電腦專題

組合數學與電腦的關係

張鎮華

§1 前言

電腦是數學家爲了完成快速計算構思出來的產物，借著電機工程的協助，這個美夢得以實現，並且日新月異，一再改進。現在，電腦已經脫離它的母親——數學，自己茁壯成長，另成一天地；但是它和數學的關係，就像母子天倫，並不因其富貴而消失。

電腦可分兩大部份，一是硬體，這是電機知識的結晶；另一是軟體，其基本則在數學，簡言之，其中有算法（ algorithm ）及離散結構（ discrete structure ）等重要的數學概念。要學好電腦，離散數學是不可或缺的關鍵之一。近年來，國內外有知之士，已經在提倡，大學一年級除了應該學微積分以外，也要學習離散數學。中學數學教材裏適度的加入離散數學，也是遲早的事情。

從另一方面來看，電腦的興起，對於數學，特別是離散數學的研究也有很大的影響，最有名的例子是1978年證明的四色定理。四色問題源自十九世紀，有人問這樣的題目：將任意地圖上的國家塗色，使相鄰兩國的顏色不同，請問最少要用幾種顏色？很快的就有人「證明」四種顏色就夠了，但後來被發現證明有誤，四色問題遂搖身一變，成爲大家矚目的難題，許多數學家圍繞著它做了很多文章，一直到1978年才被人「用電腦證明」這定理正確，引起很大轟動。

電腦證明定理？誰相信！地圖有無窮多種，電腦如何能在有限的時間內驗證無窮多種地圖都可以用四色著色？可以的！簡單的說，方法是這樣的，先有一些概念和定理，利用這些幫助，四色定理的證明可分若干情況，每一情況對應一特殊地圖，證明某一情況等於驗證對應的地圖。在驗證某一情況時，有時就結束這情況，有時又細分成若干個情況，視對應的地圖而定。如此繼續下去，可能一直有細分的情況，則不能下任何結論；若是做到某時候，每個情況都停止，則表示四色定理成立。這些細分的過程既繁且雜，就借著電腦來做，而很幸運的，在花費很多計算時間以後，每個情況都停止了，所以四色定理就得證。

另一個重要的影響是，組合最優化因電腦而興起。早期的組合學著重在定性的定理以及計數理論，有了電腦之後，求某種最優解的組合問題就成了有興趣的問題。有關這方面的介紹，請參考數播第37期第53頁茁作「相異代

表系古今談」，並參考本期楊照崑及楊重駿的「未來數學家的挑戰——計算量問題」。

以上的說法，著重在研究上。在教學上，電腦的介入也越來越多，本文主要的內容就是介紹電腦在組合數學教學上的影響。

§2 組合學教學利用電腦的兩個好處

電腦既然與數學，特別是組合數學關係這麼密切，我們就來看看現行的敎育方法下，我們是否眞的將這個關係告訴了學生。令人驚訝的是，並沒有太多這樣的嘗試。縱然我們已經開始在電腦系有了「離散數學」的課程，但並不表示學生們學了這門課就眞正意識到，它對軟體程式的撰寫有何貢獻。基於這種考慮，我們提出兩個看法。

㈠由做組合數學的題目學習電腦程式的技巧，提高對電腦及數學的興趣。

㈡利用電腦當做敎組合數學的工具，特別要敎導學生用電腦當做實驗工具以猜測定理。

有關第一點，筆者於1985年春天在研究所開授「組合最優化II」時，做了一個嘗試，將所敎的組合問題，要求學生撰寫電腦程式，將所敎的算法眞正變爲有用的程式。敎學的內容包括圖的連通性、圖的連通分量、二分圖求對集、多弦圖的判別、多弦圖的最大穩定集、多弦圖的最小點團覆蓋……等問題。在課程中，從圖的表示法到Depth first search 以及 Breadth first search，學生們學到很多撰寫程式的技巧，同時在寫過程式以後，對原來的問題有更深入的瞭解。

有關第二點，筆者1985年秋天所開授的「組合學」中曾做了一些實驗。在這個課裏，我們的目標不是要學生學習高級的程式技巧，相反的，只要求他們懂得最基本寫程式的事情即可。目標是要利用電腦幫助觀察。

這個課程的開宗明義是介紹數學歸納法，這個從高中就爲學生所熟知的原理。它是一個基本工具，特別是在組合數學中，它以各種形式（如最小原理）出現。但是歸納法不只是在做形式的驗證而已，舉例來說，我們都有過證明

$$\begin{aligned}&1^2+2^2+\cdots\cdots+n^2\\&=n(n+1)(2n+1)/6\end{aligned}\qquad\text{(丙)}$$

的經驗，方法可能如下：

當 $n=1$ 時，$1^2=1(1+1)(2\times1+1)/6$ 顯然成立。

假設 $1^2+2^2+\cdots\cdots+n^2=n(n+1)(2n+1)/6$ 成立，則

$$\begin{aligned}&1^2+2^2+\cdots\cdots+n^2+(n+1)^2\\&=n(n+1)(2n+1)/6+(n+1)^2\\&=(n+1)[(n+1)+1][2(n+1)\\&\quad+1]/6\end{aligned}$$

第二行到第三行是經過一番化簡得到的。由數學歸納法得知㈣式對所有 n 均成立。證畢。

一個基本的問題是，如果我們要問

$$1^2+2^2+\cdots\cdots\cdots+n^2=?$$

那要從何著手？許多數學，特別是組合學上的問題都一樣，我們不會那麼幸運先知道答案，而必須去尋找答案，而數學歸納法的證明，常常只是我們找到正確答案以後的形式驗證罷了。

如何去尋找答案呢？最基本，看起來最笨，但也極有效的方法是做實驗，從特例觀察去發現通性。所以數學家也要做實驗，數學家的實驗器材是紙筆，是黑板粉筆，是木條沙堆，數學家的實驗室隨處均可。現在我們要告訴的是，電腦是一種很好的實驗工具，可以幫忙我們快速的做各種實驗，借以觀察通性，建立定理。

以下各節是我們在「組合學」課程中做過的一些電腦實驗，且讓我們逐一道來。

§3 n階乘有多大

最基本的排列組合和n階乘有不可分的關係，所以瞭解一點$n!$的性質是有興趣的事情。一般來說$n!$隨著n的增加而「快速」增加，要瞭解這一點有很多方法可行。用分析的方法，可以得到所謂的 Stirling公式

$$n! \approx \sqrt{2\pi n}\ \left(\frac{n}{e}\right)^n$$

這個公式顯示$n!$和n^n幾乎一樣，所以會快速增加。另外也可以拿一只計算器算看看$n!$，以我手邊的科學用計算器來說，$69! = 1.7112245 \times 10^{98}$，但是70! 就太大而容不下，所以出現0。

為了看出更大的$n!$，寫了一個FORTRAN程式，它可以告訴任何n值的$n!$。試過1000!，高達256位數；2000! 則達5735位。另外利用現成的套裝軟體MACSYMA也可以做同樣的事情，但只要打入1000或2000就可印出答案。

```
      INTEGER DIGIT(10000),LENGTH,N,A,Q,R
*
      PRINT *, 'This program calculates N!, please type a positive ',
     *         'integer as N.'
      ACCEPT *, N
*
      DIGIT(1)=1
      LENGEH=1
*
      NN=1
*
   10 Q=0
      J=1
   20 A=DIGIT(J)*NN+Q
      Q=A/10
      DIGIT(J)=A-Q*10
      J=J+1
      IF (J-LENGTH) 20,20,25
*
   25 IF (Q) 40,40,30
   30 LENGTH=LENGTH+1
      A=Q
      Q=A/10
      DIGIT(LENGTH)=A-Q*10
      GO TO 25
*
   40 NN=NN+1
      IF (NN-N) 10,10,90
*
   90 WRITE (6,95) N,(DIGIT(I),I=LENGTH,1,-1)
   95 FORMAT(1X,I5,'! = ',10(1X,5I1)/(10X,10(1X,5I1)))
*
      STOP
      END
```

§4 巴斯卡三角形的奇偶數分配

次一個基本的是$C(n,k)$，這表示由n不同物中取出k物的方法數。寫成公式是$C(n,k) = n!/k!(n-k)!$，或者用歸納法定義$C(0,0)=1$，$C(n,k)=C(n-1,k)+C(n-1,k-1)$。一個有趣的問題是，$C(n,k)$何時為奇數？何時為偶數？為了回答這個問題，我們開始做實驗，次頁是用BASIC 寫成的程式，印出巴斯卡三角形，但並非眞正算出$C(n,k)$的值，只算出它是奇

數還是偶數，圖中「 * 」表示奇數，「 · 」表示偶數。

從印出來的結果可以觀察到一些有趣的結果。例如，$n=2^m-1$（請看 $m=4$，$n=15$ 的例子）時，$C(n,k)$ 恒爲奇數；第 0 到第 2^m-1 列所構成的三角形，從三個方向看都是一樣的；而且這個三角形在接下來的 2^m 列中（第 2^m 到 $2^{m+1}-1$ 列）又出現相同的兩次，一在左，一在右，中間有一個全爲「 · 」的倒三角形。以上的規則事實上可以寫成下面的公式：若 $0\le k\le n=2^m+n'<2^{m+1}$，則

$$C(n,k)=\begin{cases} C(n',k) & (\bmod 2) \\ \quad 當\ 0\le k\le n' & (情況甲) \\ 0 & (\bmod 2) \\ \quad 當\ n'<k<2^m & (情況乙) \\ C(n',k-2^m) & (\bmod 2) \\ \quad 當\ 2^m\le k\le n & (情況丙) \end{cases}$$

```
100 DEFINT A-Z
110 DIM P(50)
120 P(0)=1: N=40
130 FOR I=0 TO N
140     LPRINT SPACE$(N-I);
150     FOR J=0 TO I: IF P(J)=1 THEN LPRINT " *"; ELSE LPRINT " .";
155     NEXT J
160     P(I+1)=1
170     FOR J=I TO 1 STEP -1: IF P(J)=P(J-1) THEN P(J)=0 ELSE P(J)=1
175     NEXT J
180     LPRINT I
190 NEXT I
200 END
```

```
                                         * 0
                                        * * 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
         * . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . * 32
        * * . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . * * 33
       * . * . . . . . . . . . . . . . . . . . . . . . . . . . . . . . * . * 34
      * * * * . . . . . . . . . . . . . . . . . . . . . . . . . . . . * * * * 35
     * . . . * . . . . . . . . . . . . . . . . . . . . . . . . . . . * . . . * 36
    * * . . * * . . . . . . . . . . . . . . . . . . . . . . . . . . * * . . * * 37
   * . * . * . * . . . . . . . . . . . . . . . . . . . . . . . . . * . * . * . * 38
  * * * * * * * * . . . . . . . . . . . . . . . . . . . . . . . . * * * * * * * * 39
 * . . . . . . . * . . . . . . . . . . . . . . . . . . . . . . . * . . . . . . . * 40
```

以上這個公式可以換一種形式，用數的二進位表示法來說，則更加簡明。將 n 及 k 分別用二進位表示法表示如下

$$n = a_m a_{m-1} a_{m-2} \cdots\cdots a_1 a_0 (B)$$

其中 $a_m = 1$

$$k = b_m b_{m-1} b_{m-2} \cdots\cdots b_1 b_0 (B)$$

則得

$$n' = a_{m-1} a_{m-2} \cdots\cdots a_1 a_0 (B)$$

並假設

$$k' = b_{m-1} b_{m-2} \cdots\cdots b_1 b_0 (B)$$

以上的表示法中，所有的 a_i 和 b_j 均是0或1的數。

情況甲表示 $b_m = 0$，也就是 $k = k'$，並且 $k' \leq n'$，此時公式表示我們可以忘掉 n 和 k 的最左一位，得到 n' 和 k'，求 $C(n, k)$ 的奇偶相當求 $C(n', k')$ 的奇偶。

情況乙表示 $b_m = 0$，也就是 $k = k'$，並且 $n' < k'$，此時恒有 $C(n, k)$ 爲偶數的結論。請注意，$n' < k'$ 表示存在某個 $b_i = 1 > 0 = a_i$。若我們定義 $n' < k'$ 時 $C(n', k')$ 是0，則也知道 $C(n, k) = C(n', k')$。

情況丙表示 $b_m = 1$，並且 $k' \leq n'$，此時和情況甲一樣可以得到 $C(n, k)$ 和 $C(n', k')$ 有同樣的奇偶性。

綜合以上三種情況，可見 $0 \leq k \leq n$ 時恒有 $C(n, k) = C(n', k') \pmod 2$。而 n' 和 k' 其實只是 n 和 k 二進位表法的最左一位去掉的結果。重複做上面的動作將可以得到 $C(n, k)$ 的奇偶性。結論是這樣的：

(1)若 $a_i \geqslant b_i$ 恒成立，則 $C(n, k)$ 爲奇數。

(2)若 $a_r < b_r$ 對某個 r 成立，則 $C(n, k)$ 爲偶數。

特殊的情況是 $n = 2^m - 1$ 時，所有 a_i 均等於1，所以所有 $C(n, k)$ 均是奇數。另一個有趣的結論是，當存在 s 個 a_i 等於1時，共有 2^s 個 k 使得 $C(n, k)$ 爲奇數，因爲 a_i 爲0時，b_i 必定爲0，a_i 爲1時，b_i 有兩種可能。

上面的結論是用觀察得到規律，再推出來的，一旦有了結論，再來的證明就是數學歸納法，並非難事。

定理：假設 n 和 k 是兩非負整數，其二進位表示法分別爲 $n = a_m a_{m-1} \cdots\cdots a_1 a_0 (B)$ 和 $k = b_m b_{m-1} \cdots\cdots b_1 b_0 (B)$，則 $C(n, k)$ 爲奇數的充份必要條件是 $b_i \leqslant a_i$，$i = 0, 1, \cdots\cdots, m$。

證明：$m = 1$ 時，本定理顯然成立。假設這定理對 $m-1$ 成立，考慮 m 的情況。

當 $a_0 = 0$（表示 n 爲偶數）且 $b_0 = 1$（表示 k 爲奇數）時，

$$\begin{aligned} C(n,k) &= \frac{n(n-1)(n-2)\cdots\cdots(n-k+1)}{k(k-1)(k-2)\cdots\cdots\cdots\cdots 1} \\ &= \frac{n}{k} C(n-1, k-1) \end{aligned}$$

因爲 n 是偶數，k 是奇數，$C(n-1, k-1)$ 是整數，所以 $C(n, k)$ 爲偶數，這時是因爲存在 $r = 0$ 使得 $b_r > a_r$ 的關係。

當 $a_0 = 0$（表示 n 爲偶數）且 $b_0 = 0$（表示 k 爲偶數）時，

$$\begin{aligned} C(n,k) &= \frac{n(n-2)\cdots\cdots(n-k+2)}{k(k-2)\cdots\cdots\quad 2} \cdot \frac{(n-1)(n-3)\cdots\cdots(n-k+1)}{(k-1)(k-3)\cdots\cdots\quad 1} \\ &= \frac{n'(n'-1)\cdots\cdots(n'-k'+1)}{k'(k'-1)\cdots\cdots\quad 1} \cdot \frac{\text{奇數}}{\text{奇數}} \end{aligned}$$

其中 $n' = n/2$ 且 $k' = k/2$，第二式是因爲第一式中右邊分子分母的偶數各除2的結果。因此 $C(n, k)$ 的奇偶和 $C(n', k')$ 的奇偶相同。但是 n' 和 k' 的二進位表示法分別是 n 和 k 的二進位表示法中的最右一位去掉的結果，由數學歸納法假設可知 $C(n', k')$ 爲奇數的充要條件是 $b_i \leqslant a_i$，$i = 1, 2, \cdots\cdots, m$，

但已知$b_0 = a_0$，所以也知道$C(n,k)$爲奇數的充要條件是$b_i \leqslant a_i$，$i = 0, 1, 2, \cdots\cdots, m$。

當$a_0 = 1$的情況，其討論類似，茲從略。由數學歸納法可見本定理成立。

§5　一個雙重數列

這個問題是由黃華民教授處聽來的。定義雙重數列$a(i,j)$，$i, j = 0, 1, 2, \cdots\cdots$如下：

$a(0,0) = 0$，

$a(i,j)$是不出現在$a(0,j)$，$a(1,j)$，$\cdots\cdots$，$a(i-1,j)$及$a(i,0)$，$a(i,1)$，$\cdots\cdots$，$a(i,j-1)$中的最小非負整數。

請問$a(i,j)$如何表示成i和j的式子？

我們也可以利用電腦實驗的方法得到如下的結論（有關詳細的過程，請參見數播第36期第53頁，9302最小非負整數的排列法，茲不重複）：

$$i = a_m a_{m-1} \cdots\cdots\cdots\cdots a_1 a_0 (B)$$
$$j = b_m b_{m-1} \cdots\cdots\cdots\cdots b_1 b_0 (B)$$
$$k = c_m c_{m-1} \cdots\cdots\cdots\cdots c_1 c_0 (B)$$

其中$c_i = a_i .\mathrm{XOR}. b_i$，或者$c_i = a_i + b_i \pmod 2$，則$a(i,j) = k$。

§6　魔方陣

魔方陣最早以驅鬼降魔的姿態出現在人類文明上，很多人都對它發生濃厚的興趣。一個n階魔方陣是將$1, 2, 3, \cdots\cdots, n^2$這$n^2$個數填入$n$乘$n$的方陣中，使其各行、各列及兩主對角線的和均爲定數。當然這個數是

$$(1 + 2 + \cdots\cdots + n^2) / n = n(n^2 + 1) / 2。$$

2階魔方陣顯然不存在，3階魔方陣只有一種（旋轉得到的不算新方陣），我國易經中的九宮圖其實是世界上第一個3階魔方陣。下面就是這個方陣。

8	1	6
3	5	7
4	9	2

以下我們介紹一七世紀 de la Loubere建造奇階魔方陣的方法（更多造魔方陣的方法參見數播各期）

㈠將1放在第一列正中央的方格上，接下來的數按順序放在前一數的右上方格，但有下面三點小修正。

㈡到達第一列時，下一數放在最底列，假想最底列是第一列的上面一列。

㈢到達最右列時，下一數放在最左列，假想最左列是最右列的右邊一列。

㈣當某一方格已經塡過數字，但又重新到達一次，則這數字放在前一數所在方格的下面一格。

以$n = 5$爲例可以得到下面的魔方陣。

17	24	1	8	15
24	5	7	14	16
4	6	13	20	22
10	12	19	21	3
11	18	25	2	9

爲了更進一步瞭解這種魔方陣，我們寫了一個簡單的BASIC程式，印出不同大小的奇魔方陣。

```
10 ' Magic square of odd order (de la Loubere's method).
20 ' See R.A.Brualdi: Introductory Combinatorics, page 6-7.
30 '----------------------------------------------------
40 DIM MAGIC(30,30)
50 INPUT "Print an odd number between 3 and 30 (print 0 for STOP)";N
51 IF N=0 THEN END
55 LPRINT: LPRINT: LPRINT "A magic square of order";N: LPRINT
57 FOR I = 1 TO N: FOR J = 1 TO N: MAGIC(I,J)=0: NEXT: NEXT
60 I=1:J=(N+1)/2
70 FOR K = 1 TO N*N
80      MAGIC(I,J)=K
90      IF I=1 THEN I1=N ELSE I1=I-1
100     IF J=N THEN J1=1 ELSE J1=J+1
110     IF MAGIC(I1,J1) <> 0 THEN I=I+1 ELSE I=I1:J=J1
120     NEXT K
130 FOR I = 1 TO N
140 FOR J = 1 TO N: LPRINT USING "####"; MAGIC(I,J);: NEXT J
150 LPRINT: NEXT I
160 GOTO 50
```

A magic square of order 5

17	24	1	8	15
23	5	7	14	16
4	6	13	20	22
10	12	19	21	3
11	18	25	2	9

A magic square of order 9

47	58	69	80	1	12	23	34	45
57	68	79	9	11	22	33	44	46
67	78	8	10	21	32	43	54	56
77	7	18	20	31	42	53	55	66
6	17	19	30	41	52	63	65	76
16	27	29	40	51	62	64	75	5
26	28	39	50	61	72	74	4	15
36	38	49	60	71	73	3	14	25
37	48	59	70	81	2	13	24	35

A magic square of order 15

122	139	156	173	190	207	224	1	18	35	52	69	86	103	120
138	155	172	189	206	223	15	17	34	51	68	85	102	119	121
154	171	188	205	222	14	16	33	50	67	84	101	118	135	137
170	187	204	221	13	30	32	49	66	83	100	117	134	136	153
186	203	220	12	29	31	48	65	82	99	116	133	150	152	169
202	219	11	28	45	47	64	81	98	115	132	149	151	168	185
218	10	27	44	46	63	80	97	114	131	148	165	167	184	201
9	26	43	60	62	79	96	113	130	147	164	166	183	200	217
25	42	59	61	78	95	112	129	146	163	180	182	199	216	8
41	58	75	77	94	111	128	145	162	179	181	198	215	7	24
57	74	76	93	110	127	144	161	178	195	197	214	6	23	40
73	90	92	109	126	143	160	177	194	196	213	5	22	39	56
89	91	108	125	142	159	176	193	210	212	4	21	38	55	72
105	107	124	141	158	175	192	209	211	3	20	37	54	71	88
106	123	140	157	174	191	208	225	2	19	36	53	70	87	104

經過觀察的結果，發現這樣的規律。將方陣分成 n 個右對角線，第 i 條右對角線包括 $(1, i), (2, i-1), \cdots\cdots, (i, 1), (i+1, n), (i+2, i-1), \cdots\cdots, (n, i+1)$ 這 n 個格子，或者是所有 (r, s) 使得 $r+s=i+1 \pmod n$ 的格子。觀察的結果是：

$1, 2, \cdots\cdots, n$ 出現在第 $(n+1)/2$ 右對角線，

$n+1, n+2, \cdots\cdots, 2n$ 出現在第 $(n+3)/2$ 右對角線，

⋮ ⋮

$in+1, in+2, \cdots\cdots, in+n$ 出現在第 $(n+1)/2+i \pmod n$ 右對角線。

如果將 $in+j$ 表示成一個 n 進位數，個位是 j，十位是 i，爲了使 j 小於 n，我們不妨將魔方陣的每一數減 1（這樣仍然保持每列、每行及兩主對角線和均爲定數的特性），則每一數的 n 進位表示法中，十位數對應它應該在那一條右對角線，個位數則和它在該右對角線的位置有關。再寫一程式做這件事情，以 $n=9$ 爲例，印出下面的結果。

5	6	7	8	0	1	2	3	4
6	7	8	0	1	2	3	4	5
7	8	0	1	2	3	4	5	6
8	0	1	2	3	4	5	6	7
0	1	2	3	4	5	6	7	8
1	2	3	4	5	6	7	8	0
2	3	4	5	6	7	8	0	1
3	4	5	6	7	8	0	1	2
4	5	6	7	8	0	1	2	3

1	3	5	7	0	2	4	6	8
2	4	6	8	1	3	5	7	0
3	5	7	0	2	4	6	8	1
4	6	8	1	3	5	7	0	2
5	7	0	2	4	6	8	1	3
6	8	1	3	5	7	0	2	4
7	0	2	4	6	8	1	3	5
8	1	3	5	7	0	2	4	6
0	2	4	6	8	1	3	5	7

以上這兩個方陣都是拉丁方陣，也就是用 $0, 1, \cdots\cdots, n-1$ 這 n 個數字填成，使每一行（列）不出現兩個相同數字。事實上第一個拉丁方陣 $A=(a_{ij})$ 定義如下：

$$a_{ij} = i + j + (n+1)/2 \pmod n, \quad i, j = 0, 1, \cdots\cdots, n-1。$$

第二個拉丁方陣 $B=(b_{ij})$ 定義如下：

$$b_{ij} = i + 2j + 1 \pmod n, \quad i, j = 0, 1, \cdots\cdots, n-1。$$

這兩個拉丁方陣和原來 de la Loubere 造出來的魔方陣 $M=(m_{ij})$ 的關係是

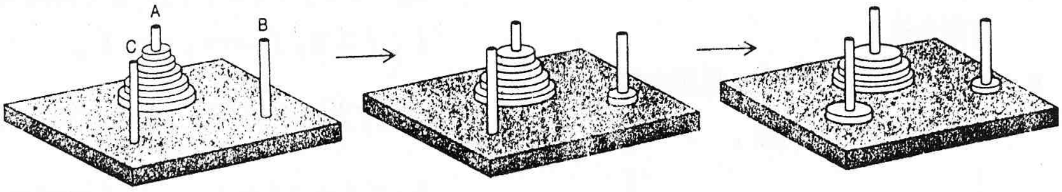
$$m_{ij} = n a_{ij} + b_{ij} + 1, \quad i, j = 0, 1, \cdots\cdots, n-1。$$

M 所以是一個魔方陣，其理由可以從 A 和 B 的性質看出來。首先，A 和 B 各行、各列及兩主對角線和都是常數，也就是 $0+1+\cdots\cdots+(n-1)=n(n-1)/2$，所以 M 也有這樣的性質。請注意，一般的拉丁方陣並不要求主對角線和等於 $n(n-1)/2$。其次，A 和 B 是所謂的正交拉丁方陣，也就是 (a_{ij}, b_{ij}) 將所有 (r, s)，$0 \leq r, s \leq n-1$，各出現一次，所以 m_{ij} 各不相同，也就是由 1 到 n^2 的所有數都出現在 M 中。

以上表示尋求正交拉丁方陣是造魔方陣的方法之一，請注意要處理對角線和爲定數這件事情。

§ 7 河內之塔

傳說河內在古時候有座塔，上有三根柱子，其中之一放了 64 個大小不相同的盤子，小的放在較大的上面。廟中的僧侶奉神之命，將盤子由一根柱子移到另一根，但是移動的過程中，小盤子不能放在大盤子之上。據說，將 64 個盤子由原來的柱子全部移到另一根柱子後，世界末日到了。下圖是只有 5 個盤子的前兩次移動情況。



馬上就會問，如何移動才會使移動的次數最少？這個數目是多少？若含有 n 個盤子，最少要做 $f(n)$ 次移動，才可以將所有盤子由原來的柱子移到另一根柱子，下面的遞迴方法可以解決這個問題。

假設 n 個盤子原來放在第 1 根柱子，我們可以用 $f(n-1)$ 次將上面的 $n-1$ 個盤子移到第 2 根柱子，然後再將最大的盤子移到第 3 根柱子，最後將第 2 根柱子的 $n-1$ 個盤子用 $f(n-1)$ 次移到第 3 根柱子。這樣的做法事實上是最好的方法，因爲要將最大的盤子由第 1 根柱子移到第 3 根柱子，一定得先把上面的 $n-1$ 個盤子移走，爲了要使最大的盤子能在不違反「小盤在大盤之上」的原則，這 $n-1$ 個盤子必須先移到第 2 根柱子。所以

$$\begin{aligned} f(n) &= f(n-1) + 1 + f(n-1) \\ &= 1 + 2f(n-1) \end{aligned}$$

再由 $f(1) = 1$ 可以得知

$$\begin{aligned} f(n) &= 1 + 2f(n-1) \\ &= 1 + 2 + 2^2 f(n-2) \\ &= 1 + 2 + 2^2 + 2^3 f(n-3) \\ &\ \ \vdots \qquad\qquad\qquad \vdots \\ &= 1 + 2 + 2^2 + \cdots\cdots + 2^{n-1} \\ &= 2^n - 1 \end{aligned}$$

$f(64) = 2^{64} - 1 \approx 1.8 \times 10^{19}$ 是個天文數字，所以河內的僧侶要將盤子全部移好，實非易事。另一件值得深思的事情是，這些僧侶到底要怎樣去移動盤子？如果用一個遞迴的程式可能是這樣的。

TOWER (n , source → target) :

  TOWER ($n-1$, source → spare)

  MOVE (n , source → target)

  TOWER ($n-1$, spare → target)

這個程式簡單明白，完全將前述 $f(n) = f(n-1) + 1 + f(n-1)$ 的本質表現無遺。這樣的程式在ALGOL型的程式語言都可以做。但是除非眞正有一部電腦，這樣的程式對那些僧侶們實無好處。遞迴程式（或公式）原是一種簡潔而抽象的做法，並不利於人們瞭解。例如，定義 n 的階乘 $n!$，可以是

$0! = 1$;

$n! = (n-1)!n$ ，當 $n \geqslant 1$ 時。

但不如寫成

$$n! = 1 \cdot 2 \cdot 3 \cdot \cdots\cdots \cdot n$$

更容易瞭解。這中間的差別是，前者簡潔而且合於本世紀數學家所謂「嚴密」的風格；後者是將前者反覆「代入」，做實驗而寫成，一眼明顯 $n!$ 的本質，但不是「嚴密」的定義方法。

爲了那些沒有電腦的僧侶們，我們想把河內之塔的遞迴程式看出它非遞迴的本質是什麼。寫一個 PASCAL 的程式，印出下面的實驗結果。

```
move disk 1 from stick 1 to stick 2
move disk 2 from stick 1 to stick 3
move disk 1 from stick 2 to stick 3
move disk 3 from stick 1 to stick 2
move disk 1 from stick 3 to stick 1
move disk 2 from stick 3 to stick 2
move disk 1 from stick 1 to stick 2
move disk 4 from stick 1 to stick 3
move disk 1 from stick 2 to stick 3
move disk 2 from stick 2 to stick 1
move disk 1 from stick 3 to stick 1
move disk 3 from stick 2 to stick 3
```

```
move disk 1 from stick 1 to stick 2
move disk 2 from stick 1 to stick 3
move disk 1 from stick 2 to stick 3
move disk 5 from stick 1 to stick 2
move disk 1 from stick 3 to stick 1
move disk 2 from stick 3 to stick 2
move disk 1 from stick 1 to stick 2
move disk 3 from stick 3 to stick 1
move disk 1 from stick 2 to stick 3
move disk 2 from stick 2 to stick 1
move disk 1 from stick 3 to stick 1
move disk 4 from stick 3 to stick 2
move disk 1 from stick 1 to stick 2
move disk 2 from stick 1 to stick 3
move disk 1 from stick 2 to stick 3
move disk 3 from stick 1 to stick 2
move disk 1 from stick 3 to stick 1
move disk 2 from stick 3 to stick 2
move disk 1 from stick 1 to stick 2
```

仔細觀察，發現移動的盤子竟然是按照這樣的規律在移動：交替做㈎和㈄直到不能做為止。

㈎將 1 號盤子從第 i 根柱子移到第 $i+1 \pmod 3$ 根柱子。

㈄移動一個異於 1 號的盤子。

請注意，在㈄的情況，恰有一個盤子可以移動，因為不含 1 號盤的兩根柱子中恰有一個盤子（較小的）可以移動。這樣的規律簡明可愛，容易記住，容易執行，遞迴程式就不再需要了。

更進一步的觀察上面印出來的結果，發現 1 號盤每 2 次出現一次，2 號盤每 4 次出現一次，3 號盤每 8 次出現一次，……。這樣的規律用數的二進位表方來說是這樣的。

```
0 0 0 0 0
0 0 0 0 1   (1)
0 0 0 1 0   (2)
0 0 0 1 1   (1)
0 0 1 0 0   (3)
0 0 1 0 1   (1)
0 0 1 1 0   (2)
0 0 1 1 1   (1)
0 1 0 0 0   (4)
```

將 0 到 2^n-1 用長度為 n 的二進位依次寫下來（上圖是 $n=5$ 的前 9 個數），對任一數 $s=a_n a_{n-1} \cdots\cdots a_n$ 和它前面一個數 $s-1=b_n b_{n-1} \cdots\cdots b_1$，恰有一個 r 使得 $a_r=1$ 但 $b_r=0$，上圖的 r 就是寫在 s 的二進位表法右邊的括號內。在第 s 次，總是移動第 r 號盤子。

§8 後 記

在這個課程中，我們還有其他一些體材，其精神和上面的做法都一樣，用電腦做實驗來幫助觀察規律，創造定理，再來證明它。限於篇幅我們無法一一列舉。

最後我要特別感謝王九逵老師，以上利用電腦做實驗的想法實是得自於他。另外王老師在其他許多事務上的做法，也處處為我的模範，終生受用不盡。

參考資料

1. R. A. Brualdi, *Introductory Combinatorics*, 1977.
2. C. L. Liu, *Introduction to Combinatorial Mathematics*, 1960.
3. W. W. Rouseball and H. S. M. Coxeter, *Mathematical Recreatons and Essays*, 12nd edition, 1974.

～本文作者現任教於國立中央大學數學系～