

# Assignment #3: March月考

---

Updated 1537 GMT+8 March 6, 2024

2024 spring, Compiled by ==吴至超 城环

## 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

==（请改为同学的操作系统、编程环境等）==

操作系统: Windows 11

Python编程环境: pycharm2023.2.3

## 1. 题目

---

### 02945: 拦截导弹

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

思路: 动态规划

1. 建立一个长度为导弹个数的列表dp，初始值都为1（省去遍历第一个），dp[i]意义在于：对于missile[0:i+1]的最长减序列的长度

2. 状态转移，局部最优解逐渐扩大为全局最优解。对missile[i]以前的每一个数字遍历，如果missile[j]>=missile[i],表明原最优解可以再加1变长

一次次遍历完之后，得到新的局部最优解

代码

```
k=int(input())
missile=[int(x) for x in input().split()]
dp=[1]*k

for i in range(1,k):#循环，dp[i]表示[0，i]的最长增序列
    for j in range(i):
        if missile[i]<=missile[j]:
            dp[i]=max(dp[i],dp[j]+1)
print(max(dp))
```

代码运行截图 ==（至少包含有"Accepted"） ==

OpenJudge

题目ID, 标题, 描述

23n2300013289

信封

账号

CS101 / 题库

题目 排名 状态 提问

02945:拦截导弹

查看 提交 统计 提问

总时间限制: 1000ms 内存限制: 65536kB

描述

某国为了防御敌国的导弹袭击，开发出一种导弹拦截系统。但是这种导弹拦截系统有一个缺陷：虽然它的第一发炮弹能够达到任意的高度，但是以后每一发炮弹都不能高于前一发的高度。某天，雷达捕捉到敌国的导弹来袭，并观测到导弹依次飞来的高度，请计算这套系统最多能拦截多少导弹。拦截来袭导弹时，必须按来袭导弹袭击的时间顺序，不允许先拦截后面的导弹，再拦截前面的导弹。

输入

输入有两行。  
第一行，输入雷达捕捉到的敌国导弹的数量k（k<=25），  
第二行，输入k个正整数，表示k枚导弹的高度，按来袭导弹的袭击时间顺序给出，以空格分隔。

输出

输出只有一行，包含一个整数，表示最多能拦截多少枚导弹。

样例输入

8  
300 207 155 300 299 170 158 65

样例输出

6

来源

医学部计算概论2006期末考试题

查看 提交 统计 提问

全局题号 1947

添加于 2023-11-17

提交次数 130

尝试人数 71

通过人数 68

你的提交记录

#	结果	时间
1	Accepted	2024-03-10

© 2002-2022 POJ 京ICP备20010980号-1

English 帮助 关于

04147:汉诺塔问题(Tower of Hanoi)

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

思路：递归思想，过渡柱和目标柱的变换有点复杂。

代码

```
#汉诺塔
n,a,b,c=map(str,input().split())
n=int(n)
def hannuo(n,a,b,c):#n层, a, b, c柱子
    if n==1:
        print(f"1:{a}->{c}")
    else:
        hannuo(n-1,a,c,b)
        print(f"{n}:{a}->{c}")
        hannuo(n-1,b,a,c)
hannuo(n,a,b,c)
```

代码运行截图 == (至少包含有"Accepted") ==

OpenJudge 题目ID, 标题, 描述 23n2300013289 信箱 账号

CS101 / 题库 题目 排名 状态 提问

04147:汉诺塔问题(Tower of Hanoi) 查看 提交 统计 提问

总时间限制: 1000ms 内存限制: 65535kB

描述

一、汉诺塔问题

有三根杆子A、B、C。A杆上有N个(N>1)穿孔圆盘，盘的尺寸由下到上依次变小。要求按下列规则将所有圆盘移至C杆： 每次只能移动一个圆盘； 大盘不能叠在小盘上面。 提示：可将圆盘临时置于B杆，也可将从A杆移出的圆盘重新移回A杆，但都必须遵循上述两条规则。

问：如何移？ 最少要移动多少次？

汉诺塔示意图如下：



三个盘的移动：



二、故事由来

全局题号 8200

添加于 2021-11-22

提交次数 196

尝试人数 137

通过人数 133

你的提交记录

#	结果	时间
3	Accepted	2024-03-06
2	Wrong Answer	2024-03-06
1	Wrong Answer	2024-03-06

03253: 约瑟夫问题No.2

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

思路：一边进另一边出。

当列表长度大于p时，踢去p位置的人

当列表长度小于p时，踢去末位置的人

代码

```
flag=1

while flag==1:
    pp = []
    n,p,m=(int(x) for x in input().split())
    if n==0:
        flag=0
        break
    else:
        alist=[]
        alist=list(range(1,n+1))
        while len(alist)>=1:
            if len(alist) >= p:
                for u in range(1,m):
                    alist.insert(len(alist)-1,alist.pop(0))
                pp.append(alist.pop(p-1))
            else:
                for u in range(1,m+1):
                    alist.insert(len(alist) - 1, alist.pop(0))
                pp.append(alist.pop())

        print(",".join(map(str,pp)))
```

代码运行截图 == (AC代码截图，至少包含有"Accepted") ==

OpenJudge

题目ID, 标题, 描述

23n2300013289

信箱

账号

CS101 / 题库

题目 排名 状态 提问

03253:约瑟夫问题No.2

查看 提交 统计 提问

总时间限制: 1000ms 内存限制: 65536kB

描述

n 个小孩围坐成一圈，并按顺时针编号为1,2,...,n，从编号为 p 的小孩顺时针依次报数，由1报到m，当报到 m 时，该小孩从圈中出去，然后下一个再从1报数，当报到 m 时再出去，如此反复，直至所有的小孩都从圈中出去。请按出去的先后顺序输出小孩的编号。

输入

每行是用空格分开的三个整数，第一个是n,第二个是p,第三个是m (0 < m,n < 300)。最后一行是:  
0 0 0

输出

按出圈的顺序输出编号，编号之间以逗号间隔。

样例输入

```
8 3 4
0 0 0
```

样例输出

```
6, 2, 7, 4, 3, 5, 1, 8
```

提示

可以利用模拟方法解题。  
与 [http://net.pku.edu.cn/~course/cs101/book2/pp\\_list.txt](http://net.pku.edu.cn/~course/cs101/book2/pp_list.txt) 中的 poj 2746 Joseph issue极为相似。

来源

cs10107 C++ Final Exam

查看 提交 统计 提问

全局题号 2255

添加于 2020-12-15

提交次数 222

尝试人数 119

通过人数 109

你的提交记录

#	结果	时间
2	Accepted	2024-03-06
1	Wrong Answer	2024-03-06

21554:排队做实验 (greedy)v0.2

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

思路：贪心，观察样例也可以猜出。

对于n个人，壹人消耗时间为t1，貳人消耗t2(大于t1)，如果把貳人放在第一位，那团队耗时 (n-1) t2，肯定大于壹人在前只需消耗 (n-1) t1

代码

```
n=int(input())
T=[int(x) for x in input().split()]
for i in range(len(T)):
    T[i]=[i+1,T[i]]
T.sort(key=lambda x : (x[1],x[0]))

cac=0

if len(T)==1:
    print(T[0][0])
    print("0.00")
else:
    for m in range(len(T)-1):
        print(T[m][0],end=" ")
        cac+=(len(T)-m-1)*T[m][1]

    print(T[-1][0])
    print(f"{cac/n:.2f}")
```

代码运行截图 == (AC代码截图，至少包含有"Accepted") ==

OpenJudge

题目ID, 标题, 描述

23n2300013289

信箱

账号

CS101 / 20240306 cs201 2024 Mock Exam 已经结束

题目

排名

状态

统计

提问

M21554:排队做实验 (greedy)v0.2

查看

提交

统计

提问

总时间限制: 1000ms 内存限制: 65536kB

描述

某学院的学生都需要在某月的1号到2号期间完成课程实验，他们每个人的实验需要持续不同的时长，而实验室管理员要安排这些学生，按照一定顺序，在1号到2号期间全部完成实验。假设该学院的实验室同时只能容纳一名学生实验，而这些学生都非常积极，都希望被排在1号的第一个尽快完成实验。

假设该学院有n个学生，实验室管理员收到了n个学生每位需要占用实验室的时长T1,T2,...,Tn。由于学生发送预约邮件的时间比较接近，没办法完全按照先到先得的办法给学生分配实验时间（实验时间相同的话，先到先得）。管理员很犯愁，他希望有一种能让所有学生平均等待时间尽可能小的顺序，来安排这n位同学的实验时间，请问你能帮他吗？

描述部分对应英文：

All the students of a certain college need to complete their course experiments between the 1st and the 2nd of a certain month, and each of them needs to last for a different length of time. The lab manager has to arrange for all these students to complete their experiments in a certain order between the 1st and the 2nd. Suppose that the college's lab can only accommodate one student at a time, and that all of these students are very motivated and want to be the first in line to complete the experiment as soon as possible.

Assume that there are n students in the college and the lab manager has received the duration each of the n students needs to occupy the lab T1,T2,...,Tn. There is no way to assign lab times to students on a first-come, first-served basis because the students send their appointment emails close together. The administrator is very worried, he wants to have a sequence that can make the average waiting time for all students as small as possible to arrange the lab time for these n students.

输入

输入为2行  
第一行为n，为学生人数(n<=1000)  
第二行分别表示第一位学生到第n位学生的实验时长T1, T2, ..., Tn，每个数据间有一个空格

输出

输出为2行

19963:买学区房

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

思路：正常逻辑，就是要避免 $a=[1,2,3]$ ,  $f=a$ ,  $f.sort()$ 会对 $a$ 进行排序的情况（指针），如要避免，新建一个 $f$ ，然后把 $a$ 的元素一个个添加进去。

## 代码

```
n=int(input())
distance=[x[1:len(x)-1] for x in input().split()]
price=[int(x) for x in input().split()]
newdistance=[]
for i in distance:
    x,y=i.split(",")
    x,y=int(x),int(y)
    newdistance.append(x+y)
price_pos=[]
for m in range(n):
    price_pos.append(newdistance[m]/price[m])
#找中位数
ee=[int(x) for x in price]
ee.sort()
ff=[float(x) for x in price_pos]
ff.sort()
if n%2==0:
    price_mid=(ee[n//2]+ee[n//2-1])/2
    price_posmid=(ff[n//2]+ff[(n//2)-1])/2
else:
    price_mid=ee[(n-1)//2]
    price_posmid=ff[(n-1)//2]
cnt=0
for p in range(n):
    if price_pos[p]>price_posmid and price[p]<price_mid:
        cnt+=1
print(cnt)
```

代码运行截图 == (AC代码截图，至少包含有"Accepted") ==

OpenJudge

题目ID, 标题, 描述

23n2300013289

信箱

账号

CS101 / 题库

题目

排名

状态

提问

19963:买学区房

查看

提交

统计

提问

总时间限制: 1000ms 内存限制: 65536kB

描述

小明同学的家长为了让小明同学接受更好的教育，最近在考虑买某重点中学附近的房子，已知他们买房子要考虑两个因素：房子离学校的距离，以及房子的价格。现在他们有一系列备选的房子，已知这些房子距离学校的x方向距离和y方向距离，以及每栋房子的价格。小明的家长认为，只有同时满足了以下两个条件的房子H买了才是不亏的：  
1.小明的家长比较精打细算，所以房子的性价比大于所用备选房子性价比的中位数（定义见下图）  
2.小明的家长想攒钱，所以房子的价格小于所有备选房子价格的中位数  
注：  
性价比 = 房子和学校之间的交通距离 / 房子的价格  
(由于该城市的街道布局接近方格状，且从学校到房子无法穿墙而过，房子H去学校的交通距离定义为，房子H距离学校的x方向距离和y方向距离的和)  
现在需要你帮小明的家长判断，一系列备选房子里，值得买房子有多少栋。

Even-sized population [ edit ]

Consider an ordered population of 10 data values [3, 6, 7, 8, 8, 10, 13, 15, 16, 20]. What are the 4-quantiles (the "quartiles") of this dataset?

Quartile	Calculation	Result
Zeroth quartile	Although not universally accepted, one can also speak of the zeroth quartile. This is the minimum value of the set, so the zeroth quartile in this example would be 3.	3
First quartile	The rank of the first quartile is $10 \times (1/4) = 2.5$ , which rounds up to 3, meaning that 3 is the rank in the population (from least to greatest values) at which approximately 1/4 of the values are less than the value of the first quartile. The third value in the population is 7.	7
Second quartile	The rank of the second quartile (same as the median) is $10 \times (2/4) = 5$ , which is an integer, while the number of values (10) is an even number, so the average of both the fifth and sixth values is taken—that is $(8+10)/2 = 9$ , though any value from 8 through to 10 could be taken to be the median.	9
Third quartile	The rank of the third quartile is $10 \times (3/4) = 7.5$ , which rounds up to 8. The eighth value in the population is 15.	15
Fourth quartile	Although not universally accepted, one can also speak of the fourth quartile. This is the maximum value of the set, so the fourth quartile in this example would be 20. Under the Nearest Rank definition of quantile, the rank of the fourth quartile is the rank of the biggest number, so the rank of the fourth quartile would be 10.	20

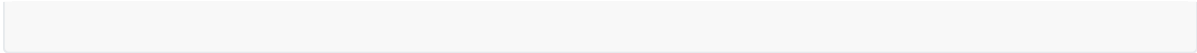
27300: 模型整理

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

思路：注意字典sorted（）会生成一个已经sort好的列表，正合我意。

代码

```
n=int(input())
dict={}
for i in range(n):
    m=input()
    x,y=m.split("-")
    if x not in dict:
        dict[x]=[]
    dict[x].append(y)
so=sorted(dict)
for i in so: #key
    jilu=[]
    for m in range(len(dict[i])): #value
        if dict[i][m][-1]=="B":
            jilu.append([dict[i][m],float(dict[i][m][0:-1:1])*1000000000])
        else:
            jilu.append([dict[i][m],float(dict[i][m][0:-1:1]) * 1000000])
    jilu.sort(key=lambda x: x[1])
    print(i+":",end=" ")
    for t in range(len(jilu)-1):
        print(jilu[t][0]+",",end=" ")
    print(jilu[-1][0])
```



代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==

OpenJudge

题目ID, 标题, 描述

23n2300013289

信箱

账号

CS101 / 题库

题目

排名

状态

提问

27300:模型整理

查看

提交

统计

提问

总时间限制: 1000ms 内存限制: 65536kB

描述

深度学习模型（尤其是大模型）是近两年计算机学术和业界热门的研究方向。每个模型可以用“模型名称-参数量”命名，其中参数量的单位会使用两种：M，即百万；B，即十亿。同一个模型通常有多个不同参数的版本。例如，Bert-110M, Bert-340M 分别代表参数量为 1.1 亿和 3.4 亿的 Bert 模型。GPT3-350M, GPT3-1.3B 和 GPT3-175B 分别代表参数量为 3.5亿, 13亿和 1750 亿的 GPT3 模型。参数量的数字部分取值在 [1, 1000] 区间（一个 8 亿参数的模型表示为 800M 而非 0.8B, 10 亿参数的模型表示为 1B 而非 1000M）。计算机专业的学生小 A 从网上收集了一份模型的列表，他需要将它们按照名称归类排序，并且同一个模型的参数量从小到大排序，生成“模型名称: 参数量1, 参数量2, ...”的列表。请你帮他写一个程序实现。

输入

第一行为一个正整数  $n$  ( $n \leq 1000$ )，表示有  $n$  个待整理的模型。

接下来  $n$  行，每行一个“模型名称-参数量”的字符串。模型名称是字母和数字的混合。

输出

每行一个“模型名称: 参数量1, 参数量2, ...”的字符串，符号均为英文符号，模型名称按字典序排列，参数量按从小到大排序。

样例输入

5  
GPT-1.3B  
Bert-340M  
GPT-350M  
Bert-110M  
GPT-175B

样例输出

Bert: 110M, 340M  
GPT: 350M, 1.3B, 175B

全局题号 27300

添加于 2023-12-28

提交次数 136

尝试人数 64

通过人数 62

Other language versions

English

你的提交记录

#	结果	时间
2	Accepted	2024-03:10
1	Wrong Answer	2024-03:10

## 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。==

至于月考，个人心理素质比较差，看到旁边大佬飞速做完会感觉到很强大的压迫感，希望下次月考能把心思更多放在自己思考的层面。

对递归与动态规划的理解并不是很好，底子很薄，学习完第一第二题之后又明白一些。第三题回来看课件做出1.0版本，2.0版本并不难。第四题也简单。第五第六题只是字多反而很简单，却被标成T，下次遇到不会做的别吊死，多看看其他的题目有没有思路。

原本打算周六周日要花时间在数算上的，不幸的是周六凌晨突然发烧，不得不休息。下周周中也可以多练练，不一定全砸在周末。