

# 一〇八學年度高級中學資訊學科能力競賽

## 彰雲嘉區複賽程式設計試題(二)

共 8 頁

### 七、矩陣之對角線 (10 分)

在如下 5\*5 矩陣中 AAAAA 所形成的線段，假設稱它為第一正對角線。  
兩條 BBBB 所形成的線段，假設稱它為第二正對角線  
兩條 CCC 所形成的線段，假設稱它為第三正對角線  
兩條 DD 所形成的線段，假設稱它為第四正對角線  
因 E 只有 1 個元素，不能稱為對角線

A	B	C	D	E
B	A	B	C	D
C	B	A	B	C
D	C	B	A	B
E	D	C	B	A

同樣在如下 5\*5 矩陣中 EEEEE 所形成的線段，假設稱它為第一反對角線。  
兩條 DDDD 所形成的線段，假設稱它為第二反對角線  
兩條 CCC 所形成的線段，假設稱它為第三反對角線  
兩條 BB 所形成的線段，假設稱它為第四反對角線  
同樣 A 不能稱為對角線

A	B	C	D	E
B	C	D	E	D
C	D	E	D	C
D	E	D	C	B
E	D	C	B	A

現要在一  $N \times N$  的矩陣中，找出在一正 OR 反對角線上之元素都相同的正反對角線數目；其中  $2 \leq N \leq 10000$

#### 輸入說明：

輸入資料中第一列為一整數  $m$ ，代表接下來有  $m$  組測試資料。

接下來每組測資的第一列有 1 個正整數，為  $N$  值。接下  $N$  列為  $N \times N$  矩陣內容。

#### 輸出說明：

請輸出符合條件之正反對角線數目

範例輸入：

2

5

A B C D E

B A B C D

C B A B C

D C C A B

E D C B A

4

A B C D

B A B C

C B A B

D C C A

範例輸出：

8

5

## 八、統計係數 (10 分)

列聯表卡方值的統計係數適用於兩個變項(A 與 B)中至少有一個變項是多分類資料(不只分為兩個類別)時，描述變項之間的相互關係，進而計算兩變項之間的品質相關係數或獨立性等。假設統計資料以表格表示，表格共有 I 列 J 行，第 i 列第 j 行的元素表示為  $f_{ij}$ 。下列表格為 I = 2, J = 3 (代表 A 變項有兩個類別，B 變項有 3 個類別)之輸入範例。

60	30	30
20	20	40

列聯表卡方值計算方法如下，首先由輸入資料中計算總人數  $N$ ，其計算公式為  $N = \sum_{i=1}^I \sum_{j=1}^J f_{ij}$ ；本例中， $N = \sum_{i=1}^I \sum_{j=1}^J f_{ij} = 200$ 。其次，計算每一列與每一行的總和，下表第一列之總和記為  $SA_1$  (120)，第一行之總和記為  $SB_1$  (80)，其餘依此類推。

$f_{ij}$	j=1	j=2	j=3	總和 (SA)
i=1	60	30	30	120
i=2	20	20	40	80
總和 (SB)	80	50	70	200

接著，再計算期望次數表  $E$ ，該表中第 i 列第 j 行的元素表示為  $E_{ij}$ ，計算公式為  $E_{ij} = \frac{SA_i SB_j}{N}$ ，如下表。 $E_{11} = \frac{SA_1 SB_1}{N} = \frac{120 \times 80}{200} = 48$ ，其餘依此類推，計算後的期望次數表  $E$  如下表。

$E_{ij}$	j=1	j=2	j=3
i=1	48	30	42
i=2	32	20	28

最後計算列聯卡方值，其公式為  $\chi^2 = \sum_{i=1}^I \sum_{j=1}^J \frac{(f_{ij} - E_{ij})^2}{E_{ij}}$ 。本範例中，

$$\begin{aligned} \chi^2 &= \sum_{i=1}^I \sum_{j=1}^J \frac{(f_{ij} - E_{ij})^2}{E_{ij}} = \frac{(60-48)^2}{48} + \frac{(30-30)^2}{30} + \frac{(30-42)^2}{42} + \frac{(20-32)^2}{32} + \frac{(20-20)^2}{20} + \frac{(40-28)^2}{28} \\ &= 3 + 0 + 3.43 + 4.50 + 0 + 5.14 = 16.07 \end{aligned}$$

設計一程式，輸入資料表，並計算出列聯表卡方值。

### 輸入說明：

輸入一行資料第一個數字代表 I 值，第二個數字代表 J 值，表格數字依列數由小至大將資料由左自右依序排列，數字間以一個空格分開。

**輸出說明：**

輸出計算卡方值至小數點下兩位，四捨五入。

**範例輸入一：**

2 3 60 30 30 20 20 40

**範例輸出一：**

16.07

**範例輸入二：**

2 4 7 33 130 30 6 13 60 21

**範例輸出二：**

3.17

## 九、計算服務需要的時間(10 分)

有一銀行業務的申請包括了  $K$  個步驟，且需依序進行才可以完成。 $K$  個步驟中的每個步驟由一個櫃檯人員負責處理。同一櫃檯同時只能服務一人，一旦開始服務某人，就必須繼續服務到這位申請人的步驟全部完成才可以換下一位。然而，每個申請人會因個人情況不同，在每個步驟都可能需要不同的處理時間。若有  $N$  個申請人排隊，他們必須依序完成  $K$  個步驟，且不可插隊或中途離開。

輸入每個人於每個步驟所需要的處理時間，請輸出完成全部的申請服務需要花多少時間；也就是從第一個人開始接受服務，到最後一位完成最後步驟，總共需要多少時間。

例如：有三個人排隊，服務包含四個步驟。第一個人  $A$  於各個步驟所需的處理時間依序為 7，3，5，8 分鐘。第二個人  $B$  於各個步驟所需的處理時間依序為 1，6，4，5 分鐘。第三個人  $C$  於各個步驟所需的處理時間依序為 6，2，7，3 分鐘。

一開始，只有第一個客戶  $A$  可以到第一櫃檯處理，需要 7 分鐘的服務時間。請參考下表。

第 8 分鐘開始時，第一個客戶  $A$  前進到第二櫃檯，需要 3 分鐘。而同時，第二個客戶  $B$  可以到第一櫃檯，需要 1 分鐘。

雖然第 8 分鐘結束時，第二個客戶  $B$  已經完成於第一櫃檯的服務，但是因為第一個客戶  $A$  仍在第二櫃檯接受服務，所以第二個客戶  $B$  必須於第二櫃檯前排隊，一直等到第 11 分鐘開始時，才可以接受第二櫃檯的服務，因為第一個客戶  $A$  在第二櫃檯，需要 3 分鐘的服務。

下表列出每個客戶在每個櫃檯接受服務的時間對應關係。注意：在第 16 分鐘時，第三櫃台是空閒的，原因是因為雖然  $A$  已經於第 15 分鐘完成在第三櫃台的服務，但是  $B$  卻必須在第 16 分鐘時停留在第二櫃檯繼續接受完服務。

在此範例中，總共的服務時間是 31 分鐘。

時 間	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
第一 櫃檯	A	A	A	A	A	A	A	B	C	C	C	C	C	C																	
第二 櫃檯								A	A	A	B	B	B	B	B	B	C	C													
第三 櫃檯											A	A	A	A	A		B	B	B	B	C	C	C	C	C	C	C				
第四 櫃檯																A	A	A	A	A	A	A	A	B	B	B	B	B	C	C	C

### 輸入說明：

第一行： $N$   $K$ ，其中  $N$  表示多少人， $K$  表示多少步驟

第二行： $K$  個數字，代表第一個人依序於各步驟所需的處理時間，兩時間以空白分隔

第三行： $K$  個數字，代表第二個人依序於各步驟所需的處理時間，兩時間以空白分隔

.....

第 N+1 行：K 個數字，代表第 N 個人依序於各步驟所需的處理時間，兩時間以空白分隔

**輸出說明：**

總共的服務時間

**範例輸入一：**

3 4

7 3 5 8

1 6 4 5

6 2 7 3

**範例輸出一：**

31

**範例輸入二：**

3 5

2 8 7 6 3

1 5 13 6 1

7 2 6 11 4

**範例輸出二：**

51

## 十、利用霍夫曼樹進行資料壓縮 (10 分)

霍夫曼樹(Huffman Tree)經常被用來作資料壓縮，將字串資料轉成霍夫曼碼(Huffman Code)以減少字串資料所佔的空間大小，其詳細作法如下：

- (1) 字串中所有的字母元素都放在最底層的葉節點上，節點中的值為此字母的出現次數。
- (2) 每次從尚無父節點的所有節點中，尋找值最小(出現次數最少)的兩個節點來建立霍夫曼樹，產生其父節點的值為兩個節點值相加的和。最後只剩下根節點沒有父節點。
- (3) 從根節點開始往下編碼，左分支為 0，右分支為 1，每個葉節點的字母元素會產生一個霍夫曼碼(由 0 和 1 所組成)，字母元素出現的頻率越高則霍夫曼碼越短。

舉例來說，若每個字元佔 1 byte=8 bits 的空間，則“queue”字串要 5 bytes=40 bits 空間，我們將“queue”字串壓縮成霍夫曼碼，其中字母 q 出現 1 次，u 出現 2 次，e 出現 2 次，逐步建立的霍夫曼樹如圖 1 所示，而霍夫曼碼的編碼結果如圖 2 所示。因此，原本 40 bits 的“queue”字串可以壓縮轉為 00011011 的霍夫曼碼，只需要 8 個 bits 就可以儲存，所以壓縮率可以達到  $\frac{40-8}{40} = 80\%$  (霍夫曼編碼的對應表格所需儲存之資料量相對於要壓縮的檔案資料量來說，一般會少了很多，可以忽略不計算)。

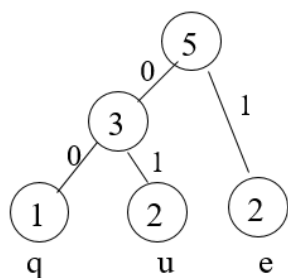


圖 1、建立霍夫曼樹

字母元素	霍夫曼碼
q	00
u	01
e	1

圖 2、霍夫曼碼編碼結果

### 輸入說明：

輸入一系列字串(長度最多為100個字元，且此字串中不含空格)，利用霍夫曼樹對此字串作資料壓縮，計算其壓縮率。

### 輸出說明：

輸出此字串壓縮前所佔的 bit 數(每個字元佔 8 bits)、壓縮後所需的 bit 數(霍夫曼碼的對應表格不計入)、以及壓縮率百分比(顯示格式為 n%，n 取整數，小數點無條件捨去)，以空白隔開。

### 範例輸入一：

queue

### 範例輸出一：

40 8 80%

範例輸入二：

abadcafcabfdec

範例輸出二：

112 35 68%

範例輸入三：

aaaaaaamcmhhhhhmkkkkkkkkmbbmdddddmyymppppp

範例輸出三：

352 132 62%

範例輸入四：

BCBBB

範例輸出四：

40 5 87%