

商管程式設計 (112-2)

作業五

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 **4 月 17 日中午十二點**。在你開始前，請閱讀課本的第三、六章¹。為這份作業設計測試資料並且提供解答的助教是胡捷翔。

第一題

(20 分) 有一家工廠生產並銷售 n 種商品（編號為商品 1、商品 2 直到商品 n ），每一種商品都有其單位生產時間 m_i 分鐘，以及單價 p_i 元。工廠有恰好 2 台機器，每天各自的產能依序為 K_1 以及 K_2 分鐘（意即兩台機器每天最多能運轉 K_1 和 K_2 分鐘）。已知第一台機器比第二台機器新，因此 $K_1 \geq K_2$ 。一台機器一天只能生產一種商品，且一種商品一天只能被一台機器生產。請幫助此工廠選擇恰好兩種商品生產，使其可以獲得最大的收益；換句話說，就是去 n 種商品中找「價格時間比」 $\frac{p_i}{m_i}$ 最大的前兩種商品，讓第一台機器（產能較高的）生產 $\frac{p_i}{m_i}$ 最大的商品，第二台機器（產能較低的）則生產 $\frac{p_i}{m_i}$ 次大的商品。如果有些商品的價格時間比相同，就以編號較小的商品優先。

舉例來說，假設這家工廠總共有 3 種商品，依序的單位生產時間為 5、7、8 分鐘，而單價依序為 3、4、6 元。假設機器一跟機器二的產能分別是 560 分鐘和 280 分鐘，則經計算可知三種商品的價格時間比依序為 $\frac{3}{5}$ 、 $\frac{4}{7}$ 、 $\frac{6}{8}$ ，因此，第一台機器應該生產價格時間比最大的三號商品，總共可以生產 70 個並且幫公司賺進 420 元，而第二台機器應該生產價格時間次大的一號商品，總共可以生產 56 個並且幫公司賺進 168 元。公司總共能賺 588 元。

在本題中，給定商品、產能資訊後，我們要請你找出機器一和機器二各應該生產哪個商品，以及在這樣最佳的生產計劃下公司一天能賺多少錢。此外，由於要幫助大家練習定義和實作函數，本題我們要請你使用函數來實作。請你定義一個函數叫做 `best_product_info`，會根據傳入的商品種類數量 `item_cnt`、存有每一種商品單位生產時間的清單 `minutes`、存有每一種商品單價的清單 `prices`，並回傳相對應的兩種最可以獲利的商品編號。這個函數大概長得像是：

```
# item_cnt: integer
# minutes: a list of item_cnt integers
# prices: a list of item_cnt integers
# return the two most profitable product index
# if tie, let the smallest index be the first
def best_product_info(item_cnt, minutes, prices):
    # do something here
    return best_index, second_best_index
```

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

請完成這個函數以達成题目的要求吧！

特別說明：在這一題需要特別注意的是，助教已經在 PDOGS 上放好會去呼叫這一個函數的主程式碼了，如附件 PBC112-2_hw05_p1.py 的內容，所以大家只需要實做並且上傳這一個函數的定義程式碼就好，PDOGS 會把助教寫的主程式跟你寫的函數拼在一起直譯、執行。換個角度講，大家也只能上傳這一個函數，如果上傳了其他程式碼（例如你自己寫的主程式），PDOGS 就會拼出重複的程式碼，結果反而無法直譯跟執行了。總之，你上傳的程式碼應該只包含一個函數的定義。

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。在每個檔案中會有三列，第一列包含三個整數，依序為 n 、 K_1 和 K_2 ；第二列含有 n 個正整數 m_1 、 m_2 直到 m_n ；第三列含有 n 個正整數 p_1 、 p_2 直到 p_n ，任兩個整數之間被一個逗號隔開。已知 $2 \leq n \leq 20$ 、 $120 \leq K_2 \leq K_1 \leq 1440$ 、 $1 \leq m_i \leq 50$ 、 $1 \leq p_i \leq 100$ ，並且所有的 m_i 都可以整除 K_1 和 K_2 。

讀入這些資訊後，請依序輸出第一台機器應該生產的商品編號、第二台機器應該生產的商品編號，以及這樣的生產計劃下公司總共能賺多少錢，兩個相鄰的數字之間用一個逗點隔開。舉例來說，如果輸入是

```
3,560,280
5,7,8
3,4,6
```

則價格時間比依序是 $\frac{3}{5}$ 、 $\frac{4}{7}$ 、 $\frac{3}{4}$ ，商品 3 和商品 1 有前兩大的價格時間比，因此要把商品 3 交給機器 1，可以賺 420 元；商品 1 則交給機器 2，可以賺 168 元。因此輸出應該是

```
3,1,588
```

如果輸入是

```
4,1200,800
8,2,10,16
6,1,6,12
```

則價格時間比依序是 $\frac{3}{4}$ 、 $\frac{1}{2}$ 、 $\frac{3}{5}$ 和 $\frac{3}{4}$ ，其中商品 1 和商品 4 有前兩大的價格時間比，因此要生產的是商品 1 和商品 4，且要把編號較小的商品 1 交給機器 1，可以賺 900 元；編號較大的商品 4 則交給機器 2，可以賺 600 元。因此輸出應該是

```
1,4,1500
```

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僅包含題目指定的函數。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) 延續作業二的第三題，此處將完整複製該題題目，並加上一些新的任務敘述。

這星期小傑再度來到花店，希望可以給他心儀已久的女生再次送上「多束」精心挑選的花束。花店本期推出了一種「幸福花束」，每一束都由 r_1 朵紅玫瑰花、 w_1 朵白玫瑰花、 y_1 朵黃玫瑰花組成，定價為 800 元。花店更進一步推出了一種「真心花束」，每一束都由 r_2 朵紅玫瑰花、 w_2 朵白玫瑰花、 y_2 朵黃玫瑰花組成，定價為 1000 元。為了真的送到對方的心坎裡，小傑事先與朋友們分析了那位女生的喜好，據可靠情報指出，小傑買的玫瑰花數量要符合以下三項條件的至少一項，才可能告白成功：

- 紅玫瑰花朵數不小於 K_1 朵；
- 白玫瑰花朵數以及黃玫瑰花朵數加總後不小於 K_2 朵；
- 紅玫瑰花朵數的三倍加上白玫瑰花朵數不小於 K_3 朵。

小傑的預算為 B 元，請幫助小傑找出所有不超過預算的策略共有幾種（顯而易見地，可以選擇不買任何一束花束，雖然這樣基本上不會告白成功），接著在這些策略中，找出所有至少符合一種指定條件的可行策略。最後依序找出「不超過預算的策略」數量，以及「可行策略中的最低價」，如果沒有任何一種策略可行，那麼就先找出「不超過預算的策略」數量，再直接回傳「-1」。

本週我們將練習進一步使用以下的 pseudocode 和函數來實作這一題，本題的 pseudocode 可能長這樣²：

```
Let the given values be read.
Let A_1 be the maximum number of "Happiness Bouquets" that can be bought.
Let A_2 be the maximum number of "True Heart Bouquets" that can be bought.
Let min_price be the budget + 1.
Initialize a counter to 0.

for i from 0 to A_1:
  for j from 0 to A_2:
    if within_budget(i, j, price_happiness, price_heart, budget) is True:
      Increment the counter by 1.
      Let price = meet_one_req(i, j, happy_flower, heart_flower,
        price_happiness, price_heart, budget, requirement). # If it does
        not meet any condition, return the budget B + 1 as the price.
    if price < min_price:
      Let min_price become price.
```

²這裡一層縮排空兩個空白而非四個，主要是因為題目文件寬度有限，這樣排版比較方便；另外，pseudocode 不是程式碼，格式原本也就比較有彈性)

```
Output the results.
```

其中有兩個函數，首先是 `within_budget`，會根據傳入的幸福花束數量 `i`、真心花束數量 `j`、幸福花束的價格 `price_happiness`、真心花束的價格 `price_heart`、小傑的預算 `budget`，並回傳相對應的布林值，大概長這樣：

```
# i, j, budget: integers
# price_happiness: integer
# price_heart: integer
# return true if B is enough to purchase i sets of "Happiness Bouquets" and
  j sets of "True Heart Bouquets"
# return false otherwise
def within_budget(i, j, price_happiness, price_heart, budget):
    is_within_budget = True
    # do something here
    return is_within_budget
```

`meet_one_req` 是第二個函數，參數包含幸福花束數量 `i`、真心花束數量 `j`、存有幸福花束含有的紅、白、黃玫瑰花數量的清單 `happy_flower`、存有真心花束含有的紅、白、黃玫瑰花數量的清單 `heart_flower`、幸福花束的價格 `price_happiness`、真心花束的價格 `price_heart`、小傑的預算 `budget`，以及存有買花的三個目標值 K_1 、 K_2 以及 K_3 的清單 `requirement`，並在這種買法能滿足目標時回傳這種買法的價格，或在不能滿足時回傳小傑的預算值 $+1$ ，大概像：

```
# i, j, budget: integers
# happy_flower: a list of three integers (red, white, and then yellow)
# heart_flower: a list of three integers (red, white, and then yellow)
# price_happiness: integer
# price_heart: integer
# budget: integer
# requirement: a list of three integers ( $K_1$ ,  $K_2$ ,  $K_3$ )
# return the price needed to buy i sets of "Happiness Bouquets" and j sets
  of "True Heart Bouquets" if such a purchase plan meets at least one
  requirement
# return the budget value + 1 otherwise
def meet_one_req(i, j, happy_flower, heart_flower, price_happiness,
    price_heart, budget, requirement):
    # do some calculation here ...

    if # ...
        return total_price
    else:
        return budget + 1
```

請完成兩個函數以達成題目的要求吧！

特別說明：助教已經在 PDOGS 上放好會去呼叫這兩個函數的主程式碼了，所以大家只需要實做並且上傳這兩個函數的定義程式碼就好，PDOGS 會把助教寫的主程式跟你寫的函數拼在一起直譯、執行。換個角度講，大家也只能上傳這兩個函數，如果上傳了其他程式碼（例如你自己寫的主程式），PDOGS 就會因此拼出重複的程式碼，結果反而就會無法直譯跟執行了。總之，你上傳的程式碼應該只包含兩個函數的定義。

輸入輸出格式

在本題中，爲了練習模組化，助教會先在 PDOGS 上放好「資料讀取模組」的程式碼，先把題目輸入存到 `budget`、`price_happiness`、`price_heart`、`heart_flower`、`happy_flower`、`requirement`，其中前三項變數爲整數（`int`），後三項變數爲 Python 的清單（`list`）：

- `budget` 爲一個非負整數，代表小傑的預算。
- `price_happiness`、`price_heart` 各爲一個非負整數，分別代表一束幸福花束的價格與一束真心花束的價格。
- `heart_flower` 爲裝著三個非負整數的一維清單，其中 `heart_flower[0]`、`heart_flower[1]`、`heart_flower[2]` 分別代表一束真心花束擁有的紅、白、黃花朵數量。
- `happy_flower` 爲裝著三個非負整數的一維清單，其中 `happy_flower[0]`、`happy_flower[1]`、`happy_flower[2]` 分別代表一束幸福花束擁有的紅、白、黃花朵數量。
- `requirement` 爲裝著三個正整數的一維清單，其中的第零、一、二個元素分別是 K_1 、 K_2 、 K_3 ，也就是要送到小傑心儀已久的女生心坎的花朵數門檻。

再次提醒大家，助教已經在 PDOGS 上放好讀取輸入資料的程式碼了，所以大家只需要實做並且上傳後續的程式碼，PDOGS 會把助教寫的程式跟你寫的程式拼在一起直譯、執行。換個角度講，大家也只能上傳你該寫的部分，如果你的程式碼中還有 `input()` 等等不該出現的指令，PDOGS 就會因此拼出重複的程式碼，結果反而就會無法直譯跟執行了。

舉例來說，如果輸入是

```
5
10
4
3
6
2
200
100
50
6000
```

則輸出應該是

```
31,1600
```

如果輸入是

```
20
5
10
5
20
10
800
900
900
5000
```

則輸出應該是

```
23, -1
```

如果輸入是

```
20
5
10
5
20
10
800
900
900
18000
```

則輸出應該是

```
226, 11200
```

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僅包含題目指定的函數。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) 會處理文字資料 (字串) 之後, 可以做許多有趣的應用! 雖然我們暫時還無法讓大家實作一個大型語言模型, 但在本題中, 我們要來實作一個經典的文字探勘應用: 情感分析 (sentiment analysis)。情感分析的應用很多, 最常見的例子是品牌商去論壇上爬文, 並且針對有提到自家品牌的句子分析該句的情緒為正面還是負面。舉例來說, 考慮到「商管程式設計」也是一個品牌, 如果在某論壇上看到「我真的是很討厭『商管程式設計』這門課」, 那最好程式可以判斷出這是負面情緒; 如果是「我是不喜歡『商管程式設計』, 但是這門課實在讓我的實力提升了」, 最好程式可以判斷出這是正面情緒。當然這整件事是有一定困難度的, 例如需要辨識「商管程式設計」和「商管程」和「PBC」都在講同一門課, 還要知道「很討厭」是負向的而「不討厭」是正向的, 可能也要評估各種正向詞和負向詞的程度不一等等, 中文更有「斷詞」這樣的任務, 但在本題中, 我們就在最陽春的場域底下體驗一下就好!

在本題中, 我們將給你 n 個英文句子, 以及一份情緒辭典來對每一篇文章進行情感分析, 情緒辭典包含³:

- 正向字有 5 個: 「good」、「best」、「awesome」、「excellent」、「wonderful」。
- 負向字有 4 個: 「bad」、「worst」、「stupid」、「shame」。

你的任務是去比對每一個句子中各出現多少次正向字以及負向字 (同一個字出現數次的話每次都算)。若第 i 句話的正向字次數為 x_i^{pos} , 負向字次數為 x_i^{neg} , 則這個句子的情緒分數 y_i 為

$$y_i = x_i^{\text{pos}} - x_i^{\text{neg}}。$$

也就是一句話的情緒分數, 即為正向字的總出現次數減去負向字的總出現次數⁴。進行情緒字比對時, 一定要整個單字完整且恰好地出現在文章中, 才會被算作一次, 例如 `goodbye` 不算對應到正向字 `good`。此外, 比對時大小寫英文被視為相同, 例如 `Awesome` 算是有對應到 `awesome`。

舉例來說, 若題目給定以下 2 個句子 (一句一句)

The weather today IS wonderful, but the traffic IS the worstttt.
I think this new movie IS AWESOME, but the actors are stupid.

經檢查可知, 第一句的 x_1^{pos} 為 1 (出現一次 `wonderful`), x_1^{neg} 為 0 (`worstttt` 不算是 `worst`), 因此 $y_1 = 1 - 0 = 1$ 。第二句的 x_2^{pos} 為 1 (出現一次 `awesome`), x_2^{neg} 為 1 (出現一次 `stupid`), 因此 $y_2 = 1 - 1 = 0$ 。

在評估一個字 (word) 是否「完整且恰好出現」時, 只要該字前後的各一個字元有任何一個是英文字母, 那就不算完整且恰好出現; 如果該字前後的各一個字元都是標點符號或空白字元, 那就算是完整且恰好出現。請根據題目指示, 依序輸出每一個句子的情緒分數。

特別說明: 後續作業我們會延續這一題, 加入簡單的機器學習機制, 讓大家感受一下機器學習 (其中的監督式學習) 的運作原理。為了之後方便, 現在請大家盡力寫這一題, 並且把這一題的程式碼盡你所能寫得乾淨、模組化、將來好利用吧! 如果你對此還是很模糊, 可以嘗試把握以下原則: (1) 將輸入模組、演算模組和輸出模組切得乾乾淨淨; (2) 設計並實作數個函數, 每個函數都負責完成一件可以被簡單說明其輸入、任務、輸出的工作。剩下的就等之後慢慢體會就好!

³實務上的情緒辭典都非常地長, 裡面的詞成千上萬, 但這題我們簡單就好。

⁴實務上對不同的字詞可能會設定不同的權重或者分數, 但這題我們就簡單就好。

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。在每個檔案中將有 $n + 1$ 列，第一列有一個整數 n ；第二列至第 $n + 1$ 列有 1 個英文句子字串，字串內容中只包含大小寫英文字母、英文標點符號與空白字元。本題會出現的標點符號只有句點、逗點、冒號、分號、問號、驚嘆號、單引號、雙引號跟連接號 (.,:;?!'"-)。已知 $1 \leq n \leq 20$ ，每個句子的字元數不超過 1000。給定的句子不一定符合正確的英文文法，標點符號前後也不一定有被正確使用的空白字元；句子開頭可能是空白字元；句子結尾也可能是空白字元。

請讀入以上資料，接著輸出 n 個整數，依序為 y_1 、 y_2 直到 y_n ，代表每一句英文句子的情緒分數，兩兩之間以一個逗點隔開。舉例來說，如果輸入是

```
2
The weather today IS wonderful, but the traffic IS the worstttt.
I think this new movie IS AWESOME, but the actors are stupid.
```

則輸出應該是

```
1,0
```

如果輸入是

```
7
Despite the bad weather, we had a wwwonderful time at the beach.
The movie was awesome, but the seats were quite uncomfortable.
I thought the new cafe was overrated; the food was good but NOT excellent.
This has been the worst day of my life; everything that could go wrong did.
My Laptop Is The    Best Purchase I've Made. wONderful PerformANCE!
I will found the book quite STUPID, with a plot that madenosense.
Winning the award was an "Awesome " experience ,truly the best-feeling.
```

則輸出應該是

```
-1,1,2,-1,2,-1,2
```

請注意在這個例子的第五句出現了很多奇怪的大小寫和不合理的連續空白，但這不影響我們判定正負情緒字的出現；第六句有出現算是錯字的「madenosense」（應該要是「made no sense」），但也不影響我們判斷；第七句的右雙引號左邊多了一個空白字元，逗點前面多了一個後面少了一個空白字元，但這都不影響我們判定「awesome」是情緒字而「experience」不是。最後，第七句的「best-feeling」中是含有情緒字「best」的，因為「best」的前後各一個字元都不是英文字母。

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、自定義函數、字串以及所有字串處理的函式庫、利用 `print` 的任何格式化輸出法（例如百分比、f-string、`str.format()`）等等。

- 確定不可以使用的語法包含 tuple、dictionary、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！