

2018 年春季《大学计算机基础》（理科）实验指导书

实验 6 复杂算法

1. 实验目的

- (1) 了解复杂算法的设计；
- (2) 掌握用 Python 语言进行算法实现。

2. 实验任务

实验任务 6-1 NULL 的奖学金

题目描述：

NULL 同学打算申请北航优秀奖学金，只有学习成绩优秀的年级前 5 名学生才能获得奖学金。每个学生都有 3 门课的成绩：数学分析、高等代数、概率统计。学生的排序方式是先按总分从高到低排序，如果两个同学总分相同，再按数学分析成绩从高到低排序，如果两个同学总分和数学分析成绩都相同，那么规定学号小的同学排在前面，这样，每个学生的排序是唯一确定的。

任务：先根据输入的 3 门课的成绩计算总分，然后按上述规则排序，最后按排名顺序输出前五名学生的学号和总分。注意，在前 5 名同学中，每个人的奖学金都不相同，因此，你必须严格按上述规则排序。

例如，在某个正确答案中，如果前两行的输出数据(每行输出两个数:学号、总分)是：

7 279

5 279

这两行数据的含义是：总分最高的两个同学的学号依次是 7 号、5 号，这两名同学的总分都是 279（总分等于输入的数学分析、高等代数、概率统计三科成绩之和），但学号为 7 的学生数学分析成绩更高一些。如果你的前两名的输出数据是：

5 279

7 279

则按输出错误处理，不能得分。

输入格式：

输入共有 $n+1$ 行。

第 1 行为一个正整数 n ，不超过 300，表示该校参加评选的学生人数。

第 2 到 $n+1$ 行，每行有 3 个用空格隔开的数字，每个数字都在 0 到 100 之间。第 j 行的 3 个数字依次表示学号为 $j-1$ 的学生的数学分析、高等代数、概率统计的成绩。每个学生的学号按照输入顺序编号为 $1\sim n$ （恰好是输入数据的行号减 1）。

所给的数据都是正确的，不必检验。

输出格式：

输出共有 5 行。

每行是两个用空格隔开的正整数，依次表示前 5 名学生的学号和总分。

【输入样例 1】

```
6
90 67 80
87 66 91
78 89 91
88 99 77
67 89 64
78 89 98
```

【输出样例 1】

```
6 265
4 264
3 258
2 244
1 237
```

【输入样例 2】

```
8
80 89 89
88 98 78
90 67 80
```

87 66 91

78 89 91

88 99 77

67 89 64

78 89 98

【输出样例 2】

8 265

2 264

6 264

1 258

5 258

实验指导：

1、学生信息采用什么数据结构存储？

为了便于排序处理，可以采用一个列表存储所有学生的三门课成绩，每个学生的成绩分别存储于该列表的子列表中。再采用另外两个列表存储学生的总成绩和学号。

当然，也可以把学生的三门课成绩、总分和学号存储在一个列表中。

也可以定义一个类，其中存放着学生的数学分析、高等代数、概率统计成绩、总成绩和学号。

2、如何按行输入每个学生的数学分析、高等代数、概率统计的成绩，3 个成绩之间用一个空格分隔？

可以采用 for 循环，在其中采用 “a,b,c=input().split()” 语句，按行输入每个学生三门课的成绩，按照空格将输入的多个数据分开，分别赋给等号左边的变量。

再采用列表的 append 方法，将输入添加到存储所有学生三门课成绩的列表中。

3、如何实现多个关键字的排序？

可以定义函数，采用选择排序算法实现多个关键字的排序。注意两个元素交换的条件：优先以总成绩排序；总成绩相同时以数学分析成绩为依据排序；如果两个同学总成绩和数学分析成绩都相同，则学号小的同学排在前面。

提示：为便于分析程序执行过程是否正确，建议调试时采用 print 函数打印出中间结果。如下所示。

```
>>>
8
80 89 89
88 98 78
90 67 80
87 66 91
78 89 91
88 99 77
67 89 64
78 89 98
scores= [[80, 89, 89], [88, 98, 78], [90, 67, 80], [87, 66, 91], [78, 89, 91], [88, 99, 77],
[67, 89, 64], [78, 89, 98]]
total_score= [258, 264, 237, 244, 258, 264, 220, 265]
Index= [0, 1, 2, 3, 4, 5, 6, 7]

i= 0 k= 7 , k不等于i, 则交换A[i]和A[k]的值, A变为: [265, 264, 237, 244, 258, 264, 220, 258]
第i轮排序后Index变为: [7, 1, 2, 3, 4, 5, 6, 0]
-----
i= 1 k= 1 , k等于i, 不交换A[i]和A[k]的值, A仍为: [265, 264, 237, 244, 258, 264, 220, 258]
-----
i= 2 k= 5 , k不等于i, 则交换A[i]和A[k]的值, A变为: [265, 264, 264, 244, 258, 237, 220, 258]
第i轮排序后Index变为: [7, 1, 5, 3, 4, 2, 6, 0]
-----
i= 3 k= 7 , k不等于i, 则交换A[i]和A[k]的值, A变为: [265, 264, 264, 258, 258, 237, 220, 244]
第i轮排序后Index变为: [7, 1, 5, 0, 4, 2, 6, 3]
-----
i= 4 k= 4 , k等于i, 不交换A[i]和A[k]的值, A仍为: [265, 264, 264, 258, 258, 237, 220, 244]
-----
i= 5 k= 7 , k不等于i, 则交换A[i]和A[k]的值, A变为: [265, 264, 264, 258, 258, 244, 220, 237]
第i轮排序后Index变为: [7, 1, 5, 0, 4, 3, 6, 2]
-----
i= 6 k= 7 , k不等于i, 则交换A[i]和A[k]的值, A变为: [265, 264, 264, 258, 258, 244, 237, 220]
第i轮排序后Index变为: [7, 1, 5, 0, 4, 3, 2, 6]
-----

排序后的A为: [265, 264, 264, 258, 258, 244, 237, 220]
result= [[8, 265], [2, 264], [6, 264], [1, 258], [5, 258], [4, 244], [3, 237]]
前5名同学的学号和总分为:
8 265
2 264
6 264
1 258
5 258
>>>
```

实验任务 6-2 合唱排队

题目描述:

N 位同学站成一排，音乐老师要请其中的(N-K)位同学出列，使得剩下的 K 位同学排成合唱队形。

合唱队形是指这样的一种队形：设 K 位同学从左到右依次编号为 1, 2..., K, 他们的身高分别为 T1, T2, ..., TK, 则他们的身高满足 T1<...<Ti>Ti+1>...>TK(1<=i<=K)。

你的任务是，已知所有 N 位同学的身高，计算最少需要几位同学出列，可以使得剩下的同学排成合唱队形。

输入格式:

第一行是一个整数 N ($2 \leq N \leq 100$), 表示同学的总人数。第二行有 N 个整数, 用空格分隔, 第 i 个整数 T_i ($130 \leq T_i \leq 230$) 是第 i 位同学的身高(厘米)。

输出格式:

包括一行, 这一行只包含一个整数, 就是最少需要几位同学出列。

【输入样例】

8

186 186 150 200 160 130 197 220

【输出样例】

4

实验指导:

提示:

这是一道稍微有点变化的基础的动态规划问题。

在实验 5 的指导中, 给出了最长子序列的指导和状态转移方程。这道题与求最长子序列类似, 只不过它被分成了“两部分”。即中间最高的那个人把这个序列划分开了。

再结合一下动态规划的方式, 试着写出一个式子去描述这个问题。

举个例子, 什么是最长上升子序列? 就是给你一个序列, 请你在其中求出一段不断严格上升的部分, 它不一定要连续。

2, 3, 4, 7 和 2, 3, 4, 6 就是序列 2 5 3 4 1 7 6 的两种选取方案。最长的长度是 4。

设原序列为 a 。

当 $a_j < a_i$ ($j < i$) 且 $num_j + 1 > num_i$ 时, $num_i = num_j + 1$ 。最后的答案就是 num_n 。

对于每一个数, 都是在“可以接下去”的中, 从前面的最优值+1 转移而来。

实验任务 6-3 分桃子

题目描述:

孙悟空给花果山的小猴子们分桃子。首先, 他让每只小猴在左、右手上面分别写下一个整数, 悟空自己也在左、右手上各写一个整数。然后, 让这 n 只小猴排成一排, 悟空站在队伍的最前面。排好队后, 所有的小猴都会获得一些桃子, 每只小猴获得的桃子数分别是:

排在該小猴前面的所有小猴的左手上的數的乘積除以他自己右手上的數，然後向下取整得到的結果。

悟空不希望某一隻小猴獲得特別多的桃子，所以他想請你幫他重新安排一下隊伍的順序，使得獲得獎賞（即桃子）最多的小猴，所獲桃子儘可能的少。注意，悟空的位置始終在隊伍的最前面。

輸入格式：

第一行包含一個整數 n ，表示小猴數。

第二行包含兩個整數 a 和 b ，之間用一個空格隔開，分別表示悟空左手和右手上的整數。接下來 n 行，每行包含兩個整數 a 和 b ，之間用一個空格隔開，分別表示每個小猴左手和右手上的整數。

輸出格式：

包含一個整數，表示重新排列後的隊伍中獲獎賞最多的小猴所獲得的桃子數。

樣例輸入

```
3
1 1
2 3
7 4
4 6
```

樣例輸出

```
2
```

解釋：

按 1、2、3 號小猴這樣排列隊伍，獲得獎賞最多的小猴所獲得桃子數為 2；

按 1、3、2 這樣排列隊伍，獲得獎賞最多的小猴所獲得桃子數為 2；

按 2、1、3 這樣排列隊伍，獲得獎賞最多的小猴所獲得桃子數為 2；

按 2、3、1 這樣排列隊伍，獲得獎賞最多的小猴所獲得桃子數為 9；

按 3、1、2 這樣排列隊伍，獲得獎賞最多的小猴所獲得桃子數為 2；

按 3、2、1 這樣排列隊伍，獲得獎賞最多的小猴所獲得桃子數為 9。

因此，獎賞最多的小猴最少獲得 2 個桃子，答案輸出 2。

數據範圍

$1 \leq n \leq 1,000, 0 < a, b < 1000$

实验指导：

提示：相邻小猴交换对于前面的答案没影响，而且对于后面的答案也没有影响。也就是说相邻两小猴的位置交换只会对这两只小猴产生影响。先考虑两只小猴，再推广到整个队列，贪心的目标就出来了。