

Assignment #3: March月考

2024 spring, Compiled by 胡豪俊

说明:

- 1) The complete process to learn DSA from scratch can be broken into 4 parts:
 - Learn about Time and Space complexities
 - Learn the basics of individual Data Structures
 - Learn the basics of Algorithms
 - Practice Problems on DSA
- 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: Windows11

Python编程环境: Visual Studio Code

1. 题目

02945: 拦截导弹

<http://cs101.openjudge.cn/practice/02945/>

代码

```
k=int(input())
l=list(map(int,input().split()))
dp=[0]*k
for i in range(k-1,-1,-1):
    maxn=1
    for j in range(k-1,i,-1):
        if l[i]>=l[j] and dp[j]+1>maxn:
            maxn=dp[j]+1
    dp[i]=maxn
print(max(dp))
```

##dp在当年学计算概论的时候就没学好，依稀记得期末唯一没能ac的就是一道dp，现在更是忘得差不多了。我估计在规定时间内没法自己解决，就先参考了题解代码，然后学习了老师发在群里的dp问题

状态: Accepted

源代码

```
k=int(input())
l=list(map(int,input().split()))
dp=[0]*k
for i in range(k-1,-1,-1):
    maxn=1
    for j in range(k-1,i,-1):
        if l[i]>=l[j] and dp[j]+1>maxn:
            maxn=dp[j]+1
    dp[i]=maxn
print(max(dp))
```

©2002-2022 POJ 京ICP备20010980号-1

[English](#) [帮助](#) [关于](#)

基本信息

#: 44179852
题目: 02945
提交人: 22n2200011800
内存: 3616kB
时间: 24ms
语言: Python3
提交时间: 2024-03-12 10:31:47

04147:汉诺塔问题(Tower of Hanoi)

<http://cs101.openjudge.cn/practice/04147>

代码

```
def moveOneStep(numDisk:int,initiation:str,destination:str):
    print("{}: {}->{}".format(numDisk,initiation,destination))

def move(numDisks:int,initiation:str,temptation:str,destination:str):
    if numDisks==1:
        moveOneStep(1,initiation,destination)
    else:
        move(numDisks-1,initiation,destination,temptation)
        moveOneStep(numDisks,initiation,destination)
        move(numDisks-1,temptation,initiation,destination)
n,a,b,c=input().split()
move(int(n),a,b,c)
```

##这题的思路提示里讲的非常清楚,就是反复迭代,主要操作有二,一是将n-1个盘子移到temptation上,然后把最底下的盘子(编号n)移到destination上,此时他们的出发点都是initiation
##第二步相当于把n-1个盘子从temptation移动到destination,当然要借助initiation,如此递归即可。具体的代码实现参考了一下老师发的题解,我自己写的时候函数的temptation没定义好,出错了几次

状态: Accepted

源代码

```
def moveOneStep(numDisk:int,initiation:str,destination:str):
    print("{}: {}->{}".format(numDisk,initiation,destination))

def move(numDisks:int,initiation:str,temptation:str,destination:str):
    if numDisks==1:
        moveOneStep(1,initiation,destination)
    else:
        move(numDisks-1,initiation,destination,temptation)
        moveOneStep(numDisks,initiation,destination)
        move(numDisks-1,temptation,initiation,destination)
n,a,b,c=input().split()
move(int(n),a,b,c)
```

©2002-2022 POJ 京ICP备20010980号-1

[English](#) [帮助](#) [关于](#)

基本信息

#: 44176651
题目: 04147
提交人: 22n2200011800
内存: 3628kB
时间: 21ms
语言: Python3
提交时间: 2024-03-11 22:24:53

03253: 约瑟夫问题No.2

<http://cs101.openjudge.cn/practice/03253>

代码

```
l=[]

while True:
    ans=""
    n,p,m=map(int,input().split())
    x=n
    if n==0 and p==0 and m==0:
        break
    else:
        L=[int(i) for i in range(1,n+1)]
        for i in range(x):
            p=(p+m-1)%n

            x=L.pop(p-1)
            if p==0:
                p=1##这里要考虑一下如果报到最后一个人，要重新开始，把指标切换成1
            n-=1
            ans+=str(x)+", "
        l.append(ans[:-1])
for x in l:
    print(x)

##这题一开始边输入边输出错了好几次，看到老师发在群里的代码才知道要一起输出...
```

#44176630提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

源代码

```
l=[]

while True:
    ans=""
    n,p,m=map(int,input().split())
    x=n
    if n==0 and p==0 and m==0:
        break
    else:
        L=[int(i) for i in range(1,n+1)]
        for i in range(x):
            p=(p+m-1)%n

            x=L.pop(p-1)
            if p==0:
                p=1
            n-=1
            ans+=str(x)+", "
        l.append(ans[:-1])
for x in l:
    print(x)
```

基本信息

#: 44176630
题目: 03253
提交人: 22n2200011800
内存: 3656kB
时间: 24ms
语言: Python3
提交时间: 2024-03-11 22:24:10

©2002-2022 POJ 京ICP备20010980号-1

[English](#) [帮助](#) [关于](#)

21554:排队做实验 (greedy)v0.2

<http://cs101.openjudge.cn/practice/21554>

代码

```
n=int(input())
L=[int(i) for i in input().split()]
T={}
ans=""
sum=0
```

```

for i in range(n):
    if L[i] not in T.keys():
        T[L[i]]=i+1
    else:
        T[L[i]]+=i+1
L.sort()
for i in range(n):
    sum+=L[i]*(n-i-1)
L=list(set(L))
L.sort()
for t in L:
    for x in T[t]:
        ans+=str(x)+" "
average=sum/n
print(ans[:-1])
print('%.2f' % average)

```

##一个简单的贪心，具体体现在让用时少的同学先做实验，事实上给出实验时间的列表并排序后平均等待时间是可以写的，麻烦一点的是同学的做的顺序，这里我还是采用了字典，和前面打怪兽一样的思路，因为可能会出现相同时间的同学，字典的值要取成一个列表，最后根据键输出值中的元素即可

##这题做的时候脑子不是很清醒，错得莫名其妙的，字典的值应写成T[x]而非T(x)，最后输出结果用空格隔开而非逗号，这两个点卡了我很久，查了很久的题解，最后发现不是思路错了是细节错了。但是还是有收获的，比如想要输出空格隔开的数只需要print(*L)，这样列表的元素就会自动输出并且间隔空格

#44177636提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

源代码

```

n=int(input())
L=[int(i) for i in input().split()]
T={}
ans=""
sum=0
for i in range(n):
    if L[i] not in T.keys():
        T[L[i]]=i+1
    else:
        T[L[i]]+=i+1
L.sort()
for i in range(n):
    sum+=L[i]*(n-i-1)
L=list(set(L))
L.sort()
for t in L:
    for x in T[t]:
        ans+=str(x)+" "
average=sum/n
print(ans[:-1])
print('%.2f' % average)

```

基本信息

#: 44177636
 题目: 21554
 提交人: 22n2200011800
 内存: 3672kB
 时间: 24ms
 语言: Python3
 提交时间: 2024-03-11 23:37:24

©2002-2022 POJ 京ICP备20010980号-1

[English](#) [帮助](#) [关于](#)

19963:买学区房

<http://cs101.openjudge.cn/practice/19963>

代码

```

n=int(input())
ans=[1]*n
L=[i[1:-1] for i in input().split()]
D=[sum(map(int,i.split(','))) for i in L]
Price=[int(i) for i in input().split()]
Pricee=Price[0:]
Pricee.sort()

```

```

Value=[D[i]/Price[i] for i in range(n)]
valuee=Value[0:]
valuee.sort()
if n%2==1:
    middle1=Valuee[int(n/2)]
    middle2=Pricee[int(n/2)]
else:
    middle1=(Valuee[int(n/2)]+Valuee[int(n/2-1)])/2
    middle2=(Pricee[int(n/2)]+Pricee[int(n/2-1)])/2

for i in range(n):
    if value[i]<=middle1 or Price[i]>=middle2:
        ans[i]=0
num=ans.count(1)
if num==-1:
    num=0
print(num)

```

##过程就是找出题目定义的两个中位数然后进行比较，最初始的输入列表其实暗含了编号的，为了方便不用index这种我不太熟练怕出错的函数我就保留了原列表，最后再计数可行的情况

#44179220提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

源代码

```

n=int(input())
ans=[1]*n
L=[i[1:-1] for i in input().split()]
D=[sum(map(int,i.split('.'))) for i in L]
Price=[int(i) for i in input().split()]
Pricee=Price[0:]
Pricee.sort()
Value=[D[i]/Price[i] for i in range(n)]
Valuee=Value[0:]
Valuee.sort()
if n%2==1:
    middle1=Valuee[int(n/2)]
    middle2=Pricee[int(n/2)]
else:
    middle1=(Valuee[int(n/2)]+Valuee[int(n/2-1)])/2
    middle2=(Pricee[int(n/2)]+Pricee[int(n/2-1)])/2

for i in range(n):
    if Value[i]<=middle1 or Price[i]>=middle2:
        ans[i]=0
num=ans.count(1)
if num==-1:
    num=0
print(num)

```

基本信息

#: 44179220
 题目: 19963
 提交人: 22n2200011800
 内存: 4292kB
 时间: 25ms
 语言: Python3
 提交时间: 2024-03-12 09:34:20

©2002-2022 POJ 京ICP备20010980号-1

[English](#) [帮助](#) [关于](#)

27300: 模型整理

<http://cs101.openjudge.cn/practice/27300>

代码

```

n=int(input())
kind={}
key=[]
ans=[]
for i in range(n):
    model, argu=map(str, input().split("-"))
    key.append(model)
    if model in kind.keys():

```

```

        kind[model]+=argu
    else:
        kind[model]=[argu]
key=list(set(key))
key.sort()

for t in key:
    s=t+": "
    num={}
    m=[]
    for x in kind[t]:
        rank=x[-1]
        if rank=="M":
            numb=float(x[:-1])*1000000
        if rank=="B":
            numb=float(x[:-1])*1000000000
        m.append(numb)
        num[numb]=x
    m.sort()
    for t in m:
        s+=num[t]+" "
    ans.append(s)
for t in ans:
    print(t[:-2])

```

##利用字典先存入不同的模型名称，和对应的模型参数（用列表储存作值），和打怪兽还有做实验一样的思路。题目里给的限制比较好，事实上B是严格大于M的，加一个判断能省不少算力，而且我的代码中将参数数值化的一步其实也可以在输入过程中完成，这样应该还能继续优化省时间

#44179570提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

源代码

```

n=int(input())
kind={}
key=[]
ans=[]
for i in range(n):
    model,argu=map(str,input().split("-"))
    key.append(model)
    if model in kind.keys():
        kind[model]+=argu
    else:
        kind[model]=[argu]
key=list(set(key))
key.sort()

for t in key:
    s=t+": "
    num={}
    m=[]
    for x in kind[t]:
        rank=x[-1]
        if rank=="M":
            numb=float(x[:-1])*1000000
        if rank=="B":
            numb=float(x[:-1])*1000000000
        m.append(numb)
        num[numb]=x
    m.sort()
    for t in m:
        s+=num[t]+" "
    ans.append(s)
for t in ans:
    print(t[:-2])

```

基本信息

#: 44179570
 题目: 27300
 提交人: 22n2200011800
 内存: 3656kB
 时间: 22ms
 语言: Python3
 提交时间: 2024-03-12 10:10:35

2. 学习总结和收获

感觉慢慢进入状态了，这次月考因为撞课我没能去机房参加，而且由于时间安排失当导致在ddl最后一天才断断续续完成。初步估计在考场上要是进入考试状态大约能ac4或者ac5的样子，当然前提是不犯那些愚蠢的错误。接下来数算会投入更多的时间，选做会慢慢跟上节奏，而且我非常感谢群里大佬的分享，每次点开群都有收获。