

程式設計 (106-1)

第二次期中考

題目設計：孔令傑

國立臺灣大學資訊管理學系

請至 PDOGS (<http://pdogs.ntu.im/judge/>) 為後三題各上傳一份 C++ 原始碼。請在發給每位同學的紙本題目卷上寫下第一題的答案後繳回。每位學生都要上傳與繳交自己寫的解答。**作弊被確認者，整門課的成績將直接被評定為 F，沒有第二種可能（也不能停修）。**

所有的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。在所有的題目中，你都可以使用任何方法（包括課堂上沒教過的）。這次考試的截止時間是 **11 月 21 日 17 點 20 分**。不接受遲交。

第一題

(20 分) 請回答下列問題：

(a) (10 分，一小題 2 分) 考慮兩個 8 個 bit 長的帶正負號的二進位數字 $x = 00100110$ 和 $y = 01010011$ ，請做以下計算。請寫出完整的計算過程，否則不予計分。最終的答案應該是 8 個 bit 長的帶正負號的二進位數字。請把你最終的答案**清楚地框起來**，沒框起來者視同答錯。

i. 你的學號與姓名？

答：

ii. 用 1's complement 計算 $x + y$ 。

答：

iii. 用 2's complement 計算 $y - x$ 。

答：

iv. 用 1's complement 計算 $x - y$ 。

答：

v. 用 2's complement 計算 $x - y$ 。

答：

(b) (10 分) 上課時講過如何把給定的 adjacency list 以 adjacency matrix 印出。一種可行的作法是如下兩個函數：

```
void printGraph(int* neighbors [], int degrees [])
{
    for(int i = 0; i < NODE_CNT; i++)
    {
        for(int j = 0; j < NODE_CNT; j++)
            cout << isNeighbor(neighbors, degrees, i, j) << " ";
        cout << "\n";
    }
}

bool isNeighbor(int* neighbors [], int degrees [], int node1, int node2)
{
    for(int i = 0; i < degrees[node1]; i++)
    {
        if(node2 == neighbors[node1][i])
            return true;
    }
    return false;
}
```

給定一個有 n 個 node (也就是上面的程式碼中 `NODE_CNT` 為 n)、 m 個 edge 的 graph，若此 graph 的資訊已經被記錄在 `neighbors` 和 `degrees` 了，請分析上面的兩個函數，以找出這個「把給定的 adjacency list 以 adjacency matrix 印出」的演算法的 time complexity，並以 big O 表示。請注意若你的答案正確但不是最好的答案，會被酌予扣分。例如若最好的答案是 $O(n^5 \log m)$ ，但你寫 $O(n^9 m^5)$ ，則你最多只會得到部份分數。差得愈遠分數愈低，太扯的 (例如 $O(n!m!)$) 會得零分。

答：

第二題

(30 分) 給定 n 個由小到大排序的不重複整數，請找出其中有幾組三個數字滿足畢氏定理（商高定理）。更精確地說，給定 x_1, x_2 到 x_n ，三個數字 x_i, x_j 與 x_k 滿足畢氏定理表示 $x_i^2 + x_j^2 = x_k^2$ 。在求出滿足畢氏定理的組數的同時，也請在有跟其他數字一起滿足畢氏定理的數字中，找出最小的那一個。如果沒有任何一組數字滿足畢氏定理，就印出一個 0 就好。

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，第一列存放一個整數 n ，第二列存放 n 個整數 x_1, x_2 到 x_n 。已知 $1 \leq n \leq 50, 1 \leq x_i \leq 1000, x_1 < x_2 < \dots < x_n$ 。每一列中的任兩個值之間都用一個空白鍵隔開。

讀入資料後，請印出滿足畢氏定理的組數，接著印出一個空白，最後印出這些組之中最小的一個數字。舉例來說，如果輸入是

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

則輸出應該是

```
1 3
```

在 $\binom{5}{3}$ 共 10 組數字中，只有 (3, 4, 5) 符合畢氏定理，而在 3、4、5 這三個數字中 3 最小。如果輸入是

```
7
1 2 8 9 12 15 17
```

則輸出應該是

```
2 8
```

在 $\binom{7}{3}$ 共 35 組數字中，只有 (8, 15, 17) 和 (9, 12, 15) 符合畢氏定理，而在 8、9、12、15、17 這五個數字中 8 最小。如果輸入是

```
1
100
```

則輸出應該是

```
0
```

第三題

(30 分) 給定 n 個整數 x_1 、 x_2 到 x_n ，我們想要先求出平均數

$$\mu = \frac{1}{n} \sum_{i=1}^n x_i,$$

和標準差

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \mu)^2},$$

接著用如下的方式求出「離群值」(outlier)：給定 $k > 0$ ，若一個數字 x_i 滿足

$$|x_i - \mu| > \lfloor \sigma k \rfloor,$$

則我們說它是離群值。請注意 $\lfloor \sigma \rfloor$ 是最接近而不大於 σ 的整數。請由小到大列出所有離群值。

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。在每個檔案中會有兩列，第一列包含兩個整數 n 和 k ，第二列包含 n 個整數 x_1 、 x_2 直到 x_n 。每一列的兩個整數之間被一個空白隔開。已知 $1 \leq n \leq 100$ 、 $k \in \{1, 2, 3\}$ 、 $0 \leq x_i \leq 10000$ 。

讀入資料後，請按照題目指定的規則，由小到大印出所有離群值。兩個離群值中間用一個空白鍵隔開。如果沒有任何離群值，則印出 -1 。舉例來說，如果輸入是

```
10 1
0 36 35 32 25 83 26 38 73 64
```

則輸出應該是

```
0 73 83
```

因為 $\mu = 41.2$ 、 $\sigma \approx 23.73$ 、 $\lfloor \sigma \rfloor = 23$ ，所以小於 $\mu - \lfloor \sigma \rfloor = 18.2$ 或大於 $\mu + \lfloor \sigma \rfloor = 64.2$ 的數字即為給定 $k = 1$ 的離群值。如果輸入是

```
10 2
70 69 61 46 35 44 55 53 62 54
```

則輸出應該是

```
-1
```

第四題

(20 分) 你想要從 n 個應徵者中聘請若干人，編號為 1、2 直到 n 。每個應徵者都有兩個屬性，分別是他們能位公司帶來的貢獻度，以及他們要求的年薪。我們令 c_i 與 s_i 為第 i 個應徵者的貢獻度和年薪。你一共只有 B 這麼多年預算可以聘人，因此你聘的人的總年薪不能超過 B (但可以相同)。你希望在預算範圍內聘人以最大化這些人的總貢獻度。有趣的是，這些人彼此之間有一些「王不見王」的關係，也就是存在有 m 個兩人之間的配對關係，是兩人只能聘其中一人。因此你的聘任方案不能違反王不見王的限制。給定貢獻度、年薪、預算、王不見王資訊後，請找出應該要聘的人。

請使用以下演算法來決定聘任方案。你的演算法將會執行數輪，在每一輪決定聘一個人。在每一輪你將在還沒被聘的人之中，挑選貢獻度最高的人去試圖聘他。但如果這個人和已經聘的人王不見王，就不能聘；如果聘他會超出預算，就不能聘。若不能聘貢獻度最高的人，就聘次高的，次高不行就第三高，依此類推。兩人貢獻度相同時，聘年薪較低的；年薪也平手時，聘編號在前的。你會持續聘人直到無法再聘任何人為止。

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。在每個檔案中會有 $n + m + 1$ 列，第一列包含三個整數 n 、 m 和 B ，第二列包含 n 個整數，依序是 c_1 、 c_2 直到 c_n ，第三列包含 n 個整數，依序是 s_1 、 s_2 直到 s_n 。第四列起的 m 列，每一列包含兩個整數 u_j 和 v_j ，是兩個應徵者編號，表示 u_j 和 v_j 這兩個人不能一起被聘。每一列的兩個整數之間被一個空白隔開。已知 $1 \leq n \leq 100$ 、 $0 \leq m \leq \frac{n(n-1)}{2}$ 、 $\max_{i=1, \dots, n} \{s_i\} \leq B \leq 100000$ 、 $0 \leq c_i \leq 1000$ 、 $0 \leq s_i \leq 1000$ 、 $u_j \in \{1, \dots, n\}$ 、 $v_j \in \{1, \dots, n\}$ ， $u_j < v_j$ ，且最後 m 列中沒有任兩列是一樣的。

讀入資料後，請按照題目指定的規則，依序印出每一輪你聘任的人的編號。兩個編號中間用一個空白鍵隔開。舉例來說，如果輸入是

```
5 2 100
10 10 7 9 25
30 40 25 30 20
1 2
4 5
```

則輸出應該是

```
5 1 3
```

第一輪聘 5 號，因為他貢獻度最高；第二輪選 1 號，因為他年薪比 2 號低；第三輪原本應該選 2 號 (貢獻度最高)，但他跟 1 號王不見王，所以考慮聘 4 號，但 4 號跟 5 號王不見王，因此聘 3 號。如果輸入是

```
5 2 100
10 10 7 9 25
30 40 51 30 20
1 2
4 5
```

則輸出應該是

5 1

跟上一個例子相比，我們不能聘 3 號，因為超出預算了。雖然還有預算，但按照演算法的規則，我們只能停在這裡。