商管程式設計（106-2）
作業九

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/`）為第一題上傳一個 PDF 檔、為第二題做同儕互評，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 ***2018 年 6 月 4 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吳沛璇。

第一題

（40 分）請回答以下問題。

(a)（20 分）請寫一個 Python 3.6 程式，去建立一個簡單的視窗，裡面有兩個文字方塊和一個按鈕，文字方塊可以讓使用者輸入數字，而當使用者按下按鈕，文字方塊內的數字就會被相加然後取代第一個文字方塊內的內容，第二個文字方塊內的內容則維持不變。你可以假設使用者最多輸入一個二位數字，所以相加後最多是一個三位數字。請貼上你的程式碼和你創造的視窗的截圖。

(b)（20 分）本學期至今我們也教了蠻多東西的，應該也讓一些同學從無到有地學會寫程式了。請考慮你本人所屬的科系，告訴我們一門系上必修課，是你認為適合在其中要求修課同學用我們有教的程式設計做點事的。那門課並不是要教程式設計，而是要「用」程式設計。請告訴我們是哪門課、那門課裡面的什麼任務適合融入程式設計，以及為什麼。請盡量不要寫超過三百字。

第二題

（0 分）請在 PDOGS 上批改你被隨機分配到的作業八第三題的程式碼，根據它在正確性以外的部份給它 1 至 5 分的評分，並且說明你給分的依據。建議在評分時參考以下六個面向。在前五個面向上，一個面向上做得好就得一分，還不錯則半分，不好則零分；在第六個面向上則在有必要時扣分。六個面向的分數合計後無條件進入即為你最後給的總分。

- 可讀性：變數與函數名稱是否具有合適的資訊量？程式碼排版是否良好且具有前後一致性？是否有合適的註解？關於註解，當然不需要每一行都有註解，但若你發現在某一大段落裡都沒有註解，或某個你感覺很不易看懂的部份沒有註解，你可以指出來；不要直接說「註解太少」但沒有說是哪邊缺乏註解。
- 模組化程度：是否有宣告合適的函數與 class？是否有避免將非常類似的程式片段寫複數次而非寫成函數？是否有避免一個函數做非常多事情？函數間是否有合適的 decoupling？直接閱讀程式是否能很快地理解程式在大方向上的運算邏輯？

- 效率：程式運算是否有合理的運算效率？當然我們不要求每個同學都寫出超級有效率的精妙演算法，但至少一個程式不應該進行過多不必要的運算，也不應該耗用過多不必要的記憶體空間。如果你看不出這個程式的效率有明顯的問題，我們建議你直接給一分。
- 擴充性：當要解的問題變得更複雜的時候，我們能不能簡單地修改這個程式以解決新的問題，而不是寧可砍掉重練？這個議題當然也很主觀，所以如果你不能明確地指出在怎樣的新問題上，這個程式會有擴充性問題，我們建議你直接給一分；如果你不能指出很嚴重的問題，我們建議你至少給半分。但對批改者來說，這個關於擴充性的思考其實是很好的訓練。試試看吧！
- 其他：如果有任何其他令你想扣分的理由，請明確地寫出來並且在這個面向上扣分；沒有的話就給一分。
- 題目規範：你應該檢查那份程式碼有沒有違反題目的規範，如果有（例如題目說不可以用上課沒教過的東西，但他用了，或者題目說一定要用指標和動態記憶體配置，但他沒用），就扣他三分。當然，請明確地指出他哪邊違反了題目的規範。

本題其中 10 分取決於檢視你的程式碼的同學給你的分數總和（必要時助教會出來主持公道，請不用緊張），另外 10 分取決於你對同學的程式碼的評語和評分的合理性和建設性（原則上除非被申訴，且助教檢視後發現你確實評得很不公允，否則只要有評就會得到 10 分）

第三題

（60 分）你要從一個工廠把生產出的產品送到一個市場，因此你必須選擇路徑。你得要先從工廠把產品送到一間庫房，接著從庫房送到一個物流中心，最後再從物流中心送到市場。如果你有兩個庫房、三個物流中心，那麼所有可能的路段就會如圖 1 所示。我們可以看到我們有一個工廠 F、兩個庫房 1 和 2、三個物流中心 1 到 3，以及一個市場 M。從工廠理論上可以送貨到任意一個庫房，從任一庫房理論上可以送貨到任一物流中心，從任一物流中心理論上都可以送貨到市場，但工廠不能直接到物流中心或市場，庫房不能直接到市場。在一條路段上，我們用一個括號內的數字代表此路段的最大可送貨數，以下我們將之稱為產能。舉例來說，從工廠到庫房 1 的產能為 10（可以送最多 10 單位）、從庫房 1 到物流中心 2 的產能為 4，依此類推。請注意一條路段的產能可能為 0，例如庫房 1 到物流中心 3，此時表示庫房 1 事實上不能送貨到物流中心 3。

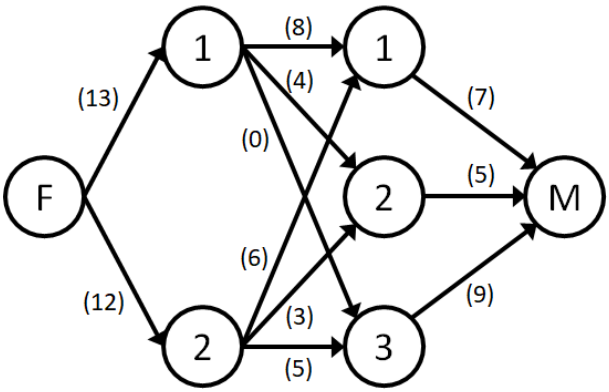

圖 1: 範例路段圖

假設你的工廠的生產產能是無上限，市場需求也無上限，你送多少貨進市場都賣得掉，那麼你的路段選擇問題基本上就是一個最大流量（maximum flow）問題：你要決定利用每一條路段的產能以最大化送進市場的總貨品數。以圖 1 來說，顯然我們無法從工廠送到庫房超過 25 單位的產品，從庫房到物流中心則最多只能送 26 單位，物流中心到市場則最多 21 單位，三者綜合之後，顯然最多只能送 21 單位進市場，但最佳方案也有可能比 21 單位還小[1]。

我們這裡考慮的最大流量問題，是一個比較特別的版本（因為工廠不能到物流中心這一類的限制），有興趣的同學可以查查「bipartite maximum flow」；對於一般的最大流量問題，事實上是有演算法可以在有效時間內求得最佳解，不過有點複雜，所以我們要請大家實做一個比較簡單、不一定能求得最佳解的 heuristic 演算法（啟發性演算法）。方法描述如下。

首先，我們從工廠出發，在所有由工廠往外連的可用路段中，挑出剩餘產能最大的一條，抵達一個庫房。接著從該庫房出發，在所有從該庫房往外連且還沒被用過的路段中，挑出剩餘產能最大的一條，抵達一個物流中心，接著從該物流中心沿著唯一一條路段抵達市場。如果遇到有多個路段的剩餘產能都是最大的，我們就選其中物流中心編號最小的。這三段路段（arc）串起來會形成一條路線（path），上面三段路段各有一個剩餘產能，而此路線的剩餘產能自然就是三個路段的剩餘產能中最小的一個，假設叫 f，那我們就決定從工廠沿此路線運送 f 單位到庫房。連接庫房到物流中心的一條路段，最多只會被使用一次。當一個庫房連出去的路段都被用過之後，我們就把從該工廠通往該庫房的路段設為不可用。當所有從工廠連出去的路段都不可用時，演算法就結束。如果在任何時候遇到有多個路段的剩餘產能都是最大的，我們就選其中庫房編號最小（往庫房走時）或物流中心編號最小（往物流中心走時）的。

利用圖 1 的例子，我們來展示這個演算法被使用的結果。首先，我們發現從工廠到庫房 1 的剩餘產能（13）比到庫房 2 的剩餘產能（12）大，所以我們先走到庫房 1。接著從庫房 1 出發，我們發現往物流中心 1 的剩餘產能是最大的，因此我們走到物流中心 1，最後走到市場 1。三個路段的最小產能是 7，所以我們沿著這個路線運送 7 單位。圖 2 中，我們用藍色線表示被使用的路線、虛線表示被用掉的庫房到物流中心的路段、路段上括弧外的數字表示要沿著此路段運送的總貨品數，以及括弧內的數字表示路段剩餘產能。因為工廠往庫房 1 的路段產能只剩下 6，因此步驟二我們選擇由工廠往庫房 2 走。接著由於往物流中心 1 的路段產能最大，所以我們往物流中心 1 走，最後通往市場。但由於此路線上有路段（物流中心 1 到市場）已經沒有剩餘產能，所以我們雖然會「用掉」被畫成虛線的庫房 2 到物流中心 1 的路段，但並不會多運送任何產品。結果會如圖 3。

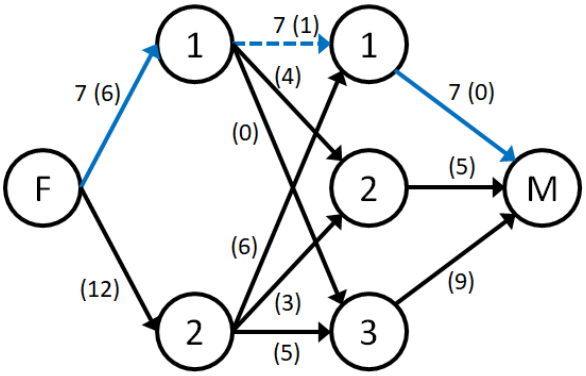

圖 2: 演算法執行範例：步驟一

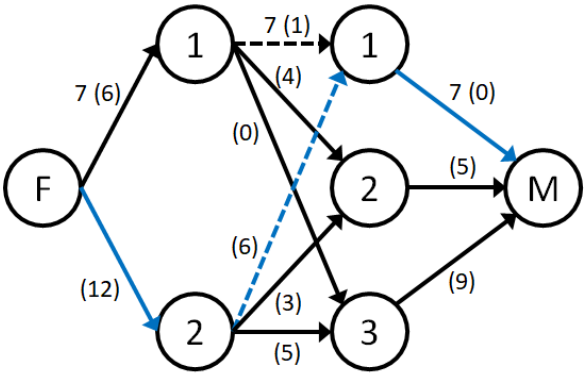

圖 3: 演算法執行範例：步驟二

[1]如果生產產能有上限，我們可以在 F 前面再加一個點 F_0，並且讓 F_0 到 F 的運送產能為工廠生產產能；如果需求有上限，我們可以在 M 後面再加一個點 M_0，並且讓 M 到 M_0 的運送產能為市場需求。如此一來，只要解一個從 F_0 到 M_0 的最大流量問題，就能把生產產能和市場需求考慮進來了。

若我們如法泡製，則在步驟三我們會運送 5 單位產品（圖 4），並在步驟四運送 3 單位產品（圖 5）。此時由於由庫房 2 出發的三條路段都被用掉了，所以我們就把通往庫房 2 的路段也標記為被用掉，在圖 5 中被標記為虛線。

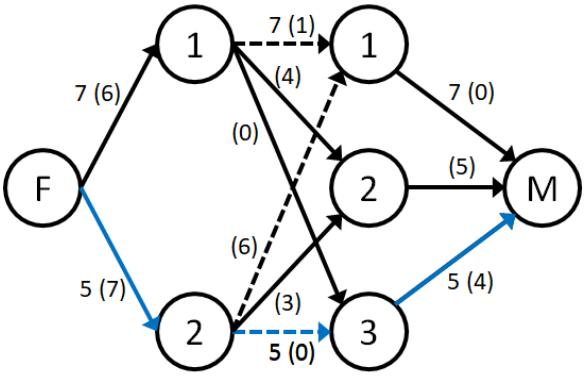

圖 4: 演算法執行範例：步驟三

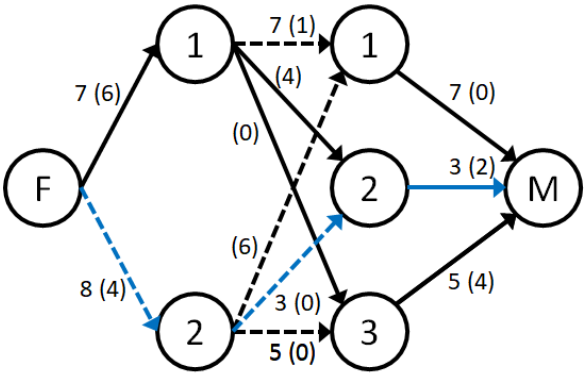

圖 5: 演算法執行範例：步驟四

在步驟五，我們會回到庫房 1，先選擇往物流中心 2（因為剩餘產能 4 比較大），如圖 6 所示；最後，我們會嘗試最後一條還沒有用掉的路段，從庫房 1 到物流中心 3，但因為這條路段上的剩餘產能是 0，所以我們當然也就無法運送任何產品了。至此演算法運算結束，最終的運送計畫即如圖 7 所示，共送了 17 單位。你可以很快地用手算算看，應該能找到更好的運送方案，不過本題並不需要你找到最佳方案。

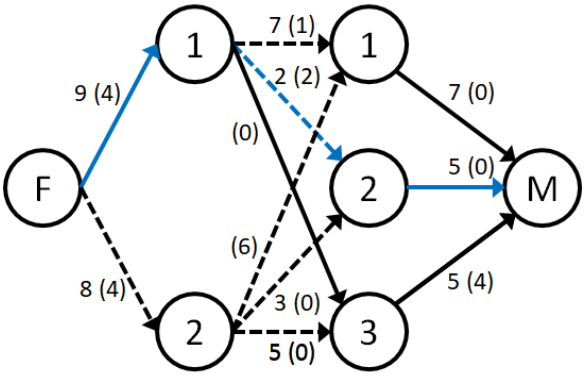

圖 6: 演算法執行範例：步驟五

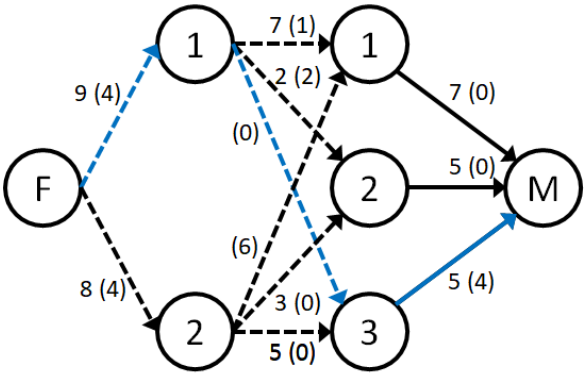

圖 7: 演算法執行範例：步驟六

在本題中，你將被給定兩個正整數 n 和 m，分別是庫房的個數和物流中心的個數。接著你將被給定 $n+m+nm$ 個非負整數，分別是工廠通往庫房、物流中心通往市場，以及庫房通往物流中心的路徑運送產能。你的任務是實做上述演算法，並輸出演算法找到的運送方案。

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 $n+3$ 列資料，其中第一列是兩個正整數 n 和 m，第二列是 n 個非負整數 c_1^F、c_2^F 到 c_n^F，其中 c_i^F 是從工廠到庫房 i 的產能，第三列是 m 個非負整數 c_1^M、c_2^M 到 c_m^M，其中 c_j^M 是從物流中心 j 到市場 i 的產能。第四列起的第 $i+3$ 列是 m 個非負整數 $c_{i,1}$、$c_{i,2}$ 到 $c_{i,m}$，其中 c_{ij} 是從庫房 i 到物流中心 j 的產能。每一列

中的兩個數字都以一個逗點隔開。已知 $1 \le n \le 20$、$1 \le m \le 20$、$0 \le c_i^F \le 10000$、$0 \le c_j^M \le 10000$，以及 $0 \le c_{ij} \le 1000$。

讀入資料後，你的程式要用本題指定的演算法找出運送方案，並以 n 列將庫房到物流中心間的 nm 個路段的運送數量依序印出，在第 i 列依序印出由庫房 i 到物流中心 1、2 直到 m 的運送量。在一列之中，兩個數字之間以一個逗點隔開，最後一個數字後面直接接換行符號，不要接逗點。最後一列後面也要有換行符號。舉例來說，如果輸入

```
2,3
13,12
7,5,9
8,4,0
6,3,5
```

則輸出應該是

```
7,2,0
0,3,5
```

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，但上課介紹過的函式庫中所有的功能都可以用。

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 分會在作業十中被評定。屆時我們會讓同學們互相檢視彼此的本題程式碼，並且就可讀性、易維護性、模組化程度、排版等面向寫評語和給評分（當然一切都是匿名的）。該任務在本題中會佔 20 分，其中 10 分取決於檢視你的程式碼的同學給你的分數（必要時助教會出來主持公道，請不用緊張），另外 10 分取決於你對同學的程式碼的評語和評分的合理性和建設性。若你在本次作業中完全沒有寫這一題，那屆時自然沒有人能檢視你的程式碼，你也就得要損失這 10 分了。

第四題（bonus）

（20 分）承上題，我們知道我們執行演算法時共會有 nm 個步驟，每一個步驟我們會用掉一條庫房到物流中心的路段，直到我們用完為止。現在我們想要使用另一個 heuristic 演算法。我們依然要執行 nm 個步驟，但在每一個步驟中，我們考慮所有還沒被用掉的路線，在其中找出該路線（含三個路段）的剩餘產能最大的，接著就把此路線的剩餘產能用光。如果有複數條路線的剩餘產能同為最大，那就選庫房編號最小的；如果庫房編號也都一樣小，那就選物流中心編號最小的。我們持續這麼做直到所有路線都被用完，或者已經沒有剩餘產能為正的路線為止。完成運算後，請依然印出運送方案。

以圖 1 的例子來說，若我們用 (F, i, j, M) 表示從工廠到庫房 i 到物流中心 j 到市場的一個路線，則執行本題的演算法會依序經過以下六個步驟：

1. 選路線 $(F, 1, 1, M)$，其剩餘產能是 $\min\{13, 8, 7\} = 7$，因此沿著此路線運送 7 單位。
2. 選路線 $(F, 2, 3, M)$，其剩餘產能是 $\min\{12, 5, 9\} = 5$，因此沿著此路線運送 5 單位。
3. 選路線 $(F, 1, 2, M)$，其剩餘產能是 $\min\{6, 4, 5\} = 4$，因此沿著此路線運送 4 單位。
4. 選路線 $(F, 2, 2, M)$，其剩餘產能是 $\min\{7, 3, 1\} = 1$，因此沿著此路線運送 1 單位。
5. 選路線 $(F, 1, 3, M)$，其剩餘產能是 $\min\{2, 0, 4\} = 0$，因此沿著此路線運送 0 單位。

本題的輸入輸出格式和第三題一模一樣。舉例來說，如果輸入

```
2,3
13,12
7,5,9
8,4,0
6,3,5
```

則輸出應該是

```
7,4,0
0,1,5
```

針對這個題目，你**可以**使用任何方法。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。