

# 商管程式設計 (110-1)

## 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/>) 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 10 月 30 日晚上九點**。在你開始前，請閱讀課本的五、七章<sup>1</sup>。為這份作業設計測試資料並且提供解答的助教是郭玟妍。

### 第一題

(20 分) 小陳在賣一個商品，他發現如果他把價格設成  $p_i$ ，則需求就會是  $q_i$ ， $i = 1, \dots, n$ 。這個商品的單位成本是  $c$ 。請幫小陳從  $p_1$ 、 $p_2$  直到  $p_n$  中找出能最大化利潤的價格。**如果有多個價格都能最大化利潤，那就挑最高的價格。請注意給定的價格未必遞增、給定的需求量未必遞增，比較低的價格也未必一定帶來比較高的需求量。**

舉例來說，如果  $n = 5$ ，五個可以設定的價格依序是 20、30、40、90、100 元，而五個相對應的需求量為 100、90、80、30、20，且單位成本  $c = 10$ ，則設定這五個價格依序可以賺得利潤 1000、1800、2400、2400、1800 元。按照規則，應該把價格定在 90 元。

### 輸入輸出格式

系統會提供數組測試資料，每組測試資料裝在一個檔案裡。在每個檔案中，第一行會有兩個正整數  $n$  和  $c$ ，第二行會有  $n$  個正整數  $p_1$ 、 $p_2$  直到  $p_n$ ，第三行會有  $n$  個正整數  $q_1$ 、 $q_2$  直到  $q_n$ 。每一行的任兩個數字之間用一個逗點隔開。已知  $1 \leq n \leq 100$ 、 $0 \leq c \leq 500$ 、 $0 \leq p_i \leq 5000$ 、 $0 \leq q_i \leq 5000$ 。

讀入這些資訊後，請依照題目指定的規則，印出能最大化利潤的價格，以及該價格下的利潤，中間用一個逗號隔開。舉例來說，如果輸入是

```
5,10
20,30,40,90,100
100,90,80,30,20
```

則輸出應該是：

```
90,2400
```

如果輸入是

```
5,10
20,30,40,80,100
100,90,80,30,20
```

<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。

則輸出應該是：

40,2400

## 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

(20 分) 承上題，但現在在有複數個價格都能帶來最大的利潤時，請選擇能帶來最大需求量的價格。以第一題的那個例子來說，應該選擇 40 元而非 90 元，因為 40 元能帶來的需求量高於 90 元。

## 輸入輸出格式

系統會提供數組測試資料，每組測試資料裝在一個檔案裡，格式和上一題的格式一模一樣。讀入這些資訊後，請依照題目指定的規則，印出能最大化利潤的價格，以及該價格下的利潤，中間用一個逗號隔開。舉例來說，如果輸入是

5,10  
20,30,40,90,100  
100,90,80,30,20

則輸出應該是：

40,2400

如果輸入是

5,10  
20,30,40,95,100  
100,90,80,30,20

則輸出應該是：

95,2550

## 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

(60 分) 你正要去登山，在思考要帶哪些東西。你有  $n$  個東西可以選，每個東西都有其重量跟效用（價值），並且不可分割（要就不帶，不然就整個帶）。令第  $i$  個東西的重量與效用各為  $w_i$  公斤與  $v_i$  單位。你的背包（或你）最多只能承受  $B$  公斤的負重，你想要在此限制下最大化你帶的東西的總效用。

下面是一個例子。你最多只能背  $B = 5$  公斤，而你的七個東西如下表所示：

編號 $i$	1	2	3	4	5	6	7
名稱	指南針	柴刀	火柴	塑膠布	望遠鏡	鋼瓶	拉拉熊
重量 (公斤) $w_i$	0.5	1.5	0.4	1	1.1	1.6	0.8
效用 $v_i$	6	5	4	4	3	4	1

顯然你不能全帶，因為這樣總負重 6.9 公斤就超過負重限制了。你可以帶指南針、柴刀、火柴、塑膠布和望遠鏡，這樣的總效用是 22；你也可以把望遠鏡換成鋼瓶，你還是背得動，且這樣總效用就提高到 23 了。事實上，在這個例子中 23 就是你所能得到的最大效用了。

這個問題就是資訊科學領域中有名的「背包問題」(knapsack problem)。如果要更精確地描述這個問題，我們可以寫一個數學模型，或者更精確地說，一個「整數規劃模型」(integer program)<sup>2</sup>。令  $x_i \in \{0, 1\}$  表示我們是否帶物品  $i$ ， $x_i = 1$  表示要帶，為 0 則否。則背包問題的模型即為

$$\begin{aligned} \max \quad & \sum_{i=1}^n v_i x_i \\ \text{s.t.} \quad & \sum_{i=1}^n w_i x_i \leq B \\ & x_i \in \{0, 1\} \quad \forall i = 1, \dots, n. \end{aligned}$$

任意一個「解」(solution) 都可以被一個維度為  $n$  的向量  $x = (x_1, \dots, x_n)$  表示，若  $\sum_{i=1}^n w_i x_i \leq B$  則  $x$  為一個「可行解」(feasible solution)，若  $\sum_{i=1}^n v_i x_i$  還不小於任何一個可行解的總效用，則  $x$  為一個「最佳解」(optimal solution)。

<sup>2</sup>老師每次藉著出作業的機會，跟同學們介紹一些資訊科學領域的經典題目時，都會順便把數學模型寫下來給同學們看，這有兩個理由。首先，因為數學是精確定義的語言，而學習程式設計本來就是為了在未來能更精確地敘述問題、任務、演算法。其次，如果各位同學未來有打算多學一點機率、統計、計量經濟、機器學習、資料科學、作業研究、資料結構、演算法等科目，就一定會見到大大小小的數學模型，先看看不是壞事，看久了就習慣了。當然老師也收過抱怨「寫什麼數學模型啊，題目就已經很難懂了，數學模型更看不懂啊」，老師也理解，所以這些數學模型純屬補充，有興趣的同學就看，沒興趣的請就跳過，不會影響你作答的。

背包問題有非常多的應用。舉例來說，當零售商要決定在店裡展示什麼商品時，他有有限的貨架空間、各商品要佔去的空間與預期營收不一，零售商要最大化其總預期營收；公司的有限預算要用來從數個專案中挑一些出來執行，每個專案所需的預算與報酬不一，公司要最大化總報酬。當然在大多數的應用中，它都伴隨著其他因子一起以更複雜的形式出現，但精神是不變的。

在資訊科學（computer science）與作業研究（operations research）領域中，學者們普遍認為背包問題是「困難的」，也就是說只要可選的物品數夠多、背包夠大，那任何演算法都無法在合理的時間內求得最佳解。因此，本題並不要求你針對給定的背包問題求得最佳解。在本題中，我們將給你一組解，你只要檢查這組解是否是可行解，以及在它是可行解的時候求出它所包含的物品的總重量與總效用即可。

## 輸入輸出格式

系統會提供數組測試資料，每組測試資料裝在一個檔案裡。在每個檔案中會有四行資料，第一行含有兩個正整數  $n$  與  $B$ ，分別是物品個數與負重上限；第二行含有  $n$  個正整數  $w_1, w_2$  直到  $w_n$ ，其中  $w_i$  是物品  $i$  的重量；第三行含有  $n$  個正整數  $v_1, v_2$  直到  $v_n$ ，其中  $v_i$  是物品  $i$  的效用；第四行含有  $n$  個正整數  $x_1, x_2$  直到  $x_n$ ，其中  $x_i$  代表物品  $i$  是否要被攜帶，若有則為 1，反之則為 0。已知  $n \in \{1, 2, \dots, 1000\}$ 、 $B \in \{1, 2, \dots, 10000\}$ 、 $w_i \in \{1, \dots, 100\}$ 、 $v_i \in \{1, \dots, 100\}$ 、 $x_i \in \{0, 1\}$ 。每一行的任兩個值之間被一個逗號隔開。

讀入這些值之後，你應該檢查第四行資料中建議攜帶的物品，如果要被攜帶的物品的總重量超過  $B$ ，則印出 -1；反之則印出兩個整數，分別建議攜帶的東西的總重量，以及建議攜帶的東西的總效用，中間用一個逗號隔開。

舉例來說，如果輸入是

```
7,50
5,15,4,10,11,16,8
6,5,4,4,3,4,1
1,1,1,1,0,1,0
```

則輸出應該是

```
50,23
```

如果輸入是

```
5,9
2,3,4,3,8
2,4,5,3,10
1,1,1,1,0
```

則輸出應該是

```
-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 3.9 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 `for`、`while`、各種維度的清單、Python 內建的所有操作清單的函數（包含參數只有一個清單的函數，以及由清單後面加一個「點」去呼叫的函數），以及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 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。本題共有 20 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）

**在你開始之前：**「加分題」的設計，是給寫完基本題還不滿足、還想練習更多的同學。所以加分題總是比較難，如果你在做基本題時就已經非常吃力了，那真的不用勉強自己一定要完成加分題，更不用煩惱「為什麼加分題這麼難，我是不是很廢」。完成基本題就完全滿足授課團隊對一個修課學生的期望了。反過來說，如果你做完基本題還有一些時間，我們當然建議你也挑戰加分題，畢竟能把加分題做出來，對自己實力的提升肯定也有幫助的。

（20 分）承上題，現在我們將給你許多組解、要求你判斷各組解是否為可行解，並在可行解中找出其中最好的解。你可以假設我們會給你至少一組可行解。**如果有多個最好的解，請挑出編號最小的那個。**

## 輸入輸出格式

系統會提供數組測試資料，每組測試資料裝在一個檔案裡。在每個檔案中會有  $k + 4$  行資料，其中  $k$  是要檢視的解的個數。第一行含有兩個正整數  $n$  與  $B$ ，分別是物品個數與負重上限；第二行含有  $n$  個正整數  $w_1, w_2$  直到  $w_n$ ，其中  $w_i$  是物品  $i$  的重量；第三行含有  $n$  個正整數  $v_1, v_2$  直到  $v_n$ ，其中  $v_i$  是物品  $i$  的效用；第四行含有一個**正整數  $k$** ；第五行起的  $k$  行各含有  $n + 1$  個正整數，第  $j + 4$  行代表第  $j$  個解。在第  $j + 4$  行中，第一個值  $m$  是解的編號，第二個到第  $n + 1$  個值則依序是  $x_{j,1}, x_{j,2}$  直到  $x_{j,n}$ ，其中  $x_{ji} \in \{0, 1\}$  代表第  $j$  組解是否有帶第  $i$  個物品，若有則為 1，反之則為 0。已知  $n \in \{1, 2, \dots, 1000\}$ 、 $B \in \{1, 2, \dots, 10000\}$ 、 $w_i \in \{1, \dots, 100\}$ 、 $v_i \in \{1, \dots, 100\}$ 、 $k \in \{3, 4, \dots, 100\}$ 、 $m \in \{1, 2, \dots, k\}$ 。每一行的任兩個值之間被一個逗號隔開。請注意編號不一定有排序。

讀入這些值之後，你應該印出三個整數，分別是最好的解的編號、要帶的東西的總重量，以及要帶的東西的總效用。兩個值中間用一個逗號隔開。舉例來說，如果輸入是

```
7,50
5,15,4,10,11,16,8
6,5,4,4,3,4,1
4
1,0,0,0,0,0,0,0
2,1,1,1,1,1,1,1
3,1,1,1,1,1,0,0
4,1,1,1,1,0,1,0
```

則輸出應該是

```
4,50,23
```

如果輸入是

```
5,9
2,3,4,3,8
2,4,5,3,10
4
1,1,1,0,1,0
4,1,0,1,1,0
3,0,1,1,1,0
2,0,0,0,0,1
```

則輸出應該是

```
2,8,10
```

請注意第二跟第四組解都是最好的解，但是第二組解的編號是 4，比第四組解的編號 2 來得大，此時按題目規定，你應該印出編號為 2 的解的資訊。

## 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。