

# 商管程式設計 (110-1)

## 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>) 為第一、二、三、四題各上傳一份 Python 3.9 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。

這份作業的截止時間是 **2021 年 11 月 6 日晚上九點**。在你開始前，請閱讀課本的第十章<sup>1</sup>。為這份作業設計測試資料並且提供解答的助教是彭晨。

### 第一題

(20 分) 炭治郎在一個有  $n$  種面額銅板的國家賣自家製作的炭，銅板面額由小到大依序是  $p_1, p_2$  直到  $p_n$ 。某日，炭治郎辛苦一天之後，發現自己收錢的盒子裡有  $p_i$  元銅板  $x_i$  枚， $i = 1, \dots, n$ 。他帶著這些錢，考慮買些東西回家。炭治郎來到糧食行，依序考慮裡面的  $m$  種食材，其中第  $j$  種食材的價格是  $q_j$  元。從  $j = 1$  開始，如果他還夠錢買第  $j$  種食材，他就會買，否則就不買，然後就考慮第  $j + 1$  種食材，直到  $m$  種食材都考慮完為止。買完他就回家。

在本題中，請計算炭治郎回家時身上剩多少錢。舉例來說，如果  $n = 3, p_1 = 1, p_2 = 5, p_3 = 10, x_1 = 18, x_2 = 6, x_3 = 8$ ，則炭治郎共賺進  $10 \times 8 + 5 \times 6 + 18 = 128$  元。若  $m = 4, q_1 = 50, q_2 = 30, q_3 = 70, q_4 = 10$ ，則炭治郎將購買食材 1、2、4，花掉 90 元，回家時身上剩 38 元。

### 輸入輸出格式

系統會提供一共數組測試資料，每組測試資料裝在一個檔案裡。在每個檔案中會有四行，第一行有兩個正整數  $n, m$ ，第二行有  $n$  個正整數  $p_1, p_2$  直到  $p_n$ ，第三行有  $n$  個非負整數  $x_1, x_2$  直到  $x_n$ ，第四行有  $m$  個正整數  $q_1, q_2$  直到  $q_m$ ，一行中任兩個相鄰的值之間被一個逗點隔開。已知  $1 \leq n \leq 10, 1 \leq m \leq 10, 1 \leq p_1 < p_2 < \dots < p_n \leq 1000, 0 \leq x_i \leq 100, 1 \leq q_i \leq 10000$ 。

請依題目所述，輸出一個整數代表炭治郎到家後身上的錢的金額。舉例來說，如果輸入是

```
3,4
1,5,10
18,6,8
50,30,70,10
```

則輸出應該是

```
38
```

如果輸入是

<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。

```
3,4
1,5,10
18,6,8
70,50,30,10
```

則輸出應該是

```
8
```

如果輸入是

```
5,8
1,2,3,50,100
1,2,3,4,5
100,100,100,100,100,100,100,100,100
```

則輸出應該是

```
14
```

## 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 3.9 程式碼。當然，你應該寫適當的註解。針對這個題目，你**可以**使用上課沒有教過的方法。

## 評分原則

這一題的所有分數都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。

## 第二題

(30 分) 經調查，某區域由於發生火災太頻繁，因此政府決定在該區域挑選一個城鎮蓋一座消防局。為了找出最佳位址，我們必須根據各城鎮的預期失火次數和彼此間的距離，最小化一整年消防車的預期移動總路徑長度。對於每個城鎮  $i$ ，我們根據以往數據統計分析，預期它一年會發生火災的次數為  $h_i$ ，而城鎮  $i$  和城鎮  $j$  間的距離我們記為  $d_{ij} = d_{ji}$  公里（自己和自己的距離則為 0，即  $d_{ii} = 0$ ），任三城鎮間的距離皆符合三角不等式。給定城鎮的資訊後，請找出最適合蓋消防局的城鎮，以及蓋在那邊的整年預期移動總路徑長度。

舉例來說，若題目給定  $h_1 = 7$ 、 $h_2 = 4$ 、 $h_3 = 9$ ， $d_{12} = d_{21} = 3$ 、 $d_{23} = d_{32} = 6$ 、 $d_{13} = d_{31} = 5$ ，則我們可分別計算消防局蓋在城鎮 1、2、3 的總預期移動路徑長，分別為：

- $0 \times 7 + 3 \times 4 + 5 \times 9 = 57$  公里。

- $0 \times 4 + 3 \times 7 + 6 \times 9 = 75$  公里。
- $0 \times 9 + 5 \times 7 + 6 \times 4 = 59$  公里。

則取其中總路徑最短的城鎮 1 為最佳選址位置，最後輸出城鎮編號 1 和總路徑長度 57，兩者以一個逗號隔開；若平手，則選擇較小的城鎮編號。

## 輸入輸出格式

系統會提供一共數組測試資料，每組測試資料裝在一個檔案裡。在每個檔案中，會有  $n + 2$  行。第一行為一個正整數  $n$ ，代表城鎮數量，其中  $1 \leq n \leq 20$ ；第二行為  $n$  個整數  $h_1, h_2$  直到  $h_n$ ，兩兩以一個逗號隔開，分別代表城鎮  $i$  的預期失火次數，其中  $0 \leq h_i \leq 20$ 。第三行到  $n + 2$  行，每行有  $n$  個整數  $d_{i,1}, d_{i,2}$  直到  $d_{i,n}$ ，兩兩以一個逗號隔開，代表城鎮  $i$  到各城鎮之距離，其中  $0 \leq d_{ij} \leq 1000$ ， $d_{ii} = 0$  且距離間滿足三角不等式。請依題目指示找出最佳選址位置，並輸出城鎮編號  $i$  和總路徑長度，兩者以一個逗號隔開。若平手，則選擇城鎮編號較小的城鎮。

舉例來說，如果輸入是

```
3
7,4,9
0,3,5
3,0,6
5,6,0
```

則輸出應該是

```
1,57
```

如果輸入是

```
4
4,10,2,8
0,2,2,9
2,0,1,2
2,1,0,1
9,2,1,0
```

則輸出應該是

```
2,26
```

如果輸入是

```
1
3
0
```

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 3.9 程式碼。當然，你應該寫適當的註解。針對這個題目，你**可以**使用上課沒有教過的方法。

## 評分原則

這一題的所有分數都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。

## 第三題

(50 分) 承上題，這次我們想要多蓋幾座消防局，因此題目將多給定一個值  $k$ ，表示我們要蓋  $k$  座消防局。我們依舊希望能最小化總預期路徑長，因此在每次加蓋一座消防局時，皆先計算剩餘城鎮和現存消防局（有可能為多座）間的距離，**取其中最短路徑長為此城鎮與現存消防局的「距離」，並選取其中「距離」最遠的城鎮為下一座消防局的選址位置**<sup>2</sup>。在本題中，要請你使用本演算法去挑出  $k$  個消防局位置，並計算你的建設計畫所帶來的預期總路徑長。底下我們用兩個例子說明。

假設  $n = 3$ 、 $k = 2$ ，題目給定  $h_1 = 7$ 、 $h_2 = 4$ 、 $h_3 = 9$ ， $d_{12} = d_{21} = 3$ 、 $d_{13} = d_{31} = 5$ 、 $d_{23} = d_{32} = 6$ ，我們依舊分別計算消防局蓋在城鎮 1、2、3 的總預期移動路徑長，分別為：

- $0 \times 7 + 3 \times 4 + 5 \times 9 = 57$  公里。
- $0 \times 4 + 3 \times 7 + 6 \times 9 = 75$  公里。
- $0 \times 9 + 5 \times 7 + 6 \times 4 = 59$  公里。

其中總路徑最短的城鎮 1 為第一座消防局的最佳選址位置，我們接著計算剩餘城鎮 2、3 與現存消防局所在位置（城鎮 1）的距離，取其中距離最長的為下一座消防局的選址。以本題來說， $d_{12} = d_{21} = 3 < d_{13} = d_{31} = 5$ ，則我們選擇城鎮 3 為第二座消防局選址。此時三座城鎮已知城鎮 1、3 皆為消防局選址，因此只須將城鎮 2 分配給較近的那一座消防局，即可算出預期總路徑。以本例來說，即為城鎮 1，可得預期總路徑  $3 \times 4 = 12$  公里。換言之，在本例中，本演算法會挑選出的  $k = 2$  個消防局位置是在城鎮 1、3，其所帶來的預期總路徑長是 12 公里。

這邊有一個超過三座城鎮的例子。假設  $n = 4$ 、 $k = 3$ ，且  $h_1 = 7$ 、 $h_2 = 4$ 、 $h_3 = 9$ 、 $h_4 = 2$ ， $d_{12} = d_{21} = 3$ 、 $d_{13} = d_{31} = 5$ 、 $d_{14} = d_{41} = 6$ 、 $d_{23} = d_{32} = 4$ 、 $d_{24} = d_{42} = 3$ 、 $d_{34} = d_{43} = 4$ ，則第一座消防局蓋在城鎮 1、2、3、4 的總預期移動路徑長計算如下：

- $0 \times 7 + 3 \times 4 + 5 \times 9 + 6 \times 2 = 69$  公里。

<sup>2</sup>我們也可以使用數學符號定義「距離」。假設  $S$  代表所有現存消防局位址之集合， $d_{ij}$  代表城鎮  $i$  和城鎮  $j$  之間的距離，則我們定義任意城鎮「與現存消防局的距離」為  $d_i(S) = \min_{j \in S} \{d_{ij}\}$ 。

- $0 \times 4 + 3 \times 7 + 4 \times 9 + 3 \times 2 = 63$  公里。
- $0 \times 9 + 5 \times 7 + 4 \times 4 + 4 \times 2 = 59$  公里。
- $0 \times 2 + 6 \times 7 + 3 \times 4 + 4 \times 9 = 90$  公里。

其中總路徑最短的城鎮為 3，因此為第一座消防局的最佳選址位置。

要挑第二座消防局的位置時，我們接著計算剩餘城鎮 1、2、4 與現存消防局所在位置（城鎮 3）的距離，取其中距離最長的為下一座消防局的選址。以本題來說， $d_{13} = d_{31} = 5$ 、 $d_{23} = d_{32} = 4$ 、 $d_{34} = d_{43} = 4$ ，與城鎮 3 距離最遠的為城鎮 1，則我們選擇城鎮 1 為第二座消防局選址。

此時已有兩座消防局，若要挑第三座消防局的位置時，我們必須計算剩餘城鎮與現存兩座消防局分別的距離，取其中較短的路徑長為「距離」去做比較，計算如下：

- 城鎮 2： $\min\{d_{12} = d_{21} = 3, d_{23} = d_{32} = 4\} = 3$  公里。
- 城鎮 4： $\min\{d_{34} = d_{43} = 4, d_{14} = d_{41} = 6\} = 4$  公里。

因城鎮 4 與現存消防局距離較遠，因此為第三座消防局最佳選取位置。

已知城鎮 1、3、4 皆為消防局設址，因此我們分配城鎮 2 給距離最近的城鎮 1 的消防局（亦即由城鎮 1 的消防局負責救城鎮 2 的火，以最小化總預期路徑），算出預期總路徑長為  $3 \times 4 = 12$  公里。

## 輸入輸出格式

系統會提供一共數組測試資料，每組測試資料裝在一個檔案裡。在每個檔案中，會有  $n + 2$  行。第一行為兩個正整數  $n$ 、 $k$ ，分別代表城鎮數量和預計要蓋的消防局個數，兩者以一個逗號隔開，其中  $1 \leq n \leq 20$ 、 $1 \leq k \leq n$ ；第二行為  $n$  個整數  $h_1$ 、 $h_2$  直到  $h_n$ ，兩兩以一個逗號隔開，分別代表城鎮  $i$  的預期失火次數，其中  $0 \leq h_i \leq 20$ 。第三行到  $n + 2$  行，每行有  $n$  個整數  $d_{i,1}$ 、 $d_{i,2}$  直到  $d_{i,n}$ ，兩兩以一個逗號隔開，代表城鎮  $i$  到各城鎮之距離，其中  $0 \leq d_{ij} \leq 1000$ ， $d_{ii} = 0$  且距離間滿足三角不等式。請依題目指示找出  $k$  個最佳選址位置，並輸出一串城鎮編號  $i$  和總路徑長度，兩編號間以一個逗點隔開，編號和總路徑長以一個分號隔開。注意：每次選取城鎮時若平手，則選擇城鎮編號較小的城鎮。

舉例來說，如果輸入是

```
3,2
7,4,9
0,3,5
3,0,6
5,6,0
```

則輸出應該是

```
1,3;12
```

如果輸入是

```
4,3
7,4,9,2
```

```
0,3,5,6
3,0,4,3
5,4,0,4
6,3,4,0
```

則輸出應該是

```
3,1,4;12
```

如果輸入是

```
3,2
2,9,1
0,4,6
4,0,4
6,4,0
```

則輸出應該是

```
2,1;4
```

## 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 3.9 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：

- 確定可以使用的語法包含之前作業說過可以使用的語法。
- 確定不可以使用的語法包含自定義函數、tuple、dictionary、利用 `print` 的任何格式化輸出法（例如百分比、f-string、`str.format()`）、類別等等。

請注意正面表列的固然是都確定可以用，但沒有被負面表列的不表示可以用喔！

## 評分原則

- 這一題的其中 30 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。本題共有 15 組測試資料，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可讀性（包含排版、變數命名、註解等等）。請寫一個「好」的程式吧！

## 第四題（加分題）

（20 分）炭治郎在一個有  $n$  種面額銅板的國家賣自家製作的炭，銅板面額由小到大依序是  $p_1$ 、 $p_2$  直到  $p_n$ 。某日，炭治郎辛苦一天之後，發現自己收錢的盒子裡有  $p_i$  元銅板  $x_i$  枚， $i = 1, \dots, n$ 。他帶著這些

錢，考慮買些東西回家。炭治郎來到糧食行，依序考慮裡面的  $m$  種食材，其中第  $j$  種食材的價格是  $q_j$  元。炭治郎想要最大化他買的食材的總價值（或者說，把身上的錢花掉愈多愈好）。請注意並不是從最貴的食材開始考慮，就一定能花掉最多的錢喔！

在本題中，請計算炭治郎回家時身上剩多少錢。舉例來說，如果  $n = 3$ 、 $p_1 = 1$ 、 $p_2 = 5$ 、 $p_3 = 10$ 、 $x_1 = 18$ 、 $x_2 = 6$ 、 $x_3 = 8$ ，則炭治郎共賺進  $10 \times 8 + 5 \times 6 + 18 = 128$  元。若  $m = 4$ 、 $q_1 = 50$ 、 $q_2 = 30$ 、 $q_3 = 70$ 、 $q_4 = 10$ ，則炭治郎應購買食材 1、3，花掉 120 元，回家時身上剩 8 元。已知  $m$  不會太大，合理地窮舉所有的排列組合後，可以在時間範圍內解出此題。

## 輸入輸出格式

系統會提供一共數組測試資料，每組測試資料裝在一個檔案裡。在每個檔案中會有四行，第一行有兩個正整數  $n$ 、 $m$ ，第二行有  $n$  個正整數  $p_1$ 、 $p_2$  直到  $p_n$ ，第三行有  $n$  個非負整數  $x_1$ 、 $x_2$  直到  $x_n$ ，第四行有  $m$  個正整數  $q_1$ 、 $q_2$  直到  $q_m$ ，一行中任兩個相鄰的值之間被一個逗點隔開。已知  $1 \leq n \leq 10$ 、 $1 \leq m \leq 10$ 、 $1 \leq p_1 < p_2 < \dots < p_n \leq 1000$ 、 $0 \leq x_i \leq 100$ 、 $1 \leq q_i \leq 10000$ 。

請依題目所述，輸出一個整數代表炭治郎到家後身上的錢的金額。舉例來說，如果輸入是

```
3,4
1,5,10
18,6,8
50,30,70,10
```

則輸出應該是

```
8
```

如果輸入是

```
5,8
1,2,3,50,100
1,2,3,4,5
100,100,100,100,100,100,100,114
```

則輸出應該是

```
0
```

## 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 3.9 程式碼。當然，你應該寫適當的註解。針對這個題目，你**可以**使用上課沒有教過的方法。

## 評分原則

這一題的所有分數都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。