

程式設計 (109-1)

作業七

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 **12 月 1 日早上八點**。在你開始前，請閱讀課本的第 7.1–7.11 節和 22.9–22.13 節¹。為這份作業設計測試資料並且提供解答的助教是鄭亦辰和吳琦艾。

第一題

(20 分) 承作業六第二和三題，本題中我們繼續探討機臺排程與維修問題。假設訂單順序已經排定了，則根據給定的如各訂單訂購的產品數量、各單位時間的生產速率、良率、良率下降函數等，在假設不維修的情況下，我們可以找出每個訂單的開始製造時間。此外，每個訂單會有各自的到期時間 (訂單 i 的到期時間為 d_i)，也就是理想上應該要交貨的時間，若未能在給定的時限內完成，則會有延遲交貨、賠償違約金的情況，造成工廠虧損。在本題中我們會被給定訂單的處理順序，必須先完成訂單 1 才能開始訂單 2，完成訂單 2 才能開始訂單 3，依此類推，則我們可以計算出各訂單的開始與完成時間，進而計算總延遲時間。為了簡單起見，以下的「一個時間段」都以一小時稱之。

舉例來說，給定 $a = 100$ 、 $b = 10$ 、 $c = 1$ 、 $p_0 = 80$ 、 $T = 10$ 、 $L = 20$ 、 $n = 3$ ，且 $q_1 = 100$ 、 $q_2 = 140$ 、 $q_3 = 60$ ， $d_1 = 3$ 、 $d_2 = 5$ 、 $d_3 = 7$ ，則我們首先可按照上題步驟得知，在沒有維修情況下，三個訂單的開始時間為 $t = 1$ 、 $t = 3$ 、 $t = 7$ 。由於訂單 2 會在 $t = 6$ 時才做完，超過 $d_2 = 5$ 一小時，因此訂單 2 被延遲了 1 小時；訂單 3 會在 $t = 9$ 時做完，超過 $d_3 = 7$ 兩小時，因此訂單 3 被延遲了 2 小時，總延遲時間為 $1 + 2 = 3$ 小時。

現在讓我們來考慮維修。在本題中，我們對於安排的維修時間有個條件是「不可中斷任一訂單的製造過程」，意思是假設今天有某個訂單從 $t = 2$ 時開始製造，需要到 $t = 4$ 時才做完，則若要插入一個維修，該維修的起始時間只能選擇在 $t = 2$ 某訂單還沒開始製造時即先進行維修 (因此該訂單的起始時間會往後延)，或是 $t = 4$ 等此訂單整個製造完畢再進行維修 (後面如果有之前排定的訂單，則也往後延)。因此，若給定一個訂單處理的順序，我們就可以計算出無維修之下所有訂單的開始與結束時間 (若前一個訂單在時段 t 結束，則下一個訂單在時段 $t + 1$ 開始)。這些時間點，就給了我們「維修不可中斷任一訂單的製造過程」的要求下，可以考慮的維修時間點，並進而評估若插入維修能否減少總延遲時間。

延續上例，若我們選擇在 $t = 1$ 時進行維修，則 $t = 2$ 時機臺良率將回到 100%，進而我們可以算出三個訂單開始時間會變成 2、3、5，且所有訂單都有在時限前製造完畢，因此總延遲時間為 0。換言之，在 $t = 1$ 維修是最佳策略。

請依此規則檢查每個可維修的時間段，找出能最小化總延遲時間的開始維修時段 (若選擇不維修則輸出 0)，若同時有多個開始維修時段可最小化延遲交貨時間，則選擇較早的那一個 (若不維修和某維修時段同時可最小化延遲時間，請選擇不維修)，並依序印出開始維修時段和總延遲時間。

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

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。每個檔案中會有兩列，第一列包含 7 個數字 a 、 b 、 c 、 p_0 、 T 、 L 、 n ，分別代表理想生產速率、良率遞減速率、維修所需時數、 $t = 0$ 的良率、總可用生產時間數、最低良率和訂單張數。其中 $1 \leq a \leq 1000$ 、 $1 \leq b \leq 10$ 、 $1 \leq c \leq 5$ 、 $1 \leq p_0 \leq 100$ 、 $1 \leq T \leq 16800$ 、 $1 \leq L \leq p_0$ 、 $1 \leq n \leq 20$ ；第二列有 n 個正整數 q_i ，代表各訂單訂購量，其中 $1 \leq q_i \leq 500$ ， $1 \leq i \leq n$ ；第三列有 n 個正整數 d_i ，代表各訂單交貨時間，其中 $1 \leq d_i \leq T$ ， $1 \leq i \leq n$ ；數字兩兩皆以一個空白隔開。請根據題目給定的資訊，依序輸出開始維修時段和總延遲時間，兩者以一個逗號隔開。

舉例來說，如果輸入是

```
100 10 1 80 10 20 3
100 140 60
3 5 7
```

則輸出應該是

```
1,0
```

如果輸入是

```
100 2 1 94 5 70 2
120 180
2 5
```

則輸出應該是

```
0,0
```

如果輸入是

```
100 10 4 80 10 20 3
200 40 15
2 8 9
```

則輸出應該是

```
0,2
```

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。

第二題

(40 分) 承上題，本題一樣會給定 n 個訂單，但此訂單給定順序並非製造順序，我們希望能排定一個訂單製造順序和維修時段，來最小化延遲交貨時間。換言之，你可以任意地安排訂單的製造順序。

某方面來說，這題是有辦法用窮舉的方式求解的：把 $n!$ 種訂單順序都考慮一次，每次就呼叫你在第一題寫的函數（如果你有寫的話）來找尋最佳維修時間與所得到的總延遲時間，然後看看哪種訂單順序最好即可。但由於 $O(n!)$ 基本上在 n 很大時是無法承受的，因此本題將請你實作一個較容易的演算法來排出訂單順序：首先，將交貨日期最早的訂單排在前面（快來不及了的火燒眉毛事趕快做啊）；若有交貨日期相同的訂單，則比較其訂購量，較少的排在前面（能很快搞定的小廢事趕快做啊）；若訂購量也相同，就比較其訂單編號，較小的排在前面。

舉例來說，若給定 $a = 100$ 、 $b = 10$ 、 $c = 1$ 、 $p_0 = 80$ 、 $T = 10$ 、 $L = 20$ 、 $n = 3$ ，且 $q_1 = 140$ 、 $q_2 = 100$ 、 $q_3 = 60$ ， $d_1 = 3$ 、 $d_2 = 3$ 、 $d_3 = 7$ ，在不維修的情況下，因為訂單 1 和 2 皆在 $t = 3$ 時需要交貨，我們比較其訂購量，由於 $q_2 < q_1$ ，因此我們將訂單 2 排在前面，訂單 1 排在後面，最後是交貨日期最晚的訂單 3；若修改資訊 $q_1 = 100$ ，則訂單 1 和 2 交貨日期、訂購量皆相同，就將編號較小的訂單 1 排在前面，然後依序是訂單 2 和 3。

請先排定這樣一個訂單處理順序，然後再呼叫你在第一題寫的函數（如果你有寫的話）來找尋最佳維修時間與所得到的總延遲時間，這樣就好了。找出可最小化延遲時間的開始維修時段後，請印出該時段和總延遲時間。

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。每個檔案中會有兩列，第一列包含 7 個數字 a 、 b 、 c 、 p_0 、 T 、 L 、 n ，分別代表理想生產速率、良率遞減速率、維修所需時數、 $t = 0$ 的良率、總可用生產時間數、最低良率和訂單張數。其中 $1 \leq a \leq 1000$ 、 $1 \leq b \leq 10$ 、 $1 \leq c \leq 5$ 、 $1 \leq p_0 \leq 100$ 、 $1 \leq T \leq 16800$ 、 $1 \leq L \leq p_0$ 、 $1 \leq n \leq 20$ ；第二列有 n 個正整數 q_i ，代表各訂單訂購量，其中 $1 \leq q_i \leq 500$ ， $1 \leq i \leq n$ ；第三列有 n 個正整數 d_i ，代表各訂單交貨時間，其中 $1 \leq d_i \leq T$ ， $1 \leq i \leq n$ ；數字兩兩皆以一個空白隔開。請根據題目給定的資訊，依序輸出開始維修時段和總延遲時間，兩者以一個逗號隔開。

舉例來說，如果輸入是

100 10 1 80 10 20 3
140 100 60
3 3 7

則輸出應該是

1,1

如果輸入是

```
100 10 1 80 10 20 3
100 100 60
3 3 7
```

則輸出應該是

```
1,1
```

如果輸入是

```
100 10 4 80 10 20 3
200 40 15
2 9 9
```

則輸出應該是

```
0,2
```

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。

第三題

(40 分) 在作業四第三題我們曾經依據給定的 n 個機場選址、建造各機場的花費 c_j ，以及兩機場間航線創造的利潤 b_{ij} ，實作過一個演算法最大化整體利潤（詳細演算法過程請見作業四第三題）。在先前的題目中，我們假設兩機場間一定會有航線且能創造利潤，但我們都知道事實上並非如此，兩機場間也可能無航線飛行，因此即使兩個機場都蓋了，也不會創造利潤；換言之，會有很多 $b_{ij} = 0$ 。

本題一樣會給定 n 個機場選址、建造各機場的花費 c_j ，另外會多給定一個 m ，代表能創造利潤的航線數量，且有可能 n 很大而 $m \ll n^2$ ，因此一個 adjacency matrix 會導致 PDOGS 上的 memory limit exceed。為了解決這個問題，我們建議你改用 adjacency list 來儲存 b_{ij} 資訊，如果 $b_{ij} = 0$ 就不要存。除了這個差異之外，請根據相同的演算法，找出能讓總利潤最大化的機場興建計畫，依序印出各機場是否要興建，並印出總利潤。

特別注意：動態記憶體配置

值得特別注意的是，等等當大家讀到輸入輸出格式時，會發現輸入資料中並不會預先告知每一個城市會有幾個其他城市是有正的潛在營收的（換言之，對每個 i 你並不知道有幾個 j 滿足 $b_{ij} > 0$ ），因此你就無法在建造 adjacency list 的時候，知道針對每個城市要宣告一個多大的動態陣列。理論上你有兩種作法：

1. 一開始的時候，為每個城市宣告一個長度為 1 的動態陣列，當看到這個城市出現第二個有潛在收益的鄰近城市時，就把這個長度為 1 的動態陣列的長度拉長 1 單位（具體做法是：宣告一個長度為 2 的動態陣列，接著把原本的動態陣列上的資訊複製過去，接著把原本的動態陣列 `delete` 掉，最後把指標指向新的動態陣列）；如果之後又看到第三個有潛在收益的鄰近城市，就再把長度從 2 拉長到 3；依此類推。這個方法非常有彈性並且不會浪費空間，但缺點是要花很多時間，特別是如果要把已經很長的動態陣列拉長 1 單位，那個宣告、複製、釋放的動作會很花時間。
2. 與前一個方法類似，但用稍微浪費空間的方式節省時間：一開始的時候，為每個城市宣告一個長度大約是 $\frac{m}{n}$ 的動態陣列，當這個城市有太多具有潛在收益的鄰近城市、原本的動態陣列裝不下時，就一口氣把陣列的長度拉長兩倍到大約 $\frac{2m}{n}$ ，再有下一次就拉長到大約 $\frac{4m}{n}$ ，依此類推。這樣做的缺點是會浪費一些空間，但時間方面會節省許多。

在本題中，我們建議你使用第二個作法。如果你使用第二個作法並且寫出不算太奇怪的程式，原則上測試資料的設計會讓你的程式在 PDOGS 上不發生 time limit exceed 或 memory limit exceed。

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。每個檔案有 $m + 2$ 行，第一行為兩個正整數 n 和 m ，代表機場選址數量和航線數量，其中 $1 \leq n \leq 10000$ 、 $1 \leq m \leq \frac{n(n-1)}{2}$ ；第二行為 n 個整數 c_i ，代表各機場建造成本，其中 $0 \leq c_i \leq 10000$ 、 $1 \leq i \leq n$ 。第 3 到 $m + 2$ 行分別有 3 個數字 i_k 、 j_k 和 b_{i_k, j_k} ，代表第 i_k 個機場與第 j_k 個機場之間航線創造的利潤，其中 $0 < b_{i_k, j_k} \leq 100$ 、 $1 \leq k \leq m$ 、 $1 \leq i_k < j_k \leq n$ 。在每一行中，相鄰的兩個數字之間都是用一個空白字元隔開。請找出能讓總利潤最大化的機場興建計畫，按照編號由小到大輸出要興建的機場的編號，兩兩以一個逗號隔開，接著輸出一個分號，最後再輸出總利潤。

舉例來說，如果輸入是

```
3 3
5 10 15
1 2 20
1 3 20
2 3 5
```

則輸出應該是

```
1,2,3;15
```

如果輸入是

```
3 2
```

```
5 15 30
1 2 20
1 3 25
```

則輸出應該是

```
1,2;0
```

如果輸入是

```
5 8
2 4 10 11 10
1 2 20
1 3 1
1 5 9
2 3 15
2 4 4
2 5 7
3 4 10
4 5 2
```

則輸出應該是

```
1,2,3,4,5;31
```

如果輸入是

```
5 1
2 4 10 11 10
1 2 20
```

則輸出應該是

```
1,2;14
```

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 20 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（加分題）

（20 分）承上題，我們要做的事情一模一樣，但是現在給定的不是機場編號，而是城市代碼（TPE、TSA、SFO 之類的），並且有四種可能的字串格式：

- 正常：三碼連著的英文大寫字母，例如「TPE」。
- 混入小寫：部分字母小寫，例如「TpE」。
- 混入空格：前兩個字母或後兩個字母之間出現一或兩個空格，例如「TP E」或「T PE」或「T P E」等各種可能。
- 同時混入小寫與空格。

但不管怎麼說，只要把小寫轉成大寫、空格刪掉，就保證能對應到一個正確的機場。

在本題中，為了簡單起見，我們保證 n 不會太大，因此 adjacency matrix 也是可以的。請讀入資訊，使用與之前相同的演算法，並按照字典順序印出應該興建機場的城市代碼，以及能賺得的總利潤。

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。每個檔案有 $m + 3$ 行，第一行為兩個正整數 n 和 m ，代表機場選址數量和航線數量，其中 $1 \leq n \leq 100$ 、 $1 \leq m \leq \frac{n(n-1)}{2}$ ，兩者之間以一個空白字元隔開；第二行為 n 個格式正常的字串，每個字串都是相連的三個英文大寫字母，第 i 個字串是第 i 個城市的代碼，兩兩之間以一個空白字元隔開；第三行為 n 個整數 c_i ，代表第 i 個機場的建造成本，其中 $0 \leq c_i \leq 10000$ 、 $1 \leq i \leq n$ 。第 4 到 $m + 3$ 行分別有兩個字串與 1 個數字，兩兩之間以一個空白字元隔開。兩個字串的格式可能混入空格與小寫，但去除空格與小寫後，便能正確辨識出是兩個城市的代碼。數字代表兩個城市都有機場後能賺得的利潤，介於 0 與 100 之間。由於字串可能混入空格，因此在給定這些潛在的航線資訊時，格式混亂的城市代碼前後會被用雙引號包住，且左雙引號與第一個字母、右雙引號與第三個字母之間都沒有空白。請找出能讓總利潤最大化的機場興建計畫，用正確格式（無小寫無空白）按照字典順序印出應該興建機場的城市代碼，兩兩以一個逗號隔開，接著輸出一個分號，最後再輸出總利潤。

舉例來說，如果輸入是

```
3 3
TPE SFO LAX
5 10 15
"TPE" "SFO" 20
"TPE" "La X" 20
"s f o" "LAX" 5
```

則輸出應該是

```
LAX ,SFO ,TPE ;15
```

如果輸入是

```
3 2
TPE SFO LAX
5 15 30
"TPE" "SF o" 20
"t p e" "l a x" 25
```

則輸出應該是

```
SFO ,TPE ;0
```

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**使用上課沒有教過的方法。

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。