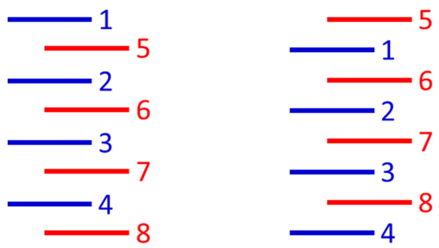
多堆鴿尾式外洗牌之最少次數探討

楊蕎卉 吳舜曄 曾巧閔 蘇柏奇* 游淑媛

苗栗縣立興華高級中學

壹、前言

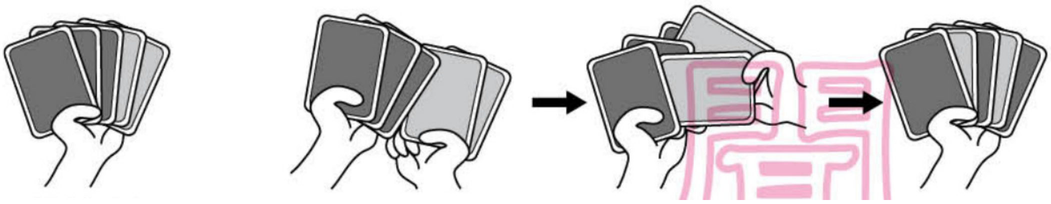
撲克牌的洗牌手法眾多，常見的鴿尾式洗牌(Riffle shuffle)為將一疊牌均分為兩疊，然後將兩疊牌一張張相互交錯形成新的排序，例如：手邊有 8 張牌時，由上而下依序編號為 1 到 8，原始排序為 12345678，將牌均分為上半疊 1234 及下半疊 5678，然後將兩疊牌交錯排列，依照最上面的牌來自於哪一疊，區分為外洗牌(out shuffle)與內洗牌(in shuffle)，所得新的排序分別為 15263748 及 51627384。以此方式重複洗牌多次，總會回到初始排序，例如本例經過 3 次外洗牌 12345678→15263748→13572468→12345678 回到初始排序；而經過 3 次內洗牌得到恰好顛倒的排序 12345678→51627384→75318642→87654321，故知經過 6 次內洗牌會回到初始排序。



外洗牌　　內洗牌

事實上，當所分兩疊牌數不相等時，也能進行鴿尾式洗牌。例如 102 年度基本學力測驗數學科試題第 28 題提到了一個撲克牌洗牌次數問題，將手上 5 張牌依顏色分成兩堆，將右手之牌依序插入左手之牌的間隔之中形成新的排序，此種洗牌方法即為鴿尾式外洗牌。不難得知，若右手之牌數多於左手之牌數，則經過一次鴿尾式外洗牌後，部分右手之牌的位置不會改變，例如兩堆分別有 3、6 張，排序為 123456789，經過一次洗牌後得到 142536789，編號 6, 7, 8, 9 的牌之位置皆未變動，即與兩堆分別有 3、2 張的情形相同，故僅需討論左手牌數大於右手牌數的情形。

* 為本文通訊作者



(圖形引自 102 年度基本學力測驗數學科試題)

貳、文獻探討

林建維、陳奕均、吳彥澄等人探討此洗牌問題，他們觀察牌位置變化的規律，發現除了左 1 之牌從未變動外，其它的位置之牌有循環的現象，例如左、右分別有 4、2 張時，左 1 到左 4 的牌為 1234，右 1 到右 2 的牌為 AB，洗牌 6 次回到初始排序，洗牌過程中左 2、左 3、右 1 三個位置的牌為一個循環（每次洗牌時，左 2、左 3、右 1 之牌分別移動至左 3、右 1、左 2），每經過 3 次洗牌，此三張牌會回到初始位置；左 4、右 2 則為另一個循環，每經過 2 次洗牌，此兩張牌會回到初始位置。故經過 2 與 3 的最小公倍數 6 次洗牌可回到初始排序。事實上，當左手偶數張牌時，都會形成兩個循環；左手有奇數張牌時，所有位置之牌形成一個循環，例如左手 3 張牌時，每次洗牌時，左 2、左 3、右 2、右 1 之牌分別移動至左 3、右 2、右 1、左 2。

位置／次數	左 1	左 2	左 3	左 4	右 1	右 2
初始排序	1	2	3	4	A	B
1	1	A	2	B	3	4
2	1	3	A	4	2	B
3	1	2	3	B	A	4
4	1	A	2	4	3	B
5	1	3	A	B	2	4
6	1	2	3	4	A	B

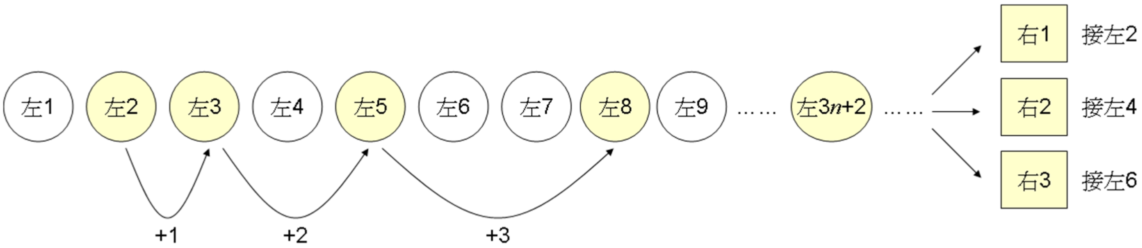
循環		
左 2	左 3	右 1
2	3	A
A	2	3
3	A	2
2	3	A
A	2	3
3	A	2
2	3	A

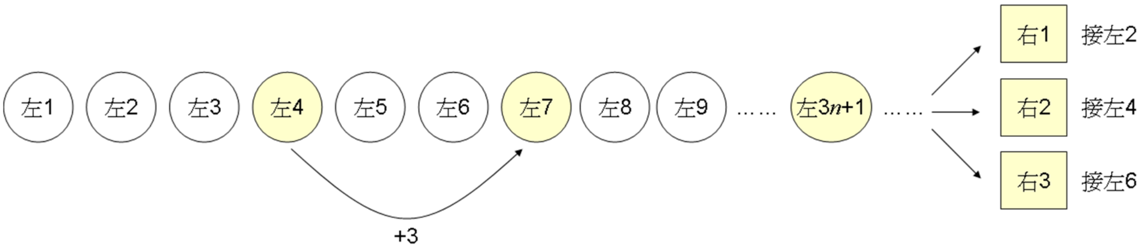
循環	
左 4	右 2
4	B
B	4
4	B
B	4
4	B
B	4
4	B

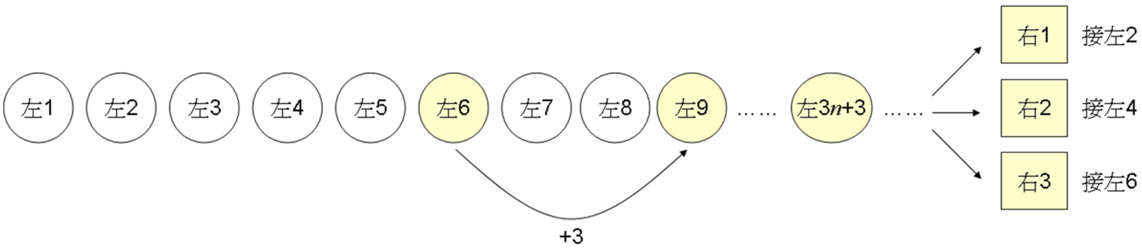
位置／次數	左 1	左 2	左 3	右 1	右 2
初始排序	1	2	3	A	B
1	1	A	2	B	3
2	1	B	A	3	2
3	1	3	B	2	A
4	1	2	3	A	B

循環			
左 2	左 3	右 2	右 1
2	3	B	A
A	2	3	B
B	A	2	3
3	B	A	2
2	3	B	A

林建維、陳奕均、吳彥澄等人接著探討右手有 3 張牌的情形，他們發現左 2 之前將被插入一張牌，故左 2 的牌移動到左 3、其餘左手的位置（除了左 1 和左 2）之前將被插入兩張牌，故右移兩個位置，而右 1 的牌移動到左 2、右 2 的牌移動到左 4。他們依照位置的變化將位置區分為以左 2、左 4、左 6 為段首，右 1、右 2、右 3 為段尾的三段（依照左手牌之數量，決定各段之尾為何，詳如下表），接著根據右 1 的牌移動到左 2、右 2 的牌移動到左 4、右 3 的牌移動到左 6 的規則來決定哪些段會組成循環，最後再由各循環之位置個數的最小公倍數得回到初始排序的最少洗牌次數。而這樣的方法，也能運用至右手有更多張牌的情形。







（圖形引自科學教育月刊第 372 期）

編號	段首位置	段尾位置		
		左手 $3k$ 張	左手 $3k+1$ 張	左手 $3k+2$ 張
1	左 2	右 2	右 1	右 3
2	左 4	右 1	右 3	右 2
3	左 6	右 3	右 2	右 1

相較於林建維、陳奕均、吳彥澄等人觀察位置變化來解題，黃莉芸、邱文均以牌的編號來取代位置的編號，因左 1 位置上的牌從未變動，故編為 0，其餘的牌自左 2 開始，依序編號為 1, 2, 3, ⋯，當左、右手分別有 3, 2 張時，初始排序記為 01234，經過一次洗牌變成 03142，可看出每次洗牌即為各張牌的重新排序，故可用抽象代數之排列 permutation 來表示洗牌過程，當左、右手分別有 n_1, n_2 張時，以σ_{n_1,n_2} 來表示洗牌操作，回到初始位置的最少洗牌次數記為$ord(\sigma_{n_1,n_2})$。其中編號 0 的牌從未變動位置，方便起見有時予以省略。

排列有兩種表示法，其一為利用矩陣符號將自然排序寫在第一列，並將排列後的排序寫在第二列，如 $\sigma_{3,2} = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 1 & 4 & 2 \end{pmatrix}$；第二種表示法稱為輪換分解，將$\sigma_{n_1,n_2}$ 表示為若干個 cycle 的合成，事實上，cycle 內的牌號可對應到林建維等人之位置編號，cycle 的長度相當於林建維等人之循環內的位置個數，故每個 cycle 長度的最小公倍數即為將牌洗回初始排序的最少洗牌次數。例如$\sigma_{3,2}$ 將 $1 \mapsto 3$，先得$(1\quad 3$, 接著$\sigma_{3,2}$ 將 $3 \mapsto 4$, 續得$(1\quad 3\quad 4$, 再接著$\sigma_{3,2}$ 將 $4 \mapsto 2$，得$(1\quad 3\quad 4\quad 2$, 最後$\sigma_{3,2}$ 將 $2 \mapsto 1$，故得$(1\quad 3\quad 4\quad 2)$，這個 cycle 的長度為 4，經過 4 次洗牌，每張牌回到初始位置，即 $ord(\sigma_{3,2}) = 4$。再例如 $\sigma_{4,2} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 4 & 1 & 5 & 2 & 3 \end{pmatrix}$先由 $1 \mapsto 4, 4 \mapsto 2, 2 \mapsto 1$ 得 $(1\quad 4\quad 2)$，且 $3 \mapsto 5, 5 \mapsto 3$, 得 $(3\quad 5)$，即得 $\sigma_{4,2} = (1\quad 4\quad 2)\ (3\quad 5)$，這兩個 cycle 的長度分別為 3, 2，經過 $lcm(3,2) = 6$ 次洗牌，每張牌回到初始位置，即$ord(\sigma_{4,2}) = 6$。

黃莉芸、邱文均觀察到當 $i \geq 6$ 時，$\sigma_{n_1,3}$ 將$i \mapsto i-3$, 故 cycle 中存在若干個公差為 -3 的等差數列，例如：

$$\begin{aligned}\sigma_{13,3} &= \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 \\ 13 & 1 & 14 & 2 & 15 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \end{pmatrix} \\ &= (1,13,10,7,4,2)(3,14,11,8,5,15,12,9,6)\end{aligned}$$

進而將牌區分為三個「圈數列」(註：相當於林建維等人的分段)，分法如下：先以右手的牌 13, 14, 15 為三個圈數列的第二項，由 $1 \mapsto 13$, $3 \mapsto 14$, $5 \mapsto 15$ 得三個首項分別為 1, 3, 5，以首項為標記將此三個圈數列分別記為$\langle C_{3,1}\rangle$、$\langle C_{3,3}\rangle$和$\langle C_{3,5}\rangle$，接著由 $13 \mapsto 10$, $14 \mapsto 11$, $15 \mapsto 12$ 得第 3 項分別為 10, 11, 12；同理再得第 4 項分別為 7, 8, 9；再得第 5 項分別為 4, 5, 6，但 5 為$\langle C_{3,5}\rangle$的首項，故$\langle C_{3,3}\rangle$的末項為 8、末項對應數為 5；再得第$\langle C_{3,1}\rangle$、$\langle C_{3,5}\rangle$

的第 6 項分別為 2, 3，同理<$C_{3,5}$>的末項為 6、末項對應數為 3；再得<$C_{3,1}$>的第 7 項為 1，同理<$C_{3,1}$>的末項為 2、末項對應數為 1。將圈數列末項對應的數以括號列於圈數列末項之後，得三個圈數列如下：

$$\begin{cases} <C_{3,1}>= & 1, & 13, & 10, & 7, & 4, & 2, & (1) \\ <C_{3,3}>= & 3, & 14, & 11, & 8, & (5) \\ <C_{3,5}>= & 5, & 15, & 12, & 9, & 6, & (3) \end{cases}$$

進一步定義 $b\sigma_{x,y}$，僅列出各圈數列首項及末項所對應的數來簡化圈數列接續的問題，因 $\sigma_{13,3}$ 之圈數列首項依序為 1, 3, 5，末項對應的數依序為 1, 5, 3，得 $b\sigma_{13,3}=\begin{pmatrix} 1 & 3 & 5 \\ 1 & 5 & 3 \end{pmatrix}=(1)(3,5)$，由此即可知<$C_{3,1}$>形成一個圈，<$C_{3,3}$>和<$C_{3,5}$>接續成一個圈。

當右手有 n_2 張時，則可分成 n_2 個圈數列，原始問題「每個 cycle 的長度為何？」轉化為「每個圈數列有幾項？」及「圈數列如何接續？（即求 $b\sigma_{n_1,n_2}$ ）」，黃莉芸、邱文均提供求解方法如下：

【文獻結論 1】（黃莉芸、邱文均[5]，定理 6）

當 $n_1=kn_2$ 時，令 $n_2+i-1=c_i\times 2^{\alpha_i}$, i=1~n_2 則

$b\sigma_{n_1,n_2}=\begin{pmatrix} 1 & 3 & 5 & \ldots & 2i-1 & \ldots & 2n_2-1 \\ c_1 & c_2 & c_3 & \ldots & c_i & \ldots & c_{n_2} \end{pmatrix}$，且 $<C_{n_2,2i-1}>$ 有 $k+\alpha_i$ 項。

以求 $ord(\sigma_{7k,7})$ 為例，說明如下：

由 $n_2=7$ 得 $6+i=c_i\times 2^{\alpha_i}$ ，即得 $\begin{cases}(c_1,c_2,\ldots,c_7)=(7,1,9,5,11,3,13) \\ (\alpha_1,\alpha_2,\ldots,\alpha_7)=(0,3,0,1,0,2,0)\end{cases}$,

再得 $b\sigma_{7k,7}=\begin{pmatrix} 1 & 3 & 5 & 7 & 9 & 11 & 13 \\ 7 & 1 & 9 & 5 & 11 & 3 & 13 \end{pmatrix}$=(1,7,5,9,11,3)(13)，

且七個圈數列依序有 k, k+3, k, k+1, k, k+2, k 項，

由(1,7,5,9,11,3)得其中一個 cycle 有 k+(k+1)+ k + k + (k+2) + (k+3) = 6k+6 項

由(13)得另一個 cycle 有 k 項，故得 $ord(\sigma_{7k,7})=lcm(6k+6,k)$.

【文獻結論 2】（黃莉芸、邱文均[5]，定理 7）

若 $n_1 = kn_2 + d$ (d=1~$n_2 - 1$)，令 $\begin{cases} n_2 + d + i - 1 = c_i \times 2^{\alpha_i} \ (\text{其中} 1 \le i < n_2 - d + 1) \\ i + d - 1 = c_i \times 2^{\alpha_i} \ (\text{其中} n_2 - d + 1 \le i \le n_2) \end{cases}$，則：

(1) $b\sigma_{n_1,n_2} = \begin{pmatrix} 1 & 3 & 5 & \ldots & 2i-1 & \ldots & 2n_2 - 1 \\ c_1 & c_2 & c_3 & \ldots & c_i & \ldots & c_{n_2} \end{pmatrix}$。

(2) $< C_{n_2, 2i-1} >$ 項數為 $\begin{cases} k + \alpha_i \ (1 \le i < n_2 - d + 1) \\ k + \alpha_i + 1 (n_2 - d + 1 \le i \le n_2) \end{cases}$。

以求 $ord(\sigma_{7k+3,7})$ 為例，說明如下：

由 $d = 3$，$n_2 = 7$ 得 $\begin{cases} 9 + i = c_i \times 2^{\alpha_i} \ (1 \le i < 5) \\ i + 2 = c_i \times 2^{\alpha_i} \ (5 \le i \le 7) \end{cases}$，即得

$\begin{cases} (c_1, c_2, \ldots, c_7) = (5,11,3,13,7,1,9) \\ (\alpha_1, \alpha_2, \ldots, \alpha_7) = (1,0,2,0,0,3,0) \end{cases}$，

再得 $b\sigma_{7k+3,7} = \begin{pmatrix} 1 & 3 & 5 & 7 & 9 & 11 & 13 \\ 5 & 11 & 3 & 13 & 7 & 1 & 9 \end{pmatrix}$=(1,5,3,11) (7,13,9)，

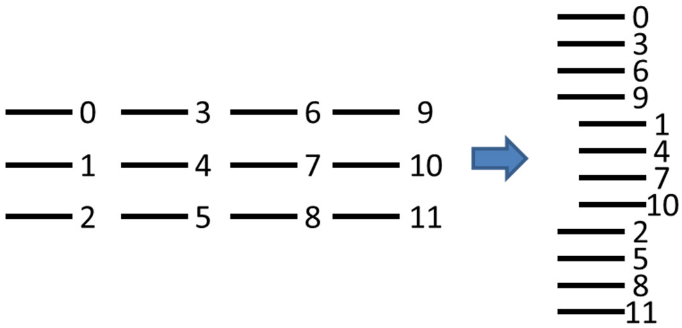
且七個圈數列依序有 k+1, k, k+2, k, k+1, k+4, k +1 項，

由(1,5,3,11) 得其中一個 cycle 有(k+1)+(k+2)+ k + (k+4) = 4k+7 項，

由(7,13,9) 得另一個 cycle 有 k + (k+1) + (k+1) = 3k+2 項，

故得 $ord(\sigma_{7k+3,7}) = lcm(4k+7, 3k+2).$

若增加牌的堆數以推廣上面的問題，首先面臨的是如何洗牌？我們發現 1994 年 P. Glaister 提到將 n 張牌均分成 m 堆的多堆洗牌程序，例如：當 12 張牌時，原始排序為 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 ,11，洗牌時先均分為 4 堆，依序抽出每堆第一張 0, 3, 6, 9，再依序抽出每堆第二張 1, 4, 7, 10，再依序抽出每堆第三張 2, 5, 8, 11，故經過一次洗牌後，變成 0, 3, 6, 9, 1, 4, 7, 10, 2, 5, 8, 11。P. Glaister 提到回到初始排序的最少洗牌次數為滿足同餘式 $m^k \equiv 1 (\bmod \ n-1)$ 的最小整數 k。（註：同餘的符號為≡，當兩個整數 a, b 除以同一個正整數 c，若得相同餘數，則稱 a, b 對於模 c 同餘，記為 $a \equiv b (\bmod\ c)$。）

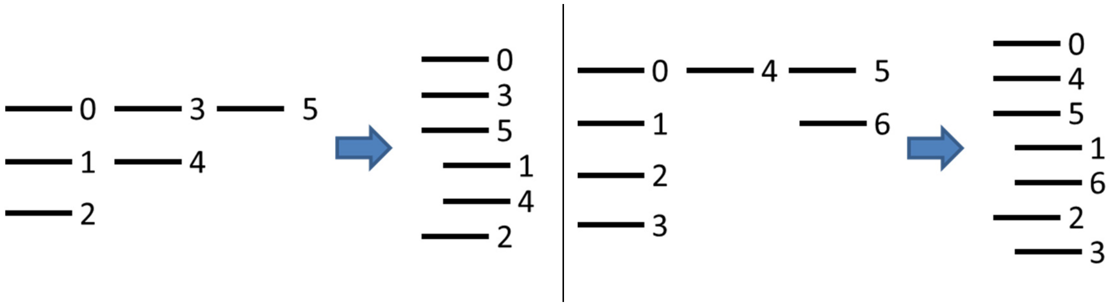


【文獻結論 3】P. Glaister

若將 n 張牌等分成 m 堆，

則回到原來次序所需的最少向下洗牌次數為滿足 $m^k \equiv 1 (\bmod\ n-1)$ 的最小整數 k.

本研究探討的洗牌程序為：當三堆分別有 3、2、1 張時，初始排序為 0, 1, 2, 3, 4, 5，依序抽出每堆第一張 0,3,5，再依序抽出每堆第二張 1,4，再依序抽出每堆第三張 2，故經過一次洗牌後成：0, 3, 5, 1, 4, 2。再例如三堆分別有 4、1、2 張時，初始排序為 0, 1, 2, 3, 4, 5, 6，經過一次洗牌後成 0, 4, 5, 1, 6, 2, 3。我們將這樣的洗牌方式稱為多堆鴿尾式外洗牌，延續並修正黃莉芸、邱文均所定義的工具來探討最少經過多少次這樣的洗牌程序才能回到初始排序。



參、多堆洗牌的最少洗牌次數

當有 h 堆牌，各堆依序有 $n_1, n_2, .., n_h$ (其中 $n_1 \geq \max\{n_2, n_3, ..., n_h\}$)張時，由左而右將牌依序編號為 0, 1, 2,⋯, $\sum_{i=1}^{h} n_i - 1$。洗牌過程中，會陸續遇到某些堆之牌已抽完的狀況，例如 $\sigma_{15,5,2}$ 可區分為三堆皆有牌（如紅色底線）、第一、二堆有牌（如藍色底線）、只有第一堆有牌（如綠色底線）等三個階段。又 $\sigma_{15,5,2}$ =(1, 15, 8, 3)(2, 20, 13, 6)(4, 16, 9, 18, 11,

19, 12, 5, 21, 14, 7, 17, 10)，故 $ord(\sigma_{15,5,2})=lcm(4,4,13)=52$ ，我們將利用圈數列等工具來求最少洗牌次數。

$$\sigma_{15,5,2}=\begin{pmatrix} \overbrace{0\ \ 1\ \ 2\ \ 3\ \ 4\ \ 5\ \ 6\ \ 7\ \ 8\ \ 9\ \ 10\ \ 11\ \ 12\ \ 13\ \ 14,}^{\text{第1堆}} & \overbrace{15\ \ 16\ \ 17\ \ 18\ \ 19,}^{\text{第2堆}} & \overbrace{20\ \ 21}^{\text{第3堆}} \\ 0\ \ 15\ \ 20\ \ 1\ \ 16\ \ 21\ \ 2\ \ 17\ \ 3\ \ 18\ \ 4\ \ 19\ \ 5\ \ 6\ \ 7, & 8\ \ 9\ \ 10\ \ 11\ \ 12, & 13\ \ 14 \end{pmatrix}$$

討論一般的情形，根據 $\sigma_{n_1,n_2,\ldots,n_h}$ ，以第 2 堆到第 h 堆的牌為第二項，定義 $s=\sum_{i=2}^{h} n_i$ 個圈數列（各個圈數列依照第二項之大小排序）及 $b\sigma_{n_1,n_2,\ldots,n_h}$ 如下：

【定義 1】$\sigma_{n_1,n_2,\ldots,n_h}$ 的 s 個圈數列及 $b\sigma_{n_1,n_2,\ldots,n_h}$

1. $<C_i>=\ c_{i,1},\ c_{i,2},\ \ldots,\ c_{i,\alpha_i},\ (b_i)$, 其中 $c_{i,2}=n_1+i-1$, i =1, 2,…, s
2. $b\sigma_{n_1,n_2,\ldots,n_h}=\begin{pmatrix} c_{1,1} & c_{2,1} & c_{3,1} & \cdots & c_{s,1} \\ b_1 & b_2 & b_3 & \cdots & b_s \end{pmatrix}$

例如： $\sigma_{15,5,2}$的 7 個圈數列及 $b\sigma_{15,5,2}$ 如下：

$$\begin{cases} <C_1>= & 1, & 15, & 8, & 3, & (1) \\ <C_2>= & 4, & 16, & (9) & & \\ <C_3>= & 7, & 17, & 10, & (4) & \\ <C_4>= & 9, & 18, & (11) & & \\ <C_5>= & 11, & 19, & 12, & (5) & \\ <C_6>= & 2, & 20, & 13, & 6, & (2) \\ <C_7>= & 5, & 21, & 14, & (7) & \end{cases} ;$$

$$b\sigma_{15,5,2}=\begin{pmatrix} 1 & 4 & 7 & 9 & 11 & 2 & 5 \\ 1 & 9 & 4 & 11 & 5 & 2 & 7 \end{pmatrix}=(1)(4,9,11,5,7)(2).$$

若能快速求得各圈數列之 $c_{i,1}$ 及 b_i 即可得 $b\sigma_{n_1,n_2,\ldots,n_h}$，再由 α_i 即可得每個 cycle 的長度，進而得到最少洗牌次數。以下將探討如何求得 $c_{i,1}$ 、 b_i 及 α_i 。

令 $n_1=sk+d$ ， $0\le d<s$ ，並將 $n_2,n_3,\ldots,n_h$ 由大而小排序為 $m_1,m_2,\ldots,m_{h-1}$ (即 $m_i\ge m_{i+1}$)，則經過抽出每堆的前 m_1 張牌後（第 2~h 堆牌皆已抽完，共抽出 $s+m_1$ 張牌），

只剩下第一堆的牌之編號為 $m_1, m_1+1, m_1+2, \ldots$，即 $\sigma = \begin{pmatrix} \ldots & s+m_1 & s+m_1+1 & s+m_1+2 & \ldots \\ \ldots & m_1 & m_1+1 & m_1+2 & \ldots \end{pmatrix}$，因此得到當 $i \ge s+m_1$ 時，$\sigma_{n_1,n_2,\ldots,n_h}$ 將 $i \,\alpha\, \ i-s$，故每個圈數列中皆有一段公差為 $-s$ 的等差數列（即 $c_{i,2}, c_{i,3}, \ldots, c_{i,k}$ 為等差數列），由 $c_{i,2} = n_1 + i - 1$ 得 $c_{i,k} = c_{i,2} - (k-2)s = 2s + d + i - 1$. 以下將由 $c_{i,2}$ 得 $c_{i,1}$，由 $c_{i,k}$ 得 c_{i,α_i}，α_i 及 b_i，以三堆為例，說明如下：

1. 由 $c_{i,2}$ 求 $c_{i,1}$：

為方便起見，定義 $\langle A \rangle = \ c_{1,1},\ c_{2,1},\ c_{3,1},\ \ldots,\ c_{s,1}$，當 $n_2 > n_3$ 時：

$$\sigma = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & \ldots & 3n_3 & 3n_3+1 & 3n_3+2 & 3n_3+3 & \ldots & \ldots \\ 0 & n_1 & n_1+n_2 & 1 & n_1+1 & n_1+n_2+1 & \ldots & n_3 & n_1+n_3 & n_3+1 & n_1+n_3+1 & \ldots & \ldots \end{pmatrix}$$

由 $1 \to n_1,\ 4 \to n_1+1,\ \ldots,\ 3n_3+1 \to n_1+n_3+1, \ldots$ 及 $2 \to n_1+n_2,\ 5 \to n_1+n_2+1, \ldots$

得 $\langle A \rangle = \underbrace{1,\ 4,\ \ldots, 3n_3-2}_{n_3\text{ 項，公差 }3}, \underbrace{3n_3+1,\ 3n_3+3, \ldots, 2n_2+n_3-1}_{n_2-n_3\text{ 項，公差 }2}, \underbrace{2,\ 5, \ldots, 3n_3-1}_{n_3\text{ 項，公差 }3}$

同理，$n_2 = n_3$ 時，$\langle A \rangle = \underbrace{1,\ 4,\ \ldots,\ 3n_2-2}_{n_2\text{ 項，公差 }3}, \underbrace{2,\ 5,\ \ldots,\ 3n_2-1}_{n_2\text{ 項，公差 }3};$

$n_2 < n_3$ 時，

$\langle A \rangle = \underbrace{1,\ 4,\ \ldots,\ 3n_2-2}_{n_2\text{ 項，公差 }3}, \underbrace{2,\ 5,\ \ldots,\ 3n_2-1}_{n_2\text{ 項，公差 }3}, \underbrace{3n_2+1,\ \ldots,\ 2n_2+n_3-1}_{n_3-n_2\text{ 項，公差 }2}$

2. 由 $c_{i,k}$ 求 b_i：

當 $j \ge k$ 時，$c_{i,j}\ \alpha\ \ c_{i,j+1}$ 之 $c_{i,j+1}$ 必為第一堆的牌，以 "－" 註記 σ 中第二、三堆的牌如下：

$$\sigma = \begin{pmatrix} 0 & - & - & 3 & - & - & \ldots & 3m_2 & - & 3m_2+2 & - & \ldots & 2m_1+m_2 & 2m_1+m_2+1 & \ldots \\ 0 & - & - & 1 & - & - & \ldots & m_2 & - & m_2+1 & - & \ldots & m_1 & m_1+1 & \ldots \end{pmatrix}$$

得遞迴式 $c_{i,j+1}=\begin{cases}\dfrac{c_{i,j}}{3}, & c_{i,j}\le 3m_2-3\\ \dfrac{c_{i,j}-m_2}{2}, & 3m_2\le c_{i,j}\le m_2+2m_1-2\\ c_{i,j}-s, & c_{i,j}\ge m_2+2m_1\end{cases}$,

反覆利用遞迴式依序得 $c_{i,k+1}$，$c_{i,k+2}$，…，c_{i,α_i}，b_i．方便起見，定義集合 $A=\{c_{i,1}\mid 1\le i\le s\}$。由 $c_{i,t}$ 算得 $c_{i,t+1}$ 時，若 $c_{i,t+1}\in A$，則 $\alpha_i=t$ 且 $b_i=c_{i,t+1}$。

例如：求 $ord(\sigma_{8k,6,2})=?$

由 $c_{i,2}=n_1+i-1$ 得 $\begin{cases}<A>=1,4,7,9,11,13,2,5\\ A=\{1,4,7,9,11,13,2,5\}\end{cases}$

由 $c_{i,k}=2s+d+i-1$ 得 $\left(c_{1,k},c_{2,k},\ldots,c_{8,k}\right)=(16,17,18,19,20,21,22,23)$

再由遞迴關係 $C_{i,j+1}=\begin{cases}\frac{C_{i,j}}{3}, & C_{i,j}\le 3\\ \frac{C_{i,j}-2}{2}, & 6\le C_{i,j}\le 12\\ C_{i,j}-8, & C_{i,j}\ge 14\end{cases}$ 得 $\begin{cases}6\mapsto 8\mapsto 3\mapsto (1)\\ 17\mapsto (9)\\ 18\mapsto 10\mapsto (4)\\ 19\mapsto (11)\\ 20\mapsto 12\mapsto (5)\\ 21\mapsto (13)\\ 22\mapsto 14\mapsto 6\mapsto (2)\\ 23\mapsto 15\mapsto (7)\end{cases}$，

即得 $\begin{cases}\left(b_1,b_2,b_3,b_4,b_5,b_6,b_7,b_8\right)=(1,9,4,11,5,13,2,7)\\ \left(\alpha_1,\alpha_2,\alpha_3,\alpha_4,\alpha_5,\alpha_6,\alpha_7,\alpha_8\right)=(k+2,k.k+1,k,k+1,k,k+2,k+1)\end{cases}$

且得 $b\sigma_{8k,6,2}=\begin{pmatrix}1 & 4 & 7 & 9 & 11 & 13 & 2 & 5\\ 1 & 9 & 4 & 11 & 5 & 13 & 2 & 7\end{pmatrix}=(1)(4,9,11,5,7)(13)(2)$

故得

$ord(\sigma_{8k,6,2})=lcm(\alpha_1,\alpha_2+\alpha_4+\alpha_5+\alpha_8+\alpha_3,\alpha_6,\alpha_7)=lcm(k+2,5k+3,k)$。

肆、同堆同色之最少洗牌次數

基測題目中，牌分成兩種顏色（同一堆的顏色相同），僅需將牌洗成兩邊顏色相同即

可（不須每張牌回到原來的次序）。先考慮三堆分別有 n_1、n_2、n_3 張牌的情形，以 1, 2, 3 來表示三種顏色，但仍標記該張牌初始位置，將牌編號為 $1_0,1_1,\ldots 1_{n_1-1},2_{n_1},\ldots,2_{n_1+n_2-1},3_{n_1+n_2-1},\ldots,3_{n_1+n_2+n_3-1}$，定義將牌洗回同堆的牌皆同色的最少洗牌次數為 $cord(\sigma_{n_1,n_2,n_3})$.

三堆牌之中，只要兩堆同時洗回同色，第三堆也必定都同色，我們選擇觀察第二、三堆牌同時洗回同色的次數。先觀察 $ord(\sigma_{4k,3,1})$ 與 $cord(\sigma_{4k,3,1})$ 的關係：

1. 由 $\sigma_{4k,3,1}=(\underbrace{1_1,2_{4k},\ldots,}_{k\text{項}}\ \underbrace{1_4,2_{4k+1},\ldots,}_{k+1\text{ 項}}\underbrace{3_{4k+3},\ldots,1_3}_{k+2\text{ 項}})(\underbrace{1_6,2_{4k+2},\ldots,1_{10}}_{k\text{項}})$，

   得 $ord(\sigma_{4k,3,1})=lcm(3k+3,k)$

2. 只考慮顏色，得 $\sigma_{4k,3,1}=(\underbrace{1,2,1\ldots,}_{k\text{項}}\ \underbrace{1,2,1\ldots,}_{k+1\text{ 項}}\underbrace{1,3,1\ldots,1}_{k+2\text{ 項}})(\underbrace{1,2,1\ldots,1}_{k\text{項}})$

   又因 $\left(a_1,a_2,a_3,\ldots,a_{n-1},a_n\right)=\left(a_2,a_3,\ldots,a_{n-1},a_n,a_1\right)$

   故 $\sigma_{4k,3,1}=(\underbrace{2,1\ldots,1}_{k\text{項}}\ \underbrace{2,1\ldots,1}_{k+1\text{ 項}}\underbrace{3,1\ldots,1}_{k+2\text{ 項}})(\underbrace{1,2,1\ldots,1}_{k\text{項}})$

   經過 3k+3 次移動，$(\underbrace{2,1\ldots,1}_{k\text{項}}\ \underbrace{2,1\ldots,1}_{k+1\text{ 項}}\underbrace{3,1\ldots,1}_{k+2\text{ 項}})$中的第二、三堆的牌回到第二、三堆；經過 k 次移動，$(\underbrace{2,1\ldots,1}_{k\text{項}})$中的第二堆的牌回到第二堆，故得 $cord(\sigma_{4k,3,1})=lcm(3k+3,k)$.

上述 $cord(\sigma_{4k,3,1})=ord(\sigma_{4k,3,1})$ 代表僅考慮顏色與每張牌都回到初始位置的洗牌次數相同。

若 cycle 包含 n 張同堆的牌（不妨設為第二堆），且 n 個圈數列分別有 $x_1,x_2,\ldots,x_n$ 項，即 $(\underbrace{2,\ldots,}_{x_1\text{項}}\underbrace{2,\ldots,}_{x_2\text{項}}\ldots\underbrace{2,\ldots}_{x_n\text{項}})$，每張牌回到原來位置的洗牌次數為 $\sum_1^n x_i$，當 $x_1=x_2=\ldots=x_n$ 時，只考慮顏色的洗牌次數為 $\frac{1}{n}\sum_1^n x_i$；而當 x_i, i=1~n 不完全相等時，只考慮顏色的洗牌次數仍為 $\sum_1^n x_i$。以下將藉由證明 x_i, i=1~n 不完全相等來驗證 $cord(\sigma_{n_1,n_2,\ldots,n_h})=ord(\sigma_{n_1,n_2,\ldots,n_h})$。

首先，考慮皆有 α 項之圈數列$< C_i >$、$< C_j >$，$i>j$ 接續成一個 cycle，討論如下：

1. 由$i>j$，得當$1\le t\le\alpha$時，$c_{i,t}>c_{j,t}$且$b_i>b_j$。

2. 由$< C_i >$、$< C_j >$接續成一個 cycle，反推得$b\sigma=\begin{pmatrix} \dots & c_{j,1} & \dots & c_{i,1} & \dots \\ \dots & b_j & \dots & b_i & \dots \end{pmatrix}$之$c_{i,1}=b_j$
   且$c_{j,1}=b_i$，又因$c_{i,1}>c_{j,1}$，則得$b_j=c_{i,1}>c_{j,1}=b_i$，即得$b_j>b_i$。

上述所得b_i，b_j之大小關係產生矛盾，故必不存在兩圈項數相等之圈數列接續成 cycle 的情形。同理可得不存在 n 個項數相同的圈數列接續成一個 cycle 的情形，得以下結果：

【定理 1】 $cord(\sigma_{n_1,n_2,\dots,n_h})=ord(\sigma_{n_1,n_2,\dots,n_h})$.

伍、特殊情況的最少洗牌次數

本節討論特定條件下的最少洗牌次數。

一、每堆牌的數量相等

文獻結論 3 的同餘式$m^k\equiv1(\bmod\ n-1)$使我們聯想到歐拉定理：「若 gcd(a, n)=1，則$a^{\varphi(n)}\equiv1(\bmod n)$」，因 h 與 $hn-1$ 互質，得 $h^{\varphi(hn-1)}\equiv1\ (\bmod\ hn-1)$，進一步得$ord(\sigma_{\underbrace{n,n,\dots,n}_{h個}})|\varphi(hn-1)$。例如：$ord(\sigma_{3,3,3})\,|\,\varphi(8)$且$\varphi(8)=4$，故$ord(\sigma_{3,3,3})=2\ or\ 4$；再例如：$ord(\sigma_{4,4,4,4})\,|\,\varphi(16)$且$\varphi(16)=8$，即$ord(\sigma_{4,4,4,4})=2\ or\ 4\ or\ 8$。

（註：對正整數 n，歐拉函數$\varphi(n)$是小於或等於 n 的正整數中與 n 互質的數的數目，例如$\varphi(5)=4$。另外，若$n=p_1^{\alpha_1}\times p_2^{\alpha_2}\times\dots\times p_r^{\alpha_r}$，則
$$\varphi(n)=n\times(1-\frac{1}{p_1})\times(1-\frac{1}{p_2})\times\dots\times(1-\frac{1}{p_r}).$$）

當每堆皆有h^t張牌時，滿足$h^k-1\equiv0(\bmod h^{t+1}-1)$的最小整數$k=t+1$，即得

$ord(\sigma_{\underbrace{h^t,h^t,h^t,\ldots,h^t}_{h\text{個}}}) = t+1$。

例如：$ord(\sigma_{3,3,3}) = ord(\sigma_{4,4,4,4}) = 2$, $ord(\sigma_{9,9,9}) = ord(\sigma_{16,16,16,16}) = 3$。

【定理 2】　1. $ord(\sigma_{\underbrace{n,n,\ldots,n}_{h\text{個}}}) \mid \varphi(hn-1)$.　　2. $ord(\sigma_{\underbrace{h^t,h^t,h^t,\ldots,h^t}_{h\text{個}}}) = t+1$.

二、每堆牌的數量不完全相等

先討論 $\sigma_{n+1,n,n}$，

由 $\sigma_{6,5,5} = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 \\ 0 & 6 & 11 & 1 & 7 & 12 & 2 & 8 & 13 & 3 & 9 & 14 & 4 & 10 & 15 & 5 \end{pmatrix}$ 以第 2、3 堆的牌為第二項，定義 10 個圈數列：

$$\begin{cases} <C_1>= & 1, & (6) \\ <C_2>= & 4, & (7) \\ <C_3>= & 7, & (8) \\ <C_4>= & 10, & 9, & 3, & (1) \\ <C_5>= & 13, & (10) \\ <C_6>= & 2, & (11) \\ <C_7>= & 5, & 12, & (4) \\ <C_8>= & 8, & (13) \\ <C_9>= & 11, & (14) \\ <C_{10}>= & 14, & 15, & (5) \end{cases}$$

可看出原先作為第二項的牌已經作為圈數列首項，故大部分圈數列僅有 1 項，若有第 2 項，則 $c_{i,2} = i+n$（如：$c_{4,2} = 9$，$c_{7,2} = 12$，$c_{10,2} = 15$）且 $c_{i,j+1} = \dfrac{c_{i,j}}{3}$，$j \ge 2$.（如：$c_{4,3} = \dfrac{c_{4,2}}{3} = 3$，$b_7 = \dfrac{c_{7,2}}{3} = 4$, $b_{10} = \dfrac{c_{10,2}}{3} = 5$）. 因此，將 $i+n$ 表示為 $p \times 3^{\gamma}$，則 $b_i = p$，$\alpha_i = 1+\gamma$。例如：上例 $\sigma_{6,5,5}$ 中，當 $i=4$，由 $i+n = 4+5 = 1\times 3^2$，故得 $b_4 = 1$ 且 $\alpha_4 = 2+1 = 3$。再如：當 $i=6$，由 $i+n = 6+5 = 11\times 3^0$，故得 $b_6 = 11$ 且 $\alpha_6 = 0+1 = 1$。

不難得到 h 堆的情形如下：

【定理 3】$\sigma_{\underbrace{n+1,\ n,\ n\ldots\ n}_{h\text{個}}}$

令 $i+n=b_i\times h^{\gamma_i}$ $(1\le i\le n(h-1))$，則：

1. $b\sigma_{\underbrace{n+1,\ n,\ n\ldots\ n}_{h\text{個}}}=\begin{pmatrix}1 & 1+h & \cdots & hn-h+1 & 2 & 2+h & \cdots & hn-h+2 & \cdots & hn-2h+2\\ b_1 & b_2 & \cdots & b_n & b_{n+2} & b_{n+2} & \cdots & b_{2n} & \cdots & b_{n(h-1)}\end{pmatrix}$

2. $\alpha_i=1+\gamma_i$.

例如：$ord(\sigma_{4,3,3,3,3})=?$

將 $n=3$、$h=5$ 代入 $i+n=b_i\times h^{r_i}$ 得 $i+3=b_i\times 5^{r_i}(1\le i\le 12)$，

$$\text{得}\begin{cases}(b_1,b_2,\ldots,b_{12})=(4,1,6,7,8,9,2,11,12,13,14,3)\\(\alpha_1,\alpha_2,\ldots,\alpha_{12})=(1,2,1,1,1,1,2,1,1,1,1,2)\end{cases}$$

$$b\sigma_{4,3,3,3,3}=\begin{pmatrix}1 & 6 & 11 & 2 & 7 & 12 & 3 & 8 & 13 & 4 & 9 & 14\\ 4 & 1 & 6 & 7 & 8 & 9 & 2 & 11 & 12 & 13 & 14 & 3\end{pmatrix}=(1,4,13,12,9,14,3,2,7,8,11,6)$$

故 $b\sigma_{4,3,3,3,3}$ 為一個圈，有 15 項，故得 $ord(\sigma_{4,3,3,3,3})=15$.

作者之一楊蕎卉編寫 VB 程式求出 $ord(\sigma_{n+1,n,n})$，$1\leqq n\leqq 1000$（流程圖及程式碼如附錄），其中發現一個有趣的結果：

當 $n=3^t$, $t=0,1,2,3,4,5,6$ 時，$ord(\sigma_{n+1,n,n})=3t+3$

進一步修改 VB 程式，增加堆數，設為 h 堆，檢驗 $n=h^t$, $t=1\sim13$ 的情形，得到以下結果：

當 $n=3^t$, $t=7\sim13$ 時，$ord(\sigma_{n+1,n,n})=3t+3$

當 $n=4^t$, $t=1\sim13$ 時，$ord(\sigma_{n+1,n,n,n})=4t+4$

當 $n=5^t$, $t=1\sim13$ 時，$ord(\sigma_{n+1,n,n,n,n})=5t+5$

根據上述程式所得的結果，初步觀察推測「$ord(\sigma_{\underbrace{h^t+1,h^t,h^t\ldots h^t}_{h\text{個}}})=h(t+1)$」，其證明及進一步的性質留待後續探討。

【推測】

$$ord(\sigma_{\underbrace{h^t+1,h^t,h^t\ldots h^t}_{h\text{個}}}) = h(t+1)\text{。}$$

參考資料：

李華介，**大學基礎代數**。http://math.ntnu.edu.tw/~li/algebra-html/algebra.pdf。

許志農等(2013)，**普通高級中學數學第二冊**。第 1 章數列與級數。臺北市：龍騰。

林建維、陳奕均、吳彥澄(2014)，**苗栗縣第 54 屆中小學科展國中組數學科作品**。「牌」山倒海－洗「排」遊戲。

林建維、陳奕均、吳彥澄、蘇柏奇、游淑媛(2014)，一個關於洗牌次數之基測題目的延伸探討。**科學教育月刊**。372，13-25。

黃莉芸、邱文均(2015)，**全國科展第 55 屆高中組數學科作品**。洗排遊戲的數學模式。

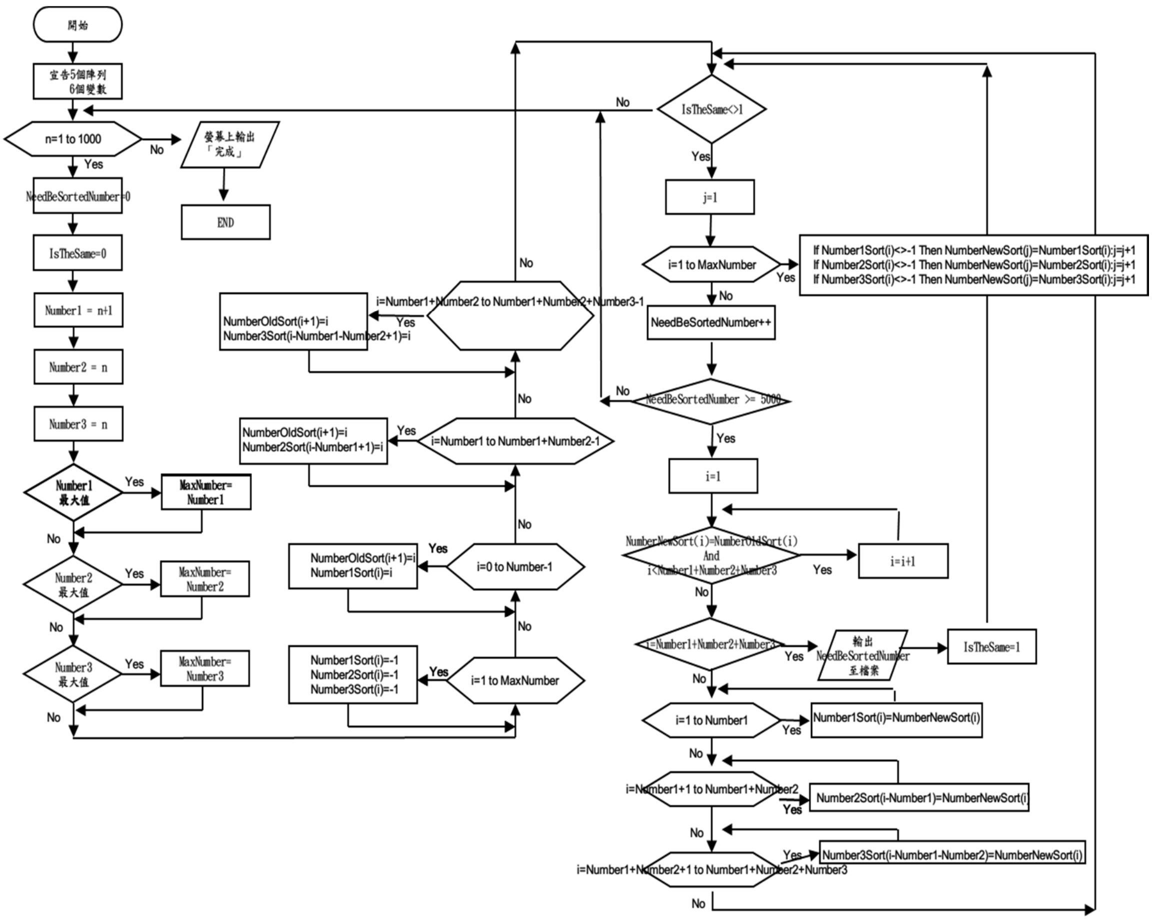
姜兆徽、吳承峰、林郁翔、廖奕瑋(2002)，**全國科展第 42 屆高中組數學科作品**。洗牌的數學。

P. Glaister (1994)，Card Shuffling for Beginners p5-7，*Mathematical -Spectrum* 27。

楊蕎卉、吳舜曄、曾巧閔(2017)，**全國科展第 57 屆高中組數學科作品**。多堆洗牌之最少次數探討。

102 年度基本學力測驗數學科試題。http://cap.nace.edu.tw/exam/102/102Math150dpi.pdf。

附錄：$ord(\sigma_{n+1,n,n})$, $1 \leqq n \leqq 1000$ 的程式流程圖與程式碼



Private Sub Command_Click()
Dim NumberOldSort(5000) As Long '一開始的陣列
Dim NumberNewSort(5000) As Long '重新排過的陣列
Dim Number1Sort(5000), Number2Sort(5000), Number3Sort(5000) As Long '三堆牌的陣列
Dim Number1, Number2, Number3, MaxNumber, IsTheSame, n As Long
Dim NeedBeSortedNumber As Long '排序的次數
Open "c:\temp\SFile001.txt" For Output As #1 '開啟文字檔,開始寫入 n 值與排序次數
For n = 1 To 1000
'設原本舊陣列與新陣列是不相同的，0 是不相同；1 是相同
IsTheSame = 0
NeedBeSortedNumber = 0
Number1 = n + 1
Number2 = n

```
Number3 = n
'找出三數之最大數，並將三個陣列的初起化皆設為-1
If Number1 >= Number2 And Number1 >= Number3 Then MaxNumber = Number1
If Number2 >= Number3 And Number2 >= Number1 Then MaxNumber = Number2
If Number3 >= Number1 And Number3 >= Number2 Then MaxNumber = Number3
For i = 1 To MaxNumber
    Number1Sort(i) = -1
    Number2Sort(i) = -1
    Number3Sort(i) = -1
Next
'從 0 循序填入陣列中，讓陣列內容呈現 0,1,2,3,4,.....
For i = 0 To Number1 - 1
NumberOldSort(i + 1) = i
Number1Sort(i + 1) = i
Next i
For i = Number1 To Number1 + Number2 - 1
NumberOldSort(i + 1) = i
Number2Sort(i - Number1 + 1) = i
Next i
For i = Number1 + Number2 To Number1 + Number2 + Number3 - 1
NumberOldSort(i + 1) = i
Number3Sort(i + 1 - Number1 - Number2) = i
Next i
'計算需要重新排序幾次，才能與原陣列一樣
While IsTheSame <> 1
    '開始從各陣列挑一個數，重新排列
    j = 1
    For i = 1 To MaxNumber
        If Number1Sort(i) <> -1 Then NumberNewSort(j) = Number1Sort(i): j = j + 1
        If Number2Sort(i) <> -1 Then NumberNewSort(j) = Number2Sort(i): j = j + 1
        If Number3Sort(i) <> -1 Then NumberNewSort(j) = Number3Sort(i): j = j + 1
    Next i
    NeedBeSortedNumber = NeedBeSortedNumber + 1
    If NeedBeSortedNumber >= 5000 Then GoTo Over
    '檢查重新排過的陣列與原陣列是否一樣
    i = 1
    While (NumberNewSort(i) = NumberOldSort(i)) And (i < Number1 + Number2 +
Number3)
       i = i + 1
    Wend
```

```
        If i = Number1 + Number2 + Number3 Then
            'MsgBox ("n=" & n & "共排序了" & NeedBeSortedNumber & "次")
            txt1 = "n=" & n & "共排序了" & NeedBeSortedNumber & "次"
            Print #1, txt1
            IsTheSame = 1
        Else
            For i = 1 To Number1
                Number1Sort(i) = NumberNewSort(i)
            Next i
            For i = Number1 + 1 To Number1 + Number2
                Number2Sort(i - Number1) = NumberNewSort(i)
            Next i
            For i = Number1 + Number2 + 1 To Number1 + Number2 + Number3
                Number3Sort(i - Number1 - Number2) = NumberNewSort(i)
            Next i
        End If
    Wend
    Next n
    Close #1
    MsgBox ("完成")
    End Sub
```