

程式設計（109-1）

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，第一題請以 PDF 檔繳交至 NTU COOL；另外請至 PDOGS (<http://pdogs.ntu.edu.tw/judge/>) 為第二、三題各上傳一份 C++ 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。請以英文或中文作答。

這份作業的截止時間是 **11 月 3 日早上八點**。在你開始前，請閱讀課本的第 5.20–5.22 和第 19 章¹。為這份作業設計測試資料並且提供解答的助教是黃楷翔。

第一題

（40 分）在一個二維平面，每個城鎮都座落在整數座標上。題目會給定 n 、 m ，則從座標平面上 $(0,0)$ 、 $(n,0)$ 、 $(0,m)$ 與 (n,m) 四個角落點所圍成的矩形區塊中，我們要挑一個整數座標來蓋一間醫院，以最大化覆蓋的總人數。題目會給定 p_{ij} ，代表座落在 (i,j) 的城鎮人口數，以及醫院的覆蓋距離 r ，表示若一個在 (i,j) 的城鎮與在 (u,v) 的醫院滿足

$$|i - u| + |j - v| \leq r,$$

則我們說該城鎮有被該醫院覆蓋到。以圖 1 為例，要計算座標 $(3,4)$ 這個候選位置覆蓋到的民眾數時，若 $r = 2$ ，則圖中紅線區域內整數點的 p_{ij} 皆要計算，包括在線上的 $(1,4)$ 、 $(2,5)$ 、 $(3,6)$ 等。

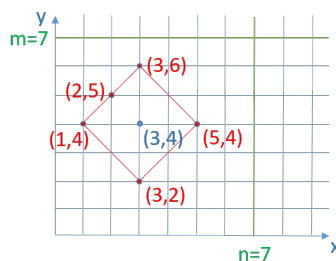


圖 1: 覆蓋範圍定義示意圖

給定醫院在 (x,y) ，要計算覆蓋人數時，我們有兩種演算法。第一種是將 $n \times m$ 座標範圍中的整數點全部掃過一遍，一一計算該座標點與醫院座標的距離，判斷其是否落在題目規定的覆蓋範圍內，若在，就加上該城鎮人口數，不在則不做任何事；另一種演算法是設定 (x,y) 座標條件，讓程式只掃過覆蓋範圍內的整數座標點，並將這些點的城鎮人口數全部相加，得到最終覆蓋的總人口數。本題要求大家利用這兩種演算法，分別寫出「找出最佳醫院位址所能覆蓋的人數」的 pseudocode，並證明哪個版本比較有效率。在本題中我們特別要求，第二種演算法的 pseudocode 中不可包含 if 判斷式，請利用 for 迴圈的範圍限制，設定覆蓋座標點的範圍。

¹課本是 Deitel and Deitel 著的 *C++ How to Program: Late Objects Version* 第七版。

評分原則

這一題的 40 分會根據 pseudocode 和證明的正確性給分。

第二題

(20 分) 上一題我們提供了兩種演算法，並請大家分析。本題我們將請大家實作第二種演算法，根據第一題的題目敘述，請找出能造福最多民眾的醫院位址，計算其覆蓋的民眾總人數，並將之輸出。有趣的是，本題若使用第一種演算法實作，可能會面臨 time limit exceed 的問題，大家也不妨嘗試看看。

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 $m + 2$ 行，第一行有三個整數， n 、 m 、 r ，分別代表 x 座標上限、 y 座標上限和醫院能覆蓋的距離，其中 $1 \leq n \leq 1000$ 、 $1 \leq m \leq 1000$ 、 $0 \leq r \leq 10$ 。第 2 到 $m + 2$ 行各有 $n + 1$ 個數字 p_{ij} ，代表座標 (i, j) 城鎮的人口數，第 2 行為 p_{00} 、 p_{10} 直到 $p_{n,0}$ ，第 3 行為 p_{01} 、 p_{11} 直到 $p_{n,1}$ ，以此類推，則第 $m + 2$ 行為 $p_{0,m}$ 、 $p_{1,m}$ 直到 $p_{n,m}$ 。其中 $0 \leq i \leq n$ 、 $0 \leq j \leq m$ 、 $0 \leq p_{ij} \leq 10000$ 。任兩個數字之間被一個空白隔開。請讀入以上資料，計算出能造福的最大民眾數量，並輸出一個整數。

舉例來說，如果輸入是

```
2 2 1
1 4 7
2 5 8
3 6 9
```

則輸出應該是

```
29
```

如果輸入是

```
3 3 2
5 3 12 20
10 2 18 5
7 1 7 19
6 9 3 17
```

則輸出應該是

```
100
```

你上傳的原始碼裡應該包含什麼

你的.cpp 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 C++ 程式碼。當然，你應該寫適當的註解。針對這個題目，你可以使用任何方法。

評分原則

這一題的 20 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。

第三題（本題為期中考第四題之延伸，但在此我們完整敘述此題）

（40 分）現在有一張只存在整數座標的二維地圖，地圖上有許多村莊，都落在整數座標上，而座標 (x, y) 上的村莊裡有 p_{xy} 位村民。已知有 s 座車站，你希望蓋幾條鐵路以連接某些車站，最大化鐵路沿線居住的村民總數，讓大家都可以自由來往各個村莊，由於預算有限，這些鐵路總長度不會超過題目給定的 L 公里（座標上一格視為一公里）。只要是住在鐵路沿線的村民就會受益，例如若有鐵路連接座標分別為 $(3, 4)$ 、 $(7, 4)$ 的車站，那麼居住在 $(3, 4)$ 、 $(4, 4)$ 、 $(5, 4)$ 、 $(6, 4)$ 、 $(7, 4)$ 這條線段上的居民都會受益。已知鐵路的方向必定為水平方向或垂直方向，不會有斜的鐵路出現，且鐵路的起迄處一定有車站；此外，地圖也有一定的邊界，超出邊界的區域視為不存在，當然也不會有村民居住。

本題我們要實作一個貪婪演算法，首先我們定義一段可建造鐵路的路段（或稱可建造的鐵路）為：一、頭尾皆是車站；二、方向必定為水平方向或垂直方向，不會有斜的；三、預算還足夠，也就是所要新增的鐵路長度不超過 L 扣掉已建造鐵路總長度。演算法會執行數輪挑選，每輪挑選一個可建造鐵路的路段去興建鐵路。對於每個可建造鐵路的路段，我們要計算他們個別的邊際效益比率，計算方法為邊際效益除以該鐵路長度，而所謂邊際效益即是加蓋這條鐵路所增加的受益村民數量。舉例來說，若圖 2 中 $(2, 7)$ 到 $(5, 7)$ 這條鐵路已經被建造，則 $(5, 5)$ 到 $(5, 7)$ 這條鐵路建造的邊際效益即是 $p_{55} + p_{56}$ ，因為 $(5, 7)$ 上的居民在之前建造鐵路時就已經受惠了。因此， $(5, 5)$ 到 $(5, 7)$ 這條鐵路的邊際效益比率即是 $\frac{p_{55} + p_{56}}{2}$ 。計算所有可建造鐵路的邊際效益比率後，我們會選擇最大的那條，若比率相同，則選擇長度較短者；若還是相同，則看各路段連接的兩個車站的編號（皆為整數），選擇編號合計最小的路段，例如若路段 A 連結車站 3 和 8 而路段 B 連結車站 4 和 5，則選擇路段 B 因為 $4 + 5 < 3 + 8$ ；若還是相同，則針對每個路段選出兩個車站編號中較小者，選擇較小車站編號最小的路段，例如若路段 A 連結車站 3 和 8 而路段 B 連結車站 4 和 7，則選擇路段 A 因為 $\min\{3, 8\} < \min\{4, 7\}$ 。重複上述步驟直到沒有可建造鐵路，或者建造任何一條鐵路都無法增加受惠的居民數，即停止演算法，並且完成所有鐵路建造。

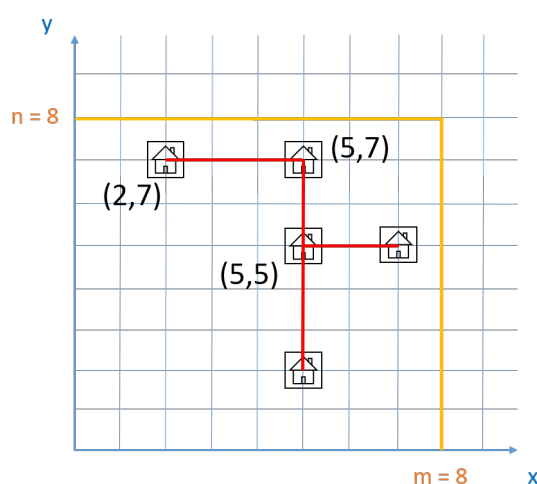


圖 2: 邊際效益計算定義示意圖

這邊提供此貪婪演算法的 pseudocode，一方面帮助大家理解演算法，一方面也做一個 pseudocode 的示範：

```
int railwayBuilding(m, n, s, L, p, 可建設路段集合)
    initialize 剩餘預算 to be L
    initialize 受益人數 to be 0
    initialize 欲建設路段集合 to be empty
    while 可建設路段集合 is not empty
        set 最大比例 to be 0
        set 目前候選路段 to be none
        for each e in 可建設路段集合
            if length(e) > L
                remove e from 可建設路段集合
            else
                // 呼叫函數去計算邊際效益
                b = marginalBenefit(e, m, n, s, p, 欲建設路段集合)
                if b / length(e) > 最大比例
                    set 最大比例 to be b / length(e)
                    set 目前候選路段 to be e
        if b equals 0
            break
        else
            update 剩餘預算 to 剩餘預算 - length(candidate)
            update 受益人數 to 受益人數 + b
            add candidate to 欲建設路段集合
            remove candidate from 可建設路段集合
    return 可建設路段集合 and 剩餘預算
```

請注意 pseudocode 是很有彈性的，精細度也必須由撰寫者自行決定，若你覺得這個 pseudocode 太精細或太粗略，也是很正常的，你可以自行把他改成你想要的樣子。此外，`marginalBenefit` 與 `length` 這兩個函數，都有待你自行定義，而「可建設路段集合」這個東西也要先由你自行定義並且根據題目給定的資料去建構出來，然後才可以執行此貪婪演算法。

請依據給定的資訊，計算並輸出最後總共有幾位因住在鐵路沿線而受益的村民，以及剩下的鐵路單位（ L 扣掉所有已建造鐵路長度）。

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。在每個檔案中有 $m + s + 2$ 列，第一列是個正整數 m 、 n 、 s 和 L ， m 代表地圖水平方向長度、 n 代表地圖垂直方向長度、 s 代表車站的數量、 L 代表鐵路最大總長度，其中 $1 \leq m \leq 100$ 、 $1 \leq n \leq 100$ 、 $0 \leq s \leq 20$ 、 $0 \leq L \leq 10000$ 。第二到第 $m + 2$ 列每列有 $n + 1$ 個整數，依序是 $p_{x,0}$ 、 $p_{x,1}$ 直到 $p_{x,n}$ ，代表座標 (x, y) 上村民的人數，其中 $0 \leq p_{xy} \leq 1000$ 、 $0 \leq x \leq m$ 、 $0 \leq y \leq n$ 。第 $m + 3$ 列起共有 s 列，每列有 2 個整數，代表編號 i 的車站的座標 (x_i, y_i) ，其中 $1 \leq i \leq s$ 、 $0 \leq x_i \leq m$ 、 $0 \leq y_i \leq n$ 。同一列中的任兩個數字之間被一個空白隔

開。

讀入這些資料之後，請輸出因鐵路而受益的村民總人數，和剩下的鐵路長度（預算），兩者以一個逗號隔開。舉例來說，如果輸入是

```
3 2 3 3
1 5 7
2 3 8
9 6 10
3 2 7
1 2
3 1
3 2
```

則首先我們對三個車站做處理，可以判定有兩條可能的路段，分別是連接車站 1 和車站 3 的路段，讓我們稱呼它為路段 A（長度為 2），以及連接車站 2 與車站 3 的路段，讓我們稱呼它為路段 B（長度為 1）。在第一輪的時候，路段 A 的邊際效益比率為 $\frac{8+10+7}{2} = 12.5$ 、路段 B 的為 $\frac{2+7}{1} = 9$ ，因此我們選擇路段 A。此時預算（可以興建的鐵路長度單位）還剩下 $3 - 2 = 1$ ，總受益人數為 25。第二輪只剩下路段 B 可以考慮，預算夠，其邊際效益比率 $\frac{2}{1} = 2$ 高於 0，因此選擇路段 B。由於沒有可興建的路段了，因此演算法結束，總受益人數為 27，剩餘預算為 0，因此輸出應該是

```
27,0
```

如果輸入是

```
3 2 4 3
1 5 7
2 3 8
9 6 10
3 2 7
1 0
1 2
3 1
3 2
```

則輸出應該是

```
27,0
```

如果輸入是

```
4 4 5 7
1 3 2 4 1
2 8 1 4 5
1 3 7 3 5
2 9 7 10 15
4 5 3 6 9
```

1	1
1	4
2	1
2	4
3	4

則在第一輪的時候，連接車站 4 和車站 5 的鐵路邊際效益比率 $\frac{5+15}{1} = 20$ 是最大的，因此選擇該路段，興建後預算剩下 6，總受益人數為 20。在第二輪中，連接車站 1 和車站 3 的鐵路邊際效益比率 $\frac{8+3}{1} = 11$ 是最大的，因此選擇該路段，興建後預算還剩下 5，總受益人數為 31。在第三輪中，連接車站 2 和車站 4 的鐵路邊際效益比率 $\frac{5}{1} = 5$ 是最大的，因此選擇該路段，興建後預算還剩下 4，總受益人數為 36。在第四輪中，連接車站 3 和車站 4 的鐵路邊際效益比率 $\frac{7+3}{3} = 3.33$ 是最大的，因此選擇該路段，興建後預算還剩下 1，總受益人數為 46。此時已經沒有任何可建設路段了（因為預算不夠），因此演算法結束。輸出應該是

46,1

你上傳的原始碼裡應該包含什麼

你的.cpp 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 C++ 程式碼。當然，你應該寫適當的註解。針對這個題目，你可以使用任何方法。

評分原則

這一題的 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。