

# 商管程式設計 (106-1)

## 作業四

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、二題各上傳一份 Python 3.6 原始碼 (以複製貼上原始碼的方式上傳)。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 16 日凌晨一點**。為這份作業設計測試資料並且提供解答的助教是薛光佑。

### 第一題

(60 分) 在共享經濟的時代，有很多雙邊平臺 (two-sided platform) 不自己提供商品或服務，而是媒合供需雙方。Uber、Airbnb 是其中幾個特別廣為人知的例子：Uber 媒合需要搭車的乘客和會開車的人，Airbnb 則媒合需要找地方住的旅客和有空房間的人。要讓供需雙方願意在平臺上交易，一個雙邊平臺通常要能降低供需雙方之間的資訊不對稱，也就是協助一方看到另一方的品質。為此，常見的作法是在每一次交易後，讓買賣雙方各為對方打個分數。在本題中，我們將探討這樣的互評系統。

我們假設本系統中一共有  $m$  個賣方和  $n$  個買方，而我們用一個  $m \times n$  的矩陣  $B$  儲存買方給賣方打的分數。在每一次評分中，一個買方可以給一個賣方 1 至 5 分的其中一個分數。如果買方  $j$  給賣方  $i$  的分數是  $s \in \{1, 2, 3, 4, 5\}$ ，我們就在矩陣  $B$  中記錄  $B_{ij} = s$  (在第  $i$  列第  $j$  行存  $s$  這個數字)。如果沒有評分，我們就存 0。我們假設沒有任何一對買賣雙方有交易過兩次以上。

給定矩陣  $B$  之後，現在我們考慮一位剛進入平臺要找賣方的買方。在這位買方下搜尋條件後，系統會找出符合其搜尋條件的賣方 (例如夠近且在路上開著空車的司機、在旅遊目的地且該時段有空房的房東)，並回傳「綜合評價最高」的前三位賣方。對每一位賣家  $i$ ，令  $c_i$  為他被評價的次數，則我們定義他的綜合評價分數  $R_i$  為

$$R_i = \sum_{j=1}^n B_{ij} - kc_i,$$

亦即他得到的總分減去其被評價次數的  $k$  倍 (所以只有在一次服務得到超過  $k$  分，才會讓綜合評價上升；如果得到低於  $k$  分，綜合評價只會變低)。為了簡單起見，我們假設所有賣方都符合搜尋條件。

在本題中，你將被給定矩陣  $B$  與綜合評價計算係數  $k$ ，請回傳綜合評價分數最高的前三位賣方的編號，先印分數最高的，再印次高的，最後印第三高的。如果兩個賣方綜合評價分數相同，則把編號小的排在前面。舉例來說，假設平臺上一共有四位賣方、六位買方，過往評價記錄為

$$B = \begin{bmatrix} 0 & 0 & 5 & 3 & 0 & 0 \\ 0 & 3 & 1 & 0 & 1 & 5 \\ 4 & 0 & 0 & 0 & 0 & 5 \\ 5 & 0 & 2 & 4 & 0 & 5 \end{bmatrix},$$

且  $k = 4$ ，則四位賣方的綜合評價分數依序為  $R_1 = 8 - 8 = 0$ 、 $R_2 = 10 - 16 = -6$ 、 $R_3 = 9 - 8 = 1$  和  $R_4 = 16 - 16 = 0$ 。如果要依序回傳綜合評價前三高的賣方，我們會回傳 3、1、4。請注意因為賣方 1 和賣方 4 的綜合評價分數一樣，我們會把編號小的賣方 1 放在前面。

## 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  $m + 1$  行，第一行包含三個整數  $m$ 、 $n$  和  $k$ ，第二行起的第  $i + 1$  行包含  $B_{i,1}$ 、 $B_{i,2}$  到  $B_{i,n}$  等  $n$  個整數。第一行的兩個整數之間被一個分號隔開，其他行的兩個整數之間被一個逗點隔開。已知  $3 \leq m \leq 1000$ 、 $3 \leq n \leq 1000$ 、 $k \in \{2, 3, 4\}$ 、 $B_{ij} \in \{0, 1, 2, 3, 4, 5\}$ 。

讀入資料後，請按照題目指定的規則，印出三位賣方的編號。任兩個編號之間用一個空白字元隔開。舉例來說，如果輸入是

```
4;6;4
0,0,5,3,0,0
0,3,1,0,1,5
4,0,0,0,0,5
5,0,2,4,0,5
```

則輸出應該是

```
3 1 4
```

如果輸入是

```
5;6;4
0,0,5,3,0,0
0,3,1,0,1,5
4,0,0,0,0,5
5,0,2,4,0,5
5,5,5,5,5,5
```

則輸出應該是

```
5 3 1
```

## 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！

## 第二題

(40 分) 在一個城市裡，有一個銷售員，他要拜訪此城市裡的  $n - 1$  個客戶。為了簡單起見，讓我們稱呼銷售員的公司所在地為客戶 1，而這  $n - 1$  個客戶則依序為客戶 2、3 直到  $n$ 。客戶  $i$  和客戶  $j$  之間的最短距離已知為  $D_{ij}$ 。我們假設從客戶  $i$  到客戶  $j$  的距離和從客戶  $j$  到客戶  $i$  一樣，因此如果我們收集  $D_{ij}$  以組成一個  $n \times n$  的方形矩陣  $D$ ，則  $D$  是對稱的。舉例來說，假設共有四個客戶，且

$$D = \begin{bmatrix} 0 & 4 & 6 & 3 \\ 4 & 0 & 8 & 5 \\ 6 & 8 & 0 & 2 \\ 3 & 5 & 2 & 0 \end{bmatrix},$$

就表示客戶 1 跟 2 之間的距離是 4、客戶 2 跟 3 之間的距離是 8，依此類推。在本題中，我們假設三點之間的距離不一定符合三角不等式，也就是  $D_{ij} + D_{jk} < D_{ik}$  是有可能的。

銷售員的任務是從客戶 1 的位置拜訪每個客戶恰好一次，然後回到客戶 1 的位置，而總距離愈短愈好。顯然如果客戶數量不多，那找出最短的走法並不困難，但如果客戶數量很多，這個問題就相當困難了。因此，銷售員決定使用一個簡單的演算法：每一次要決定下一個目的地時，都去拜訪目前沒拜訪過的客戶中距離現在所在地最近的那一位；如果有複數個選擇，就選客戶編號最小的那個。以上面的  $D$  為例，從客戶 1 出發，銷售員會先拜訪客戶 4（因為  $3 < 4 < 6$ ），接著客戶 3（因為  $2 < 5$ ），接著客戶 2（別無選擇），然後回到客戶 1。總移動距離是  $3 + 2 + 8 + 4 = 17$ ，在這個例子裡是最短的了<sup>1</sup>。

在本題中，你將被給定客戶數  $n$ （包含客戶 1）以及兩兩客戶間的距離。請使用本題指定的方法，依序印出銷售員拜訪的客戶編號（從客戶 1 開始），並在最後印出總移動距離。

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  $n + 1$  行，第一行包含一個整數  $n$ ，第二行至第  $n + 1$  行的第  $i + 1$  行包含  $D_{i,0}$ 、 $D_{i,1}$  到  $D_{i,n}$  等  $n$  個整數。同一行的任意兩個整數之間被一個空白隔開。已知  $2 \leq n \leq 1000$ 、 $1 \leq D_{ij} \leq 10000$ 、 $D_{ii} = 0$ 。

讀入資料後，請按照題目指定的規則，依序印出銷售員拜訪的客戶編號（從客戶 1 開始），並在最後印出總移動距離。任兩個編號之間用一個空白字元隔開。舉例來說，如果輸入是

```
4
0 4 6 3
4 0 8 5
6 8 0 2
3 5 2 0
```

則輸出應該是

```
1 4 3 2 17
```

針對這個題目，你**可以**使用任何方法。這一題的 40 分都根據程式運算的正確性給分，一筆測試資料佔 2 分。

<sup>1</sup>這很顯然地未必能找到最短的走法，但直觀上看起來總是有点道理。這種「不一定最佳，但通常還不錯」的方法被稱為啟發性演算法（heuristic algorithm），在後續課程中我們會再多做介紹。