

程式設計（110-2）

期中專案

題目設計：孔令傑
國立臺灣大學資訊管理學系

說明：本專案題目的類似題目曾經出現在某幾份作業。為了閱讀上的方便，在這份文件中我們完整地描述了整個問題。你可以利用你之前的想法或程式碼¹，不過除非你想多看一些範例，否則你不需要回去看那兩個題目的敘述。

1 題目敘述

你是一家零售業者的作業長（chief operations officer，COO），現在正想要解決一個設置物流中心、建設零售店，以及補貨的問題。在一個地區裡，你有 m 個區域，在每個區域裡你可以建造一家零售店，去銷售某一種商品。你另外有 n 個可以設置物流中心的地點。為了行文方便，我們底下可能會稱呼「設置在地點 j 的物流中心」為「物流中心 j 」，以及「建造在區域 i 的零售店」為「零售店 i 」。

由於貴公司做過精確的市場調查，若在區域 i 興建零售店 i ，你們將會以一個商品 p_i 的價格販售此商品。 p_i 已經被行銷部門決定了，你沒有辦法更改。在給定價格之後，每個區域的需求不一樣。我們將區域 i 的需求量稱為 D_i ，表示若是在這個區域開了一家店，每天最多能賣掉 D_i 個商品。如果要蓋零售店 i ，它會落在 (x_i, y_i) 位置上（單位為公里）。可設置物流中心的地點 j 落在 (u_j, v_j) 位置上（單位為公里）。由於這個區域的路都是東西向或南北向的，因此點和點之間的距離是用曼哈頓距離（Manhattan distance）計算，而非歐幾里得距離（Euclidean distance）。更精確地說，如果你分別設置了物流中心和零售店在地點 j 和區域 i ，則物流中心 j 到零售店 i 的距離是

$$d_{ij} = |u_j - x_i| + |v_j - y_i| \text{ 公里。}$$

要賣東西有兩個前提。首先，你必須蓋一家店，而在區域 i 蓋一家店的成本攤銷之後是一天 f_i 元²，在每個區域你只能蓋最多一家店。其次，這家店前一天晚上要補貨。對於任意一家零售店，你必須決定由哪個或哪幾個物流中心來補貨給它，也必須指定個別的補貨數量。要在地點 j 設置一個物流中心，必須付出攤銷後每天 h_j 元的建造成本，而此物流中心的容量將是 K_j 個商品，亦即物流中心 j 最多只能放 K_j 個商

¹或許你應該感謝當時你有好好模組化，讓你的某些程式（特別是函數）可以被重新利用。如果你正在咒罵當時的你沒有好好模組化，也請不要介意，因為大部分人都是如此。從今天起好好模組化吧！

²舉例來說，如果蓋一家店的成本是一千萬，而這家店可以使用十年，那 $f_i \approx \frac{10000000}{3650} \approx 2739$ 元。

品，所以你不能要求它在一個晚上提供超過 K_j 個商品的補貨數量。補貨成本和補貨數量與補貨距離的乘積成正比，每個商品每公里的單位補貨成本是 c 元。更精確地說，若物流中心 j 每晚送 x_{ij} 個商品到零售店 i ，則其補貨成本為 $cd_{ij}x_{ij}$ 元。對於這些商品，在零售店 i 的銷貨毛利就是 $(p_i - cd_{ij})x_{ij}$ 元了。當然，如果一家店只能賣 100 個商品，而你補 1000 個給它，最後也只能賣出 100 個，剩下 900 個都是浪費錢而已。

給定上述資訊，你的任務是找出能夠最大化預期單日總銷貨淨利的物流中心設置方案、零售店建造方案，以及補貨方式。「預期單日總銷貨淨利」是總銷貨毛利（銷貨收入減補貨成本）減物流中心設置成本再減零售店建造成本。一個地點有設置物流中心，才能從該物流中心進行補貨；一個區域有建造零售店，才能在該區域做銷售；從一個物流中心送出的總補貨數量不能超過其容量；一個零售店的銷售數量不能超過該區域的需求量。物流中心和零售店都只有「蓋或不蓋」，不能花一半的錢蓋一半然後得到一半的容量或需求量。補貨方面可能是單中心補貨（single sourcing）或多中心補貨（multiple sourcing），前者表示一個零售店只能接受一個物流中心補貨，後者則表示一個零售店可以接受來自任意多物流中心的補貨。

請注意你不一定要滿足所有需求。舉例來說，對於某些非常偏遠的地方，遠到連從最近的物流中心補貨的成本都大於售價，那最好的作法就是不要在這些地方做銷售。事實上，你應該可以很輕易地找出其他理由在某些地方不要銷售或不要補貨。

2 輸入輸出格式

系統會提供許多筆測試資料，每筆測試資料裝在一個檔案裡。在每個檔案中，第一列存放四個整數 n 、 m 、 c 和 s ，其中 $s = 1$ 表示限制只能單中心補貨， $s = 2$ 則表示可以多中心補貨。第二列存放 $2n$ 個整數 u_1 、 v_1 、 u_2 、 v_2 直到 u_n 、 v_n ，表示第 i 個物流中心在位置 (u_i, v_i) 上。第三列存放 $2m$ 個整數 x_1 、 y_1 、 x_2 、 y_2 直到 x_m 、 y_m ，表示第 i 個零售店在位置 (x_i, y_i) 上。第四列存放 m 個整數 D_1 、 D_2 直到 D_m ，表示第 i 個零售店的需求量是 D_i 。第五列存放 m 個整數 f_1 、 f_2 直到 f_m ，表示第 i 個零售店的平均每日建造成本是 f_i 。第六列存放 m 個整數 p_1 、 p_2 直到 p_m ，表示第 i 個零售店所在區域的商品單位售價是 p_i 。第七列存放 n 個整數 h_1 、 h_2 直到 h_n ，表示第 j 個物流中心的平均每日設置成本是 h_j 。第八列存放 n 個整數 K_1 、 K_2 直到 K_n ，表示第 j 個物流中心的容量是 K_j 。每一列中的任兩個數字之間都用一個空白鍵隔開。已知 $1 \leq n \leq 50$ 、 $1 \leq m \leq 1000$ 、 $1 \leq c \leq 10$ 、 $s \in \{1, 2\}$ 、 $0 \leq x_i \leq 200$ 、 $0 \leq y_i \leq 200$ 、 $0 \leq u_i \leq 200$ 、 $0 \leq v_i \leq 200$ 、 $1 \leq D_i \leq 1000$ 、 $1 \leq f_i \leq 1000000$ 、 $1 \leq p_i \leq 500$ 、 $1 \leq h_j \leq 10000000$ 、以及 $1 \leq K_j \leq 100000$ 。

舉例來說，如圖 1 所示，假設我有兩個可以建造物流中心的地點 (2, 2) 和 (4, 5)、

六個可以建造零售店的地點 $(1, 1)$ 、 $(1, 2)$ 、 $(2, 5)$ 、 $(3, 1)$ 、 $(7, 1)$ 和 $(8, 4)$ 、單位補貨成本是每個商品每公里 1 元、六個區域的每日需求量各是 3、4、5、6、7 和 8 個商品、六個零售店的每日平均建造成本各是 5、5、5、3、3 和 3 元、六個區域的商品單位售價各是 10、8、7、9、9 和 9 元、兩個物流中心的平均每日設置成本各是 20 和 20 元、兩個物流中心的容量各是 50 和 50 個商品，則輸入會是

```

2 6 1 1
2 2 4 5
1 1 1 2 2 5 3 1 7 1 8 4
3 4 5 6 7 8
5 5 5 3 3 3
10 8 7 9 9 9
20 20
50 50

```

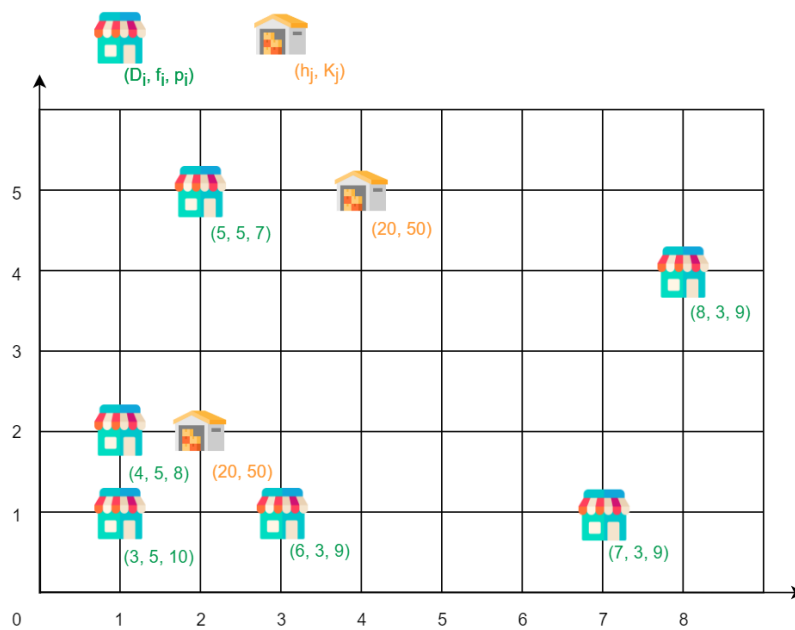


圖 1: 範例實例

請根據題目的描述，找出一個設置物流中心、建造零售店和進行補貨與銷售的計畫，並盡你所能最大化預期單日總銷貨淨利。接著請依照如下格式輸出你的計畫。在第一列首先印出一個介於 0 到 n 的整數 \bar{n} ，代表你要設置的物流中心個數，接著由小到大印出至多 n 個介於 1 到 n 的不重複整數，代表你要設置的物流中心的編號。在第二列首先印出一個介於 0 到 m 的整數 \bar{m} ，代表你要建造的零售店個數，接著由小到大印出至多 m 個介於 1 到 m 的不重複整數，代表你要建造的零售店的編號。在第三列

到第 $m + 2$ 列中，在第 i 列印出 $x_{i,1}$ 、 $x_{i,2}$ 直到 $x_{i,n}$ ，其中 x_{ij} 代表由物流中心 j 對零售店 i 的補貨數量， $\sum_{j=1}^n x_{ij}$ 則代表零售店 i 的銷售量。在每一列都用一個空白字元隔開相鄰的兩個整數。請注意如果限定為單中心補貨，則第三列起的每一列都最多只會有一個非零的數，若為多中心補貨則無此限制。

舉例來說，如圖 2 所示，假設我決定要設置物流中心 1、建造零售店 1、2 和 3，並且從物流中心 1 分別對零售店 1、2 和 3 補貨 3、4、5 個商品，則輸出應該是

1	1
3	1 2 3
3	0
4	0
5	0
0	0
0	0
0	0

根據這個計畫，平均每天的總物流中心設置成本是 $h_1 = 20$ 、總零售店興建成本是 $f_1 + f_2 + f_3 = 15$ 、零售店 1 的銷貨毛利是 $(p_1 - d_{11}c)x_{11} = (10 - 2) \times 3 = 24$ 、零售店 2 的銷貨毛利是 $(p_2 - d_{21}c)x_{21} = (8 - 1) \times 4 = 28$ 、零售店 3 的銷貨毛利是 $(p_3 - d_{31}c)x_{31} = (7 - 3) \times 5 = 20$ ，因此總銷貨淨利是 $24 + 28 + 20 - 20 - 15 = 37$ 元。

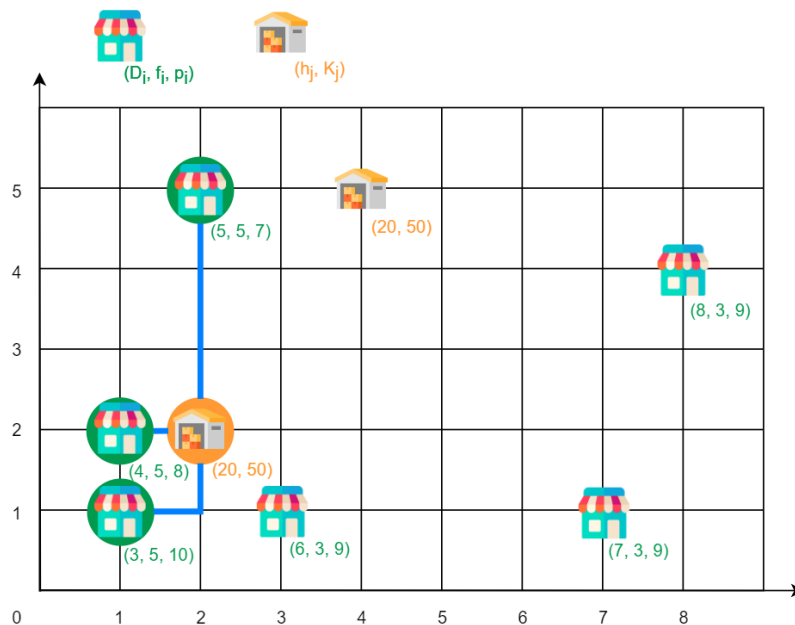


圖 2: 一個可行的範例方案

3 評分原則

這一題的其中 75 分會根據程式運算的結果給分。你的程式不需要找出真的能最大化總淨利的最佳方案 (optimal solution)。只要你的輸出符合規定，且確實是一組可行方案 (feasible solution，例如沒有物流中心的總補貨量超出其容量、沒有零售店的總銷售量超出其需求量、沒有從沒設置的物流中心做補貨之類的)，就會得到分數。對於每一組輸入，PDOGS 會檢查你的輸出，如果輸出格式不合乎要求或方案不可行，則在該筆測試資料會得到零分；如果合乎要求，且目標式值非負，則對每筆測資，我們依下列公式計分：假設 z 是這組的計畫得到的總淨利、 z_0 是所有組的計畫得到的總淨利中最低的、 z_1 是所有組的計畫得到的總淨利中最高的，則在這筆測資的得分就是

$$1.5 + 1.5 \left(\frac{z - z_0}{z_1 - z_0} \right)。$$

以上面的例子而言，得分就是 $1.5 + 1.5 \left(\frac{37 - z_0}{z_1 - z_0} \right)$ 。如果目標式值是負的，則得分為 0。

寫程式之外，每組還需要合力用中文或英文寫一份書面報告（所謂「寫」，就是用電腦打的意思），以組為單位上傳 PDF 檔至 NTU COOL。在報告裡請用文字描述你的演算法（可以用 pseudocode 但不能直接貼 code）、系統的設計（哪個函數做什麼、程式執行的流程等等）、分工方式（誰寫哪個函數、誰負責指揮、誰負責寫書面報告、誰負責買便當等等；當然一個人可以又買便當又寫程式），以及每個人的簡單心得感想。報告**不可以超過八面 A4 紙**。書面報告佔 25 分。這份專案截止後，書面報告才會被批改。

此外，授課團隊會考慮 PDOGS 上的得分以及書面報告上的解法，邀請若干組在合適的上課時間，每組用 10 分鐘跟全班同學介紹自己的解法。被邀請的組可以婉拒上台報告，但上台報告的組可以獲得專案總成績 5 分的額外加分。

4 繳交方式

請修課的同學們自行組成**每組四至五人**的小組，以組為單位繳交你們的程式和報告。

有兩件事需要注意。首先，系統會以該組內任意一位同學的最後一次上傳得到的分數，做為該組所有人的分數，所以愈傳愈低分是有可能的。其次，原則上 PDOGS 不限制兩次上傳間的時間間隔，但如果許多組在同個時間大量地上傳執行時間很長的程式，導致 PDOGS 大塞車，屆時我們會對兩次上傳的時間間隔做出限制。

繳交組員資訊的截止時間是 **11 月 17 日早上八點**，繳交程式的截止時間是 **12 月 14 日早上八點**，繳交書面報告的截止時間是 **12 月 16 日早上八點**。錯過分組期限扣 10 分，錯過書面報告期限一天內扣 10 分，超過一天扣 25 分。