

# 商管程式設計 (106-1)

## 作業三

作業設計：孔令傑  
國立臺灣大學資訊管理學系

繳交作業時，請至 PDOGS (<http://pdogs.ntu.im/judge/>) 為第一、二題上傳一個 PDF 檔，再為第三題與第四題各上傳一份 Python 3.6 原始碼 (以複製貼上原始碼的方式上傳)。第四題是 bonus 加分題。每位學生都要上傳自己寫的解答。不接受紙本繳交；不接受遲交。請以英文或中文作答。

這份作業的截止時間是 **2017 年 10 月 9 日凌晨一點**。在你開始前，請閱讀課本的第五、七、十章<sup>1</sup>。為這份作業設計測試資料並且提供解答的助教是蔡依庭。

### 第一題

(20 分) 請回答下列問題：

(a) (10 分) 對於下方程式

```
lst = [1, 5, 10]

for i in range(len(lst)):
    print(i, end = " ")
print()
for i in range(len(lst)):
    print(lst[i], end = " ")
print()

for i in lst:
    print(i, end = " ")
print()
for i in lst:
    print(lst[i], end = " ")
```

請解釋為什麼輸出的前三行會長那個樣子，以及第四行輸出到一半時會 run time error。

(b) 對於下方程式

```
tri = [[1, 2, 3], [1, 2], [1]]

for i in range(len(tri)):
    for j in tri[i]:
        print(j, end = " ")
        #print(tri[i][j])
    print()
```

<sup>1</sup>課本是 A. Downey 所著的 *Think Python 2*，在 <http://greenteapress.com/wp/think-python-2e/> 可以下載。

請說明為什麼執行這個程式可以完整地印出 `tri` 裡面的資料，但註解掉第五行、反註解第六行之後，卻會在印出一部份資料之後 `run time error`。

## 第二題

(20 分) 在第四講的課程投影片第 21 頁中，有如下程式碼：

```
# read and prepare n, m, and p
n = int(input("Number of jobs: "))
m = int(input("Number of machines: "))
pStr = input("Processing times: ")

p = pStr.split(' ')
for i in range(n):
    p[i] = int(p[i])
```

這是用來做我們上課教的 longest processing time 演算法的前置資料處理。在這段程式碼中，`processing time` 是由使用者輸入的。

現在請改一改這段程式碼，來應付下方各題的不同輸入格式。你的程式執行完之後，整數 `n`、整數 `m` 和清單 `p` 應該要跟上面程式的內容一樣。

- (a) (10 分) 使用者會先輸入工作個數，接著輸入一個分號「;」，接著輸入機臺個數，接著輸入一個分號「;」，接著輸入  $n$  個 `processing time` 的值，而這些 `processing time` 兩兩之間用一個空白字元隔開。舉例來說，在四臺機臺上處理 `processing time` 為 1、7、3、9、2、6、4、8 的八個工作時，輸入會是

```
8;4;1 7 3 9 2 6 4 8
```

**提示：**這題有很多作法，最簡單的作法可能是呼叫 `split()` 兩次。

- (b) (10 分) 使用者會在第一行輸入工作個數、第二行輸入機臺個數、第三行輸入第一個 `processing time`、第四行輸入第二個 `processing time`，直到第  $n + 2$  行輸入第  $n$  個 `processing time`。舉例來說，在四臺機臺上處理 `processing time` 為 1、7、3、9、2 的五個工作時，輸入會是

```
5
4
1
7
3
9
2
```

**提示：**這題一樣有很多作法，最簡單的作法中應該會有一個迴圈裡面裝著 `input()`。

### 第三題

(60 分) 兩年前的暑假，我跟臺灣某家知名（我相信每個同學都聽過）的 3C 產品零售商，一起討論了他們公司的需求預測和訂貨政策。他們公司在某一類型的商品（或服務）使用的需求預測規則大致如下。給定過去六週的真實需求量  $x_1$ 、 $x_2$  到  $x_6$ ，先計算出「需求趨勢係數」

$$r = \frac{x_4 + x_5 + x_6}{x_1 + x_2 + x_3}。$$

接著將趨勢係數代入一張預先設定的表，去查出「需求調整倍率」 $f(r)$ 。一個需求調整倍率的範例如表 1 所示，例如  $f(0.5) = 0.9$  和  $f(3) = 1.1$ 。最後，下週的需求就預測為

$$y = \left[ \frac{x_3 + x_4 + x_5 + x_6}{4} \right] f(r)$$

的無條件進入  $[y]$ 。舉例來說，假設過去六週的需求是 94、93、98、118、113 和 140，則  $r = \frac{118+113+140}{94+93+98} \approx 1.302$ 、 $f(r) = 1.1$ 、 $y = \frac{98+118+113+140}{4} \times 1.1 \approx 128.975$ ，最後的預測值即為  $[y] = 129$ 。

| $r$    | (0, 0.4] | (0.4, 0.7] | (0.7, 1] | (1, 3] | (3, $\infty$ ) |
|--------|----------|------------|----------|--------|----------------|
| $f(r)$ | 0.8      | 0.9        | 1        | 1.1    | 1.2            |

表 1: 需求調整倍率的範例

在本題中，你將被給定過去六週的需求，以及一組需求調整倍率的公式（由九個數字組成），請計算出下週的需求預測值。

#### 輸入輸出格式

系統會提供一共 20 組測試資料，每組測試資料裝在一個檔案裡。在每個檔案中會有兩行，第一行包含六個整數  $x_1$ 、 $x_2$  到  $x_6$ ，第二行包含代表需求調整公式的九個數字，其中前四個數字由小而大排列，代表趨勢係數  $r$  的四個切分點，而後五個數字也由小而大排列，代表  $f(r)$  的五個可能的數值。任意兩個整數之間被一個空白隔開。已知  $0 \leq x_i \leq 10000$ ，而第二行的每一個數字都介於 0 到 5 之間（包含 0 和 5）。

讀入資料後，請按照題目指定的需求預測辦法，去印出下週的需求預測值（一定是整數）。舉例來說，如果輸入是

```
94 93 98 118 113 140
0.4 0.7 1 3 0.8 0.9 1 1.1 1.2
```

則輸出應該是

```
129
```

如果輸入是

```
118 113 140 94 93 98
0.5 1 1.5 2 0.7 0.9 1 1.1 1.3
```

則輸出應該是

## 你上傳的原始碼裡應該包含什麼

你的.py 原始碼檔案裡面應該包含讀取測試資料、做運算，以及輸出答案的 Python 3.6 程式碼。當然，你應該寫適當的註解。針對這個題目，你**不可以**使用上課沒有教過的方法。

## 評分原則

- 這一題的其中 40 分會根據程式運算的正確性給分。PDOGS 會直譯並執行你的程式、輸入測試資料，並檢查輸出的答案的正確性。一筆測試資料佔 2 分。
- 這一題的其中 20 分會根據你所寫的程式的品質來給分。助教會打開你的程式碼並檢閱你的程式的運算邏輯、可讀性，以及可擴充性。請寫一個「好」的程式吧！

## 第四題 (bonus)

(20 分) 你是一家零售業者的作業長 (chief operations officer, COO, 厲害吧)，現在正想要解決一個補貨的問題。在一個區域裡，你有  $m$  個零售店，都銷售同一種商品，其中零售店  $i$  落在  $(x_i, y_i)$  位置上 (單位為公里)。你另外有  $n$  個物流中心，其中物流中心  $j$  落在  $(u_j, v_j)$  位置上 (單位為公里)。由於這個區域的路都是東西向或南北向的，因此點和點之間的距離是用曼哈頓距離 (Manhattan distance) 計算，而非歐幾里得距離 (Euclidean distance)。更精確地說，物流中心  $j$  到零售店  $i$  的距離是

$$d_{ij} = |u_j - x_i| + |v_j - y_i| \text{ 公里。}$$

每家零售店的商品售價都一樣是  $p$ ，但他們的需求不一樣。我們將零售店  $i$  的需求量稱為  $D_i$ ，表示在這家店每天最多能賣掉  $D_i$  個商品。要賣東西有個前提，就是前一天晚上要補貨。對於任意一家零售店，你必須決定由哪個 (或哪幾個) 物流中心來補貨給它；如果你指定了複數個物流中心給它，還必須指定個別的補貨數量。補貨成本和補貨數量與補貨距離的乘積成正比，每個商品每公里的單位補貨成本是  $c$  元。更精確地說，若物流中心  $j$  每晚送  $x_{ij}$  個商品到零售店  $i$ ，則其補貨成本為

$$c_{ij}(x_{ij}) = cd_{ij}x_{ij} \text{ 元。}$$

對於這些商品，在零售店  $i$  的銷貨毛利就是  $(p - cd_{ij})x_{ij}$  元了。當然，如果一家店只能賣 100 個商品，而你補 1000 個給它，最後也只能賣出 100 個，剩下 900 個都是浪費錢而已。

給定上述資訊，你的任務是找出能夠最大化總銷貨毛利的補貨方式；更精確地說，你必須決定上述的  $x_{ij}$ ，來求解

$$\begin{aligned} \max \quad & \sum_{i=1}^m \sum_{j=1}^n (p - cd_{ij})x_{ij} \\ \text{s.t.} \quad & \sum_{j=1}^n x_{ij} \leq D_i \quad \forall i = 1, \dots, m \\ & x_{ij} \geq 0 \quad \forall i = 1, \dots, m, j = 1, \dots, n. \end{aligned}$$

請注意兩件事情。首先，如果有一家店的補貨成本恰好等於商品售價，那麼請你對它進行補貨以滿足其所有需求。其次，請注意你不一定要滿足所有需求。舉例來說，要是有些店被蓋在非常偏遠的地方，導致補貨成本大於售價，那最好的作法就是不要補貨（這家店就去賣其他東西就好）。換句話說，在很詭異的情況下，搞不好最佳方案是不要補貨給任何一家店。

### 輸入輸出格式

系統會提供許多筆測試資料，每筆測試資料裝在一個檔案裡。在每個檔案中，第一列存放四個整數  $n$ 、 $m$ 、 $p$  和  $c$ 。第二列存放  $2n$  個整數  $u_1$ 、 $v_1$ 、 $u_2$ 、 $v_2$  直到  $u_n$ 、 $v_n$ ，表示第  $i$  個物流中心在位置  $(u_i, v_i)$  上。第三列存放  $2m$  個整數  $x_1$ 、 $y_1$ 、 $x_2$ 、 $y_2$  直到  $x_m$ 、 $y_m$ ，表示第  $i$  個零售店在位置  $(x_i, y_i)$  上。第四列存放  $m$  個整數  $D_1$ 、 $D_2$  直到  $D_m$ ，表示第  $i$  個零售店的需求量是  $D_i$ 。每一列中的任兩個數字之間都用一個空白鍵隔開。已知  $1 \leq n \leq 10$ 、 $1 \leq m \leq 1000$ 、 $1 \leq p \leq 500$ 、 $1 \leq c \leq 10$ 、 $0 \leq x_i \leq 200$ 、 $0 \leq y_i \leq 200$ 、 $0 \leq u_i \leq 200$ 、 $0 \leq v_i \leq 200$ ，以及  $1 \leq D_i \leq 100$ 。

請根據题目的描述，找出能最大化總銷貨毛利的補貨計畫，並且印出兩個整數，分別代表在最佳補貨計畫下能賺到的總銷貨毛利以及未被滿足的需求總量。舉例來說，如果輸入是

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

則輸出應該是

```
58 7
```

其中總銷貨毛利是  $3(5 - 2) + 4(5 - 1) + 5(5 - 2) + 6(5 - 2) = 58$ ，在零售店 1 到 4 賺得的；未滿足需求總量是 7，零售店 5 的需求量總和。請注意零售店 6 的銷貨毛利剛好是 0（從最近的物流中心進行補貨的成本恰好等於售價），所以它對總銷貨毛利沒有貢獻，但是它的需求也被視為已滿足。

### 你上傳的原始碼裡應該包含什麼

針對這個題目，你可以使用任何方法。

### 評分原則

這一題的 20 分都根據程式運算的正確性給分，一筆測試資料佔 2 分。