Python算法之旅（第6期）

上期回顾：

描述：已知元组t存储了若干个正整数，请编写函数返回各元素在元组中的排名。

排名规则：数值越大排名越靠前，数值相同则排名相同。

函数名：rank(t)

参数表：t -- 存储了正整数的元组。

返回值：返回一个列表，其元素值为相同下标的元素在元组中的排名。

示例：对于元组a=(3,2,2,4,3,5)，返回[3,5,5,2,3,1]

算法分析：

排名原理：某个元素的名次等于元组中比它大的元素的数量。

首先初始化每个元素的排名均为1，然后扫描元组，依次处理每一个元素；将每个元素t[i]依次与其他元素t[j]比较，若t[i] < t[j]，则第i个元素名次加1。

参考代码如下：

def rank\_1(t): #算法1：每次只处理一个元素，简单直观

a = [1] \* len(t)

for i in range(0, len(t)):

for j in range(0, len(t)):

if t[i] < t[j]:

a[i] += 1

return a

def rank\_2(t): #算法2：同时处理a[i]和a[j]，效率较高

a = [1] \* len(t)

for i in range(0, len(t)):

for j in range(i+1, len(t)):

if t[i] < t[j]:

a[i] += 1

elif t[i] > t[j]:

a[j] += 1

return a

根据成绩计算名次是一个很实用的功能，也有多种实现的方法，2017年4月浙江省高考技术卷中就出来这样一道题目。

题目：根据成绩算名次(2017年4月浙江省普通高校招生选考科目考试17题)

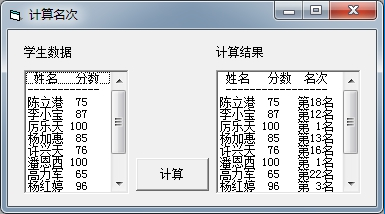
难度：3星 有趣：3星 有用：4星

分类：排序

描述：小王编写了一个依据成绩计算名次的VB程序，成绩为0到100之间的整数。算法的基本思想：先统计每个分数的个数，然后按照分数从高到低依次计算每个有效分数（该分数的个数不为0）对应的名次，分数相同时名次并列。最高分为第1名，该分数的名次与个数之和为下一个有效分数的名次，以此类推。程序用数组A存放每个分数对应的个数，数组B存放每个分数对应的名次。例如，下表中最高分100有2个，并列第1名，则分数96的名次为分数100的名次加上分数100的个数，即第3名。

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 分数 | 100 | 99 | 98 | 97 | 96 | 95 | 94 | … | 0 |
| 个数（A数组） | 2 | 0 | 0 | 0 | 1 | 0 | 3 | … | 0 |
| 名次（B数组） | 1 |  |  |  | 3 |  | 4 | … |  |

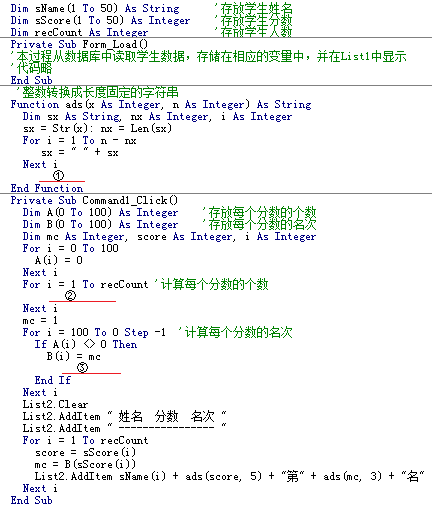
程序运行时，学生数据显示在列表框List1中，单击“计算”按钮Command1，计算结果显示在列表框List2中，程序运行界面如图所示。



实现上述功能的VB程序如下，请回答下列问题：

（1）如表所示，若分数93的个数为2，则该分数对应的名次为\_\_\_\_\_\_\_\_\_\_\_\_\_。

（2）请在划线处填入合适的代码。



答案 (1)7

(2)①ads=sx ②A(sScore(i))=A(sScore(i))+1 ③mc=A(i)+B(i) 或 mc=mc+A(i)

算法分析：

高中VB算法题的一个重要特征就是题目会把算法思路完整地呈现给考生，考生只要根据算法分析去理解代码就行了。

首先来理解数组A的含义，数组A用来存储每个分数的个数，则其每个元素的下标和值分别对应某个分数和该分数出现的次数，例如，若A(60)=5，则代表有5个学生考了60分；若A(70)=4，则代表有4个学生考了70分。

因此计算每个分数的个数，就是累计数组A中下标为sScore(i)的元素值，这是经典的桶排序思想。

在计算每个分数对应的名次时，是先规定最高分为第1名，即初始化mc = 1；然后从高到低遍历每一个分数，设置该分数的名次B(i) = mc；接着更新下一个有效分数的名次mc=A(i)+B(i)。如此循环处理每一个有效分数。

本算法的巧妙之处在于它不是直接存储每个学生的名次，而是存储每个分数对应的名次，最后根据学生成绩输出其名次。这样只需一重循环就可以完成计算，大大提高了效率。

下面我们用Python语言来完成这道题目。

描述：根据成绩算名次。已知元组t存储了若干个学生的成绩，成绩为0到100之间的整数。

请编写函数返回各学生的名次，其中最高分为第1名，成绩相同则名次也相同。

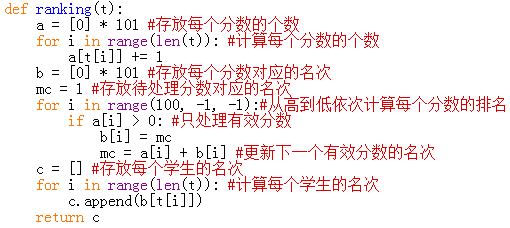
要求时间复杂度为O(n)，也可以理解成只能使用一重循环。

函数名：ranking(t)

参数表：t -- 存储了学生的成绩的元组。

返回值：返回一个列表，其元素值为相同下标的元素在元组中的排名。

示例：对于元组t = (3,2,2,4,3,5)，返回[3,5,5,2,3,1]



课后思考：

本题在计算每个分数的个数时，采用了经典的桶排序思想，那么什么是桶排序呢？

聪明的你可以先去查查相关资料，我们在下一篇文章中再和大家详谈。

如果觉得自己已经理解桶排序思想了，可以试试下面这道题目。

描述：找出字符串中第一个只出现一次的字符

函数名：first\_one(s)

参数表：s -- 被访问的字符串。

返回值：返回第一个只出现一次的字符，如果不存在返回"No"。

示例: s="asdfasdfo"，返回"o"。

另外，如果你有更 Pythonic（优雅的、地道的、整洁的）代码，或者与本文不同的算法思路和代码实现，请你一定留言或联系我，让我们一起讨论，共同进步。