

# 中華民國參加1996年（第八屆） 國際資訊奧林匹亞(IOI)競賽報告(Ⅱ)

何榮桂

國立臺灣師範大學資訊教育研究所

IOI'96	任務描述	工作
--------	------	----

輸入容器的內容：

"A"類機器：

中間容器的內容：

"B"類機器：

圖一

## 1-2 工作處理

有一間工廠有一生產線。每件工作必須執行兩個作業：首先做作業"A"，然後做作業"B"。每個作業都有數部機器可以替它們執行。圖一顯示出此生產線的結構。其工作方式如下。一台"A"類機器自輸入容器中取出一件工作，執行作業"A"，然後將此工作放入中間容器中。一台"B"類機器自中間容器中取出一件工作，執行作業"B"，然後將此工作放入輸出容器中。所有機器能夠同時處理而且相互間互為獨立，而且每個容器的容量是無限的。這些機器有不同的操作性能，每台機器有它自己的作業處理所需時間。

求出所有N件工作之作業"A"可以被完成的最早時間，假設這些工作在時間0時均備妥待處理（子任務A）。也請計算出要完成所有N件工作之兩個作業所需的最少時間量（子任務B）。

輸入資料

輸出檔案INPUT.TXT含有五行的數個正整數。第一行記載N，即工作的數量（ $1 < N <= 1000$ ）。第二行給定"A"類機器的台數M1（ $1 <= M1 <= 30$ ）。第三行含有M1個整數，為每一台"A"類機器的工作處理時間。第四行和第五行分別含有"B"類機器的台數M2（ $1 <= M2 <= 30$ ）以及每一台"B"類機器的工作處理時間。工作處理時間是以單位時間來計算的，它已包括了在處理前自一容器中取出一件工作並在處理後將之放回容器中的時間。每一工作處理時間至少為1，至多為20。

### 輸出資料

你的程式應寫入二行到檔案 *OUTPUT.TXT* 中。第一行應有一正整數，即為子任務 A 的答案。第二行應含子任務 B 的答案。

### 輸入及輸出範例

圖二為一可能的輸入檔案以及其對應的輸出檔案。

-----  
*INPUT.TXT*  
-----

5  
2  
1 1  
3  
3 1 4

-----  
*OUTPUT.TXT*  
-----

3  
5

---

IOI'96

任務描述

網路

---

## 1-3 學校網路

數個學校連在電腦網路上。這些學校達成協議：每個學校將傳送軟體給某一些學校（“接收學校”）。

如果我們有一份新的軟體且希望所有學校都能收到一份，請計算出最少我們必須先將此軟體送給幾個學校才可（子任務 A）。

另一任務，我們要確定如果送軟體到任一學校，該軟體會到達網路中所有學校。

為達成該目標，我們必須擴展接收學校。

計算最小的擴展數目，使得我們不論送軟體到任一學校，

它會到達所有其它學校（子任務 B）。

每一個擴展是任一個學校每增加一個接收學校。

### 輸入資料

檔案 *INPUT.TXT* 的第一行包含一整數  $N$ ：網路中學校的數目 ( $2 \leq N \leq 100$ )。

學校的編號是前  $N$  個正整數。接下來的  $N$  行中每行描述某學校的所有接收學校。第  $i + 1$  行列出第  $i$  個學校的接收學校編號，而且每行以 0 結束，若沒有接收學校則該行只有 0，注意，若  $B$  是學校  $A$  的接收學校，則  $A$  未必是  $B$  的接收學校。

### 輸出資料

你的程式應該輸出兩行到檔案 *OUTPUT.TXT*，第一行應該包含一正整數：子任務 A 的答案。第二行應該包含子任務 B 的答案。

### 輸入和輸出範例

圖一提供一個可能的輸入檔和對應輸出檔。

<i>INPUT.TXT</i>	<i>OUTPUT.TXT</i>
5	1
2 4 3 0	2
4 5 0	
0	
0	
1 0	

圖一

IOI'96

任務描述 (第二天)

排序

## 2-1 三數值序列之排序

排序是最常見的運算動作。假設有個特殊的排序問題：每一筆要被排序的資料（數字）都有三種可能的數值。舉例來說，當我們在排序比賽獎牌時，我們會照獎牌類 (*modal valuc*) 來排，也就是將所有的金牌排在最前面，再來是所有的銀牌，最後才是所有的銅牌。

在這個任務中，每一筆資料的可能數值皆為正整數 1, 2, 或是 3；你必須將所有資料排序成非遞減 (*non-decreasing*) 之次序。排序的方法必須是一系列的資料互換的動作。第  $p$  筆跟第  $q$  筆資料互換的動作，是指存在第  $p$  筆的資料與存在第  $q$  筆的資料的互換。

給定一系列的資料值，寫個程式來計算最少要做幾次互換的動作，才能將此系列排序完畢（子任務 A）。另外，找出這些互換的動作（子任務 B）。

### 輸入資料

輸入檔 *INPUT.TXT* 的第一行含有  $N$  ( $1 \leq N \leq 1000$ )，此是要被排序的資料個數。接下來的  $N$  行各含有一筆資料的數值。

### 輸出資料

請輸出完成排序所須之最少互換動作次數  $L$  到輸出檔 *OUTPUT.TXT* 的第一行（子任務 A）。接下來的  $L$  行，請按照互換動作的次序輸出每個互換的動作。每一行用兩個數字  $p$  和  $q$  來代表一個互換動作。 $p$  和  $q$  是兩筆應被互換的資料的位置（子任務 B）。假設資料位置是從 1 到  $N$ 。

輸入和輸出範例

圖一提供一個輸入檔和一個對應輸出檔。

INPUT.TXT	OUTPUT.TXT
9	4
2	1 3
2	4 7
1	9 2
3	5 9
3	
3	
2	
3	
1	

圖一

IOI'96

任務描述

字首

2-2 最長字首 (Prefix)

某些生物體的結構是由它們的組成序列來表示。這些組成是由大寫字母代表。生物學家想要把長序列分解成較短序列。每個短序列都叫做元件 (*primitive*)。若從某個元件集合  $P$  內可以提出  $n$  個元件  $p_1, \dots, p_n$ ，使得元件  $p_1, \dots, p_n$  的連結等於某序列  $S$ ，則我們說序列  $S$  可以由元件集合  $P$  構成。元件  $p_1, \dots, p_n$  的連結指的是它們依該次序放在一起，而且中間沒有空白。在連結時，相同的元件可以出現超過一次，而且未必所有  $P$  集合的元件都出現。例如，序列  $ABABACABAAB$  可以由下列元件集合構成：

$\{A, AB, BA, CA, BBC\}$ 。

$S$  的前  $K$  個字元，就是  $S$  的長度為  $K$  的字首。

寫一程式，輸入某元件集合  $P$  和某序列  $T$ 。計算可以由  $P$  內元件構成的最長字首的長度。

輸入資料

輸入資料出現在以下兩個檔案。檔案  $INPUT.TXT$  描述元件集合  $P$ ，而檔案  $DATA.TXT$  包含要檢查的序列  $T$ 。 $INPUT.TXT$  的第一行包含  $N$ ，即  $P$  內元件的個數 ( $1 \leq N \leq 100$ )，每個元件由兩個連續行所提供。第一行包含元件的長度  $L$  ( $1 \leq L \leq 20$ )。而第二行為長度  $L$  的大寫 (由 'A' 到 'Z') 字串元件。每一個元件都不同。

檔案  $DATA.TXT$  的每一行的第一個位置包含一個大寫字母。該檔案以單一句點 (') 的新行結束。

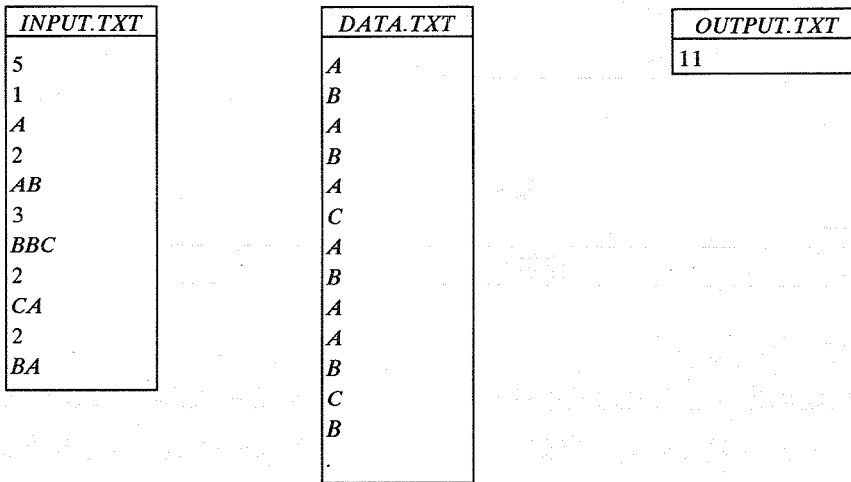
此序列的長度是至少為 1，至多為 500,000。

輸出資料

計算可以由集合 P 構成的 T 的最長字首的長度，然後輸出到檔案 *OUTPUT.TXT* 的第一行。

輸入和輸出範例

圖一提供兩個輸入檔和一個對應輸出檔。



圖一

IOI'96

任務描述

魔術

魔術矩陣

1	2	3	4
8	7	6	5

圖一：起始盤面 (Initial configuration)

在魔術方塊成功之後，Rubik 先生又開發了它的平面版本，稱為魔術矩陣。此盤面含有八塊大小相同的方格子（見圖一）。在此任務中，我們考慮的版本中每一方格子的顏色均不相同，並由前八個正整數來表示。一個盤面可由顏色序列來表示，並從最左上角的方格子開始，然後依順時針方向來得到此顏色序列。舉例來說，圖一的盤面可由序列 (1,2,3,4,5,6,7,8) 來表示。這個盤面是起始盤面 (*initial configuration*)。

有三種基本的轉換可以使用，分別以字母 'A'、'B'、'C' 來表示如下：

'A'：上一列 (*top row*) 和下一列 (*bottom row*) 互換；

'B'：此矩陣向右環狀位移 (*right circular shift*) 一格；

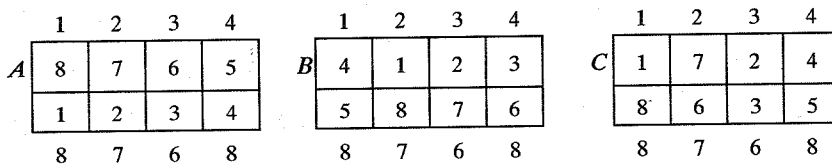
'C'：中間四個方格子做順時針旋轉一格。

所有的盤面都可以用這三種基本轉換來產生。

這些基本轉換的效果顯示在圖二上。方格外面的數字表示方格的位置。如果在位置  $p$  裡面的數字是  $i$ ，則這意味著在轉換後，原來在位置  $i$  的方格子會被移到位置  $p$ 。

子任務 A：你需要寫一個程式去求出一序列的基本轉換，使得圖一的起始盤面可以被轉換到某一個特定的目標盤面 (*target configuration*)。

子任務 B：如果你的轉換序列長度沒有超過 300，那麼此答案可額外獲得兩分。



圖二：基本轉換

#### 輸入資料

輸入檔 *INPUT.TXT* 的第一行含有八個正整數，用來描述目標盤面。

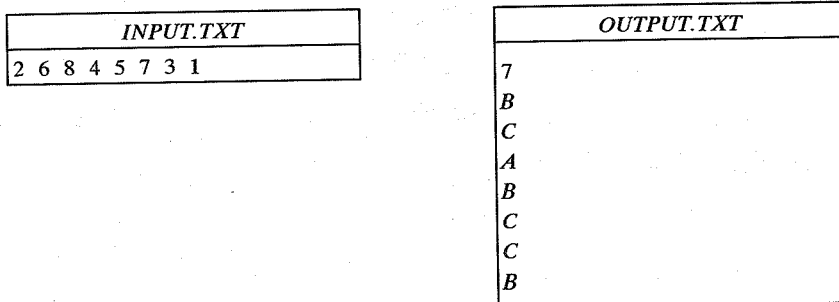
#### 輸出資料

你的程式必須寫入轉換序列的長度  $L$  到輸出檔 *OUTPUT.TXT* 的第一行中。在接下來的  $L$  行中，你的程式必須依序寫出基本轉換序列的字母代號，每一行的第一個位置放一個字母。

#### 工具

*MTOOL.EXE* 是放在任務目錄中的一個程式，它可以讓你玩魔術矩陣遊戲。執行 "*mtool input.txt output.txt*" 就可以進行目標盤面和轉換序列的實驗。

#### 輸入及輸出範例



圖三

