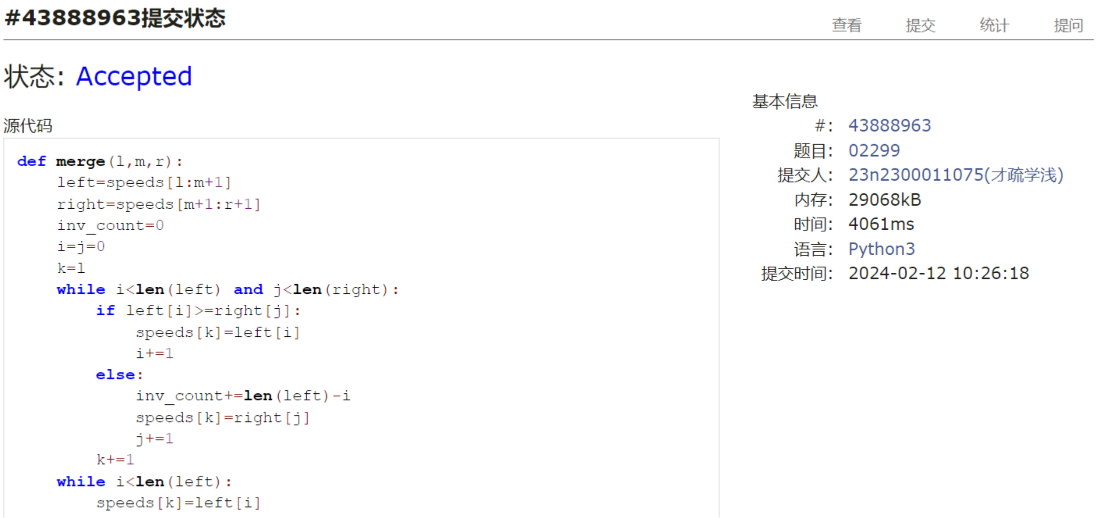
```
#
def merge(l,m,r):
    left=speeds[l:m+1]
    right=speeds[m+1:r+1]
    inv_count=0
    i=j=0
    k=l
    while i<len(left) and j<len(right):
        if left[i]>=right[j]:
            speeds[k]=left[i]
            i+=1
        else:
            inv_count+=len(left)-i
            speeds[k]=right[j]
            j+=1
        k+=1
    while i<len(left):
        speeds[k]=left[i]
        i+=1
        k+=1
    while j<len(right):
        speeds[k]=right[j]
        j+=1
        k+=1
    return inv_count

def count(l,r):
    inv_count=0
    if l<r:
        m=(l+r)//2
        inv_count+=count(l,m)
        inv_count+=count(m+1,r)
        inv_count+=merge(l,m,r)
    return inv_count

while True:
    n=int(input())
    if n==0:
        break
    speeds=[0]*n
    for i in range(n-1,-1,-1):
        speeds[i]=int(input())
    result=count(0,n-1)
    print(result)


import bisect
while True:
    n=int(input())
    if n==0:
        break
    l=[0]*n
    for i in range(n-1,-1,-1):
        l[i]=int(input())
    lst,ans=[],0
    for num in l:
        i=bisect.bisect_left(lst,num)
        bisect.insort_left(lst,num)
        ans+=i
    print(ans)
```

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==



#43888963提交状态

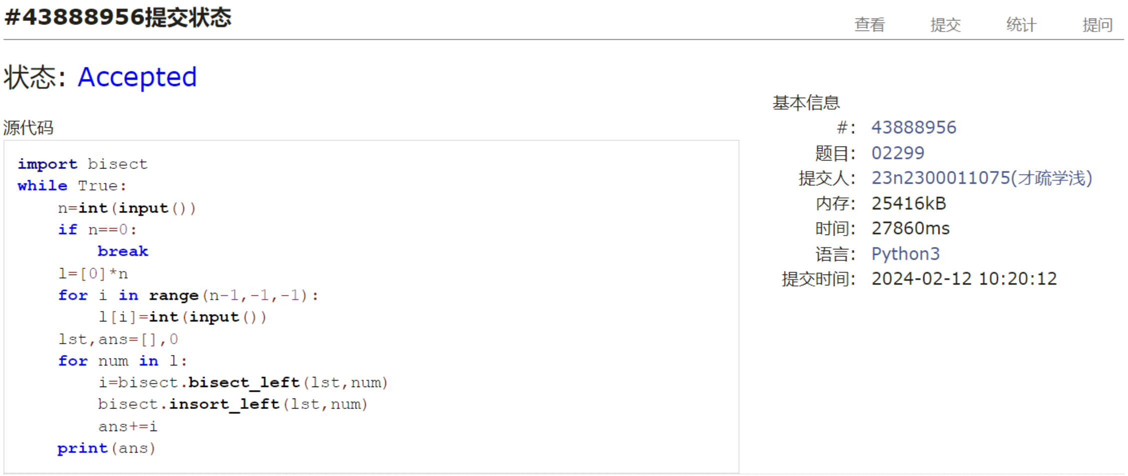
状态: Accepted

源代码

```
def merge(l,m,r):
    left=speeds[l:m+1]
    right=speeds[m+1:r+1]
    inv_count=0
    i=j=0
    k=l
    while i<len(left) and j<len(right):
        if left[i]>=right[j]:
            speeds[k]=left[i]
            i+=1
        else:
            inv_count+=len(left)-i
            speeds[k]=right[j]
            j+=1
        k+=1
    while i<len(left):
        speeds[k]=left[i]
```

基本信息

#:	43888963
题目:	02299
提交人:	23n2300011075(才疏学浅)
内存:	29068kB
时间:	4061ms
语言:	Python3
提交时间:	2024-02-12 10:26:18



#43888956提交状态

状态: Accepted

源代码

```
import bisect
while True:
    n=int(input())
    if n==0:
        break
    l=[0]*n
    for i in range(n-1,-1,-1):
        l[i]=int(input())
    lst,ans=[],0
    for num in l:
        i=bisect.bisect_left(lst,num)
        bisect.insort_left(lst,num)
        ans+=i
    print(ans)
```

基本信息

#:	43888956
题目:	02299
提交人:	23n2300011075(才疏学浅)
内存:	25416kB
时间:	27860ms
语言:	Python3
提交时间:	2024-02-12 10:20:12

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。==

近期新题练习颇少，在看数据结构，写了一道综合性的树题目四分树

```
# -*- coding: utf-8 -*-
"""
Created on Sun Mar 10 09:42:48 2024

@author: Lenovo
"""

from collections import deque
class Node:                             #节点
    def __init__(self):
        self.value=None
        self.childs=[]

def summ(matrix,n):                     #矩阵求和
    num=0
    for i in range(n):
        num+=sum(matrix[i])
    return num

def build(matrix,n):                    #建树
```

```
    node=Node()
    num=summ(matrix,n)
    if num==0:
        node.value="00"
    elif num==n*n:
        node.value="01"
    else:                               #递归求四个子矩阵的节点信息
        node.value="1"
        node.childs.append(build([matrix[i][:n//2] for i in range(n//2)],n//2))
        node.childs.append(build([matrix[i][n//2:] for i in range(n//2)],n//2))
        node.childs.append(build([matrix[i][:n//2] for i in range(n//2,n)],n//2))
        node.childs.append(build([matrix[i][n//2:] for i in range(n//2,n)],n//2))
    return node

def Print(root):                        #用队列进行层序遍历
    dic={"0000":"0","0001":"1","0010":"2","0011":"3","0100":"4","0101":"5","0110":"6","0111":"7","1000":"8","1001":"9","1010":"A","1011":"B","1100":"C","1101":"D","1110":"E","1111":"F"}       #简便2转16
    ans=""
    q=deque()
    q.append(root)
    while q:
        node=q.popleft()
        ans+=node.value
        if node.childs:
            for child in node.childs:
                q.append(child)
    ans="0"*(4-len(ans)%4)+ans
    res=""
    for i in range(0,len(ans),4):
        res+=dic[ans[i:i+4]]
    return res

k=int(input())
for _ in range(k):
    n=int(input())
    matrix=[list(map(int,input().split())) for i in range(n)]
    Tree=build(matrix,n)
    ans=Print(Tree)
    print(ans)
```