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-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. 可以选择定义一个类，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. 可以自己实现多个关键字的排序，也可以使用sorted函数和lambda匿名函数实现多个关键字的排序。在自己实现多个关键字排序时，可以使用冒泡排序等方法，注意两个元素交换的条件。

### 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<=N<=100)，表示同学的总数。第二行有N个整数，用空格分隔，第i个整数Ti(130<=Ti<=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。

输入输出样例

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。

在上次上机的指导中，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。

### 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：排在该小猴前面的所有小猴的左手上的数的乘积除以他自己右手上的数，然后向下取整得到的结果。

悟空不希望某一只小猴获得特别多的桃子，所以他想请你帮他重新安排一下队伍的顺序，

使得获得奖赏最多的小猴，所获桃子尽可能的少。注意，悟空的位置始终在队伍的最前面。

**输入格式：**

第一行包含一个整数 n，表示小猴数。   
第二行包含两个整数 a 和 b，之间用一个空格隔开，分别表示悟空左手和右手上的整数。 接下来 n 行，每行包含两个整数 a 和 b，之间用一个空格隔开，分别表示每个小猴左手和右手上的整数。

**输出格式：**

包含一个整数，表示重新排列后的队伍中获桃子最多的小猴所获得的桃子数。

**样例输入**

3

1 1

2 3

7 4

4 6

**样例输出**

**2**

**解释**

按1、2、3号小猴这样排列队伍，获得奖赏最多的小猴所获得桃子数为2；   
按1、3、2这样排列队伍，获得奖赏最多的小猴所获得桃子数为2；   
按2、1、3这样排列队伍，获得奖赏最多的小猴所获得桃子数为2；   
按2、3、1这样排列队伍，获得奖赏最多的小猴所获得桃子数为9；   
按3、1、2这样排列队伍，获得奖赏最多的小猴所获得桃子数为2；   
按3、2、1这样排列队伍，获得奖赏最多的小猴所获得桃子数为9。   
因此，奖赏最多的小猴最少获得 2 个桃子，答案输出 2。

**数据范围**

1 ≤ n ≤1,00，0 < 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。

附：

什么是最长公共子序列（LCS）?



给定序列

s1={1,3,4,5,6,7,7,8}

s2={3,5,7,4,8,6,7,8,2}

s1和s2的相同子序列，且该子序列的长度最长，即是LCS。s1和s2的其中一个最长公共子序列是 {3,4,6,7,8}

假如S1的最后一个元素 与 S2的最后一个元素相等，那么S1和S2的LCS就等于 {S1减去最后一个元素} 与 {S2减去最后一个元素} 的 LCS  再加上 S1和S2相等的最后一个元素。

假如S1的最后一个元素 与 S2的最后一个元素不等（本例子就是属于这种情况），那么S1和S2的LCS就等于 ： {S1减去最后一个元素} 与 S2 的LCS， {S2减去最后一个元素} 与 S1 的LCS 中的最大的那个序列。

求得了最长子序列，然后呢？

XAYBZA

AZBYAX

如上例，显然ABA是匹配的最长子序列，X、Y、Z是未匹配上的部分，注意一个字符串与自身颠倒决定的最长公共子序列一定是对称的，这里是ABA，剩下的就只需要给不对称元素寻找一个配对即可，让他们凑成对称的一对。上例依次添加X、Y、Z各一个，过程如下：

先考虑X怎么加，以ABA为对称轴：

X（AYBZA）

所以添加一个X：

X（AYBZA）X

再考虑Y：

（XA）Y | BZ | （AX）

添加一个Y：

（XA）Y | BZ | Y（AX）

最后考虑Z：

（XAY） | BZ | （YAX）

添加Z：

（XAY） | ZBZ | （YAX）

结论：需要添加的字符数=不匹配的字符数

5-3 LCS递归表达式：

