

程式設計（106-1）

作業六

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，再為後面三題各上傳一份 C++ 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 **11 月 6 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陳維漢。

第一題

(20 分) 針對以下五題是非題，我們會使用 PDOGS 自動批改，因此請寫一個 C++ 程式，內容就是先讀入一個整數，若讀入的數字為 i ，則印出第 i 小題的答案，若為是則印出 1、若為否則印出 0。舉例來說，如果題目只有四題，且你認為答案依序是是、否、是、是，則你上傳的程式碼應該是

```
#include <iostream>
using namespace std;

int main()
{
    int problem = 0;
    cin >> problem;
    if(problem == 1)
        cout << 1;
    else if(problem == 2)
        cout << 0;
    else if(problem == 3)
        cout << 1;
    else
        cout << 1;

    return 0;
}
```

PDOGS 會餵給你的程式的，一定是 1、2 直到 10 這十個整數。有別於作業中一般的程式題，本題在你上傳程式碼時，測試資料是還沒有放上 PDOGS 的，助教會等作業截止後才上傳測試資料（和答案）到 PDOGS 並重新批改此題。換言之，你上傳程式碼時是不會顯示你得幾分的，更不會顯示你對或錯哪些筆測試資料。你會看到你得 0 分，但此數字在助教重新批改之後就會被更新成正確的分數了。

請判斷以下的 big O 關係式是否成立：

- (a) $500n^2 + n\sqrt{n} \in O(n^2)$ 。
- (b) $300n^2 + n \log_2 n \in O(n^3)$ 。
- (c) $300n^2 + \sqrt{n} \in O(n\sqrt{n})$ 。
- (d) $n(n-1)(n-2) + n^2 + 500n + 5 \in O(n^3)$ 。
- (e) $n(n-1) + n^2m \in O(n^2 + m)$ 。

小提醒：在 PDOGS 上面讓大家繳交此題的地方，會有兩組「與上面正式要計分的題目完全無關的」範例輸入輸出，純粹是用來讓大家確認自己那個被批改的 **if-else** 程式是可以被正確執行的。請確認你的程式在針對範例輸入輸出做撰寫後，能讓你在這一題得到「Accepted」，接著再去針對要計分的題目把你的正確答案寫上去然後繳交。當然，即使你曾經看到「Accepted」，也不代表你繳交的題目在這題已經得到滿分了。

第二題

(20 分) 上課時，我們介紹過最短路徑 (shortest path) 問題，並且嘗試實做了演算法求解。本題請你修改上課時寫的程式碼，以應對另一種輸入格式。在本題中， n 是點的個數、 m 是邊的個數、 s 是起點的編號、 t 是目的地的編號、 d_{uv} 是由點 u 到點 v 的距離，圖是有向圖。

系統會提供一共 10 組測試資料，每組測試資料裝在一個檔案裡。每個檔案中有 $m+1$ 行，其中第一行是四個正整數 n 、 m 、 s 和 t ，第二行起則每行是三個整數 u_i 、 v_i 和 d_{u_i, v_i} ，表示第 i 條路是從點 u_i 走到 v_i ，距離為 d_{u_i, v_i} 。同一行的任兩個數字中間用一個空白字元隔開。點的編號為介於 1 和 n (包含 1 和 n 的正整數。已知 $n \in \{1, \dots, 10\}$ 、 $m \in \{1, \dots, n(n-1)\}$ 、 $u_i \in \{1, \dots, n\}$ 、 $v_i \in \{1, \dots, n\}$ 、 $d_{uv} \in \{0, \dots, 100\}$ ，且一條路的資訊不會出現兩次。讀入這些資料後，請輸出最短路徑的總路徑長。

舉例來說，如果輸入是

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

則輸出應該是

```
7
```

如果輸入是

6	9	1	6
1	2	1	
1	3	1	
2	4	1	
3	2	1	
3	4	1	
3	5	1	
4	5	1	
4	6	1	
5	6	1	

則輸出應該是

3

針對這個題目，你可以使用任何方法。這一題的 20 分會根據程式運算的正確性給分，一筆測試資料佔 2 分。

第三題

(60 分) 當一架軍用飛機（戰鬥機、轟炸機、偵察機、補給機等等）要從 A 地飛往 B 地時，可能需要穿越敵方佈置的一些「威脅點」，例如雷達、防空砲、火箭基地等等。不同的威脅點有不同的威脅程度（攻擊力），例如被雷達掃到只是曝光而已，被火箭射中就直接墜毀了；不同的威脅點也有不同的威脅半徑（攻擊範圍）。令 P_i 與 R_i 為威脅點 i 的威脅程度和威脅半徑，如果一架飛機和威脅點 i 的距離為 d ，則我們定義此飛機受到的來自此威脅點的「風險」為

$$P_i \max \left\{ \frac{R_i - d}{R_i}, 0 \right\},$$

亦即在威脅點正上方（距離為零）時會受到 100% 的風險、在威脅點的威脅半徑之外者零風險，而在威脅半徑內的風險則隨著距離拉遠而線性遞減。兩點間的距離以歐幾里得距離（Euclidean distance）計算，亦即若威脅點 i 的位置落在座標 (X_i, Y_i) 上，而飛機在 (x, y) ，單位皆為公里，則距離為 $d_i(x, y) = \sqrt{(X_i - x)^2 + (Y_i - y)^2}$ 公里。一架飛機受到的總風險為所有威脅點給他的風險的和，亦即若此飛機在 (x, y) ，共有 m 個威脅點，則其總風險為

$$L(x, y) = \sum_{i=1}^m P_i \max \left\{ \frac{R_i - d_i(x, y)}{R_i}, 0 \right\}.$$

在本題中，你將被給定 m 個威脅點的座標、威脅半徑與威脅程度，以及一張橫座標與縱坐標皆從 0、1、2 一直到 n 的地圖。請找出整張地圖的 $(n+1)^2$ 個格子點中總風險最高的點。舉例來說，如果 $n=6$ 、 $m=3$ ，威脅點 1、2、3 分別在 $(2, 5)$ 、 $(5, 4)$ 和 $(4, 2)$ ，威脅半徑依序為 2、3 和 2，而威脅程度依序為 2、1 和 2，則點 $(3, 3)$ 的風險為 $L(3, 3) = 0 + 1(\frac{3-\sqrt{5}}{3}) + 2(\frac{2-\sqrt{2}}{3}) \approx 0.840$ ，則點 $(4, 2)$ 的風險則為 $L(4, 2) = 0 + 1(\frac{3-\sqrt{5}}{3}) + 2(\frac{2-0}{2}) \approx 2.255$ ，因此我們說 $(4, 2)$ 的風險高於 $(3, 3)$ 。事實上，在這個範例中 $(4, 2)$ 的風險是所有 49 個地圖上的格子點中最大的。

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，會有五列，第一列含有兩個整數，依序是 n 和 m 。第二列含有 m 個整數，第 i 個整數為 X_i 。第三列含有 m 個整數，第 i 個整數為 Y_i 。第四列含有 m 個整數，第 i 個整數為 R_i 。第五列含有 m 個整數，第 i 個整數為 P_i 。同一列的任兩個數字之間被一個空白字元隔開。已知 $n \in \{1, \dots, 100\}$ 、 $m \in \{1, \dots, 10\}$ 、 $X_i \in \{0, \dots, n\}$ 、 $Y_i \in \{0, \dots, n\}$ 、 $R_i \in \{0, \dots, 10\}$ 、 $P_i \in \{1, \dots, 10\}$ 。讀入這些資料之後，請求出風險最大的格子點的位置，先印出其 x 座標，再印出其 y 座標，中間用一個空白字元隔開。如果有兩個點的總風險一樣高，選 x 座標比較小的；如果再一樣，選 y 座標比較小的。

舉例來說，如果輸入是

```
6 3
2 5 4
5 4 2
2 3 2
2 1 2
```

則輸出應該是

```
4 2
```

如果輸入是

```
6 1
2
5
2
2
```

則輸出應該是

```
2 5
```

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**不可以**使用上課沒有教過的方法。

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 會編譯並執行你的程式、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運算邏輯、可讀性，以及可擴充性（順便檢查你有沒有使用上課沒教過的語法，並且抓抓抄襲）。請寫一個「好」的程式吧！

第四題 (bonus)

(20 分) 承上題，現在我們要擴大 n 跟 m 的範圍到 1000000 和 100。很顯然地，當地圖很大（或者說，一張地圖被切得很細）且威脅點很多時，窮舉去算出每個點的風險然後挑最大值的演算法就會不夠有效率。請自行設計一個比較有效率的演算法吧。

本題除了 n 和 m 的範圍有變，其餘輸入輸出格式都和第三題一模一樣。針對這個題目，你可以使用任何方法。這一題的 20 分會根據程式運算的正確性給分，一筆測試資料佔 2 分。