1110100 的出現及最早的改錯碼之旅

林福林

南臺科技大學

故事開始於游森棚教授在數學傳播上的一篇文章，“從數學競賽到數學研究 - 從 2002-C6 說起”（游森棚，2018），文章中提到：

…這個背景稱為德布魯因(de Bruijn)序列問題，我們先看看這個益智遊戲：能否將八個 0 或 1 排成一圈，使得讀三個相鄰位元（當作二進位來讀）剛好出現 000, 001, ... , 111 各一次？答案是可以的，比如 00011101 就是一個解…。習題 18：將一共 16 個 0 或 1 排成一圈，使得讀四個相鄰位元（當作二進位來讀）剛好出現 0000, 0001,..., 1111 各一次。

筆者雖然不是數學系出身，但是看到這段文字卻有相當親切的感覺，8 個(或 16 個) 0 或 1 排成一圈，正是把通訊理論中的最大長度序列(maximal-length sequence)補一個 0 再圍成一圈就完成了。最大長度序列可以運用在改錯碼，例如漢明碼(Hamming code)；可以拿來做精準的距離測量，例如全球定位系統(GPS)；可以做無線遙控的辨識系統，例如遙控鐵捲門；最重要的是可以拿來做展頻(spread spectrum)通訊所需要的展頻碼。作者從事展頻通訊系統製造及基地台與行動台的布建時間超過 6 年，由於最大長度序列的數學一般高中生有機會聽得懂，因此本文將擷取通訊教科書的相關內容，先介紹最大長度序列是如何產生，並且舉最簡單的(7,4)漢明碼為例，說明改錯碼的基本概念及其與最大長度序列的關聯性，並同時回答了游森棚教授的習題 18。

電腦與通訊的計數基礎稱為位元(bit)，通常用0或1來表示。我們先定義它的二進制加法及二進制乘法如表1。先熱身一下，做個二位元係數多項式的乘法練習題，

$(x+1)\times(x+1)=x^2+(1+1)x+1=x^2+1$ 。

表1 ：二進制加法及二進制乘法的運算規則

二進制加法	二進制乘法
$0+0=0$	$0\times0=0$
$0+1=1$	$0\times1=0$
$1+0=1$	$1\times0=0$
$1+1=0$	$1\times1=1$

接著我們探討一種序列的產生方法(Holmes, 1982)，考慮一個簡單的移位暫存器產生器(simple shift register generator, SSRG)架構，如圖1(a)，它是由3個移位暫存器和邏輯電路所組成並連接成一個多迴路的回授電路。在SSRG的架構中，所有的回授信號經運算成為單一回授輸入信號。移位暫存器序列被定義為最後暫存器的輸出序列。

假設移位暫存器的初始狀態為100 (S1 S2 S3)。經過一個時脈(clock)觸發，輸出狀態為110 (S1 S2 S3)，當時脈連續觸發，輸出狀態(S1 S2 S3)則為100, 110, 111, 011, 101, 010, 001, 100, ...，輸出序列即是最後一個移位暫存器的輸出位元，也是輸出狀態(S1 S2 S3)的最後一個位元，紀錄為

0 0 1 1 1 0 1 0 0 1 1 1 0 1 0 0....　　(1)

在這個輸出序列中有7個單元(0或1)，且這7個單元會依序重覆出現。圖1(b)顯示圖1(a)的電路所產生的狀態圖，它分成兩群，如果初始狀態為000，輸出序列永遠是0，如果初始狀態不是000，輸出序列永遠是(1)式，(1)式的序列也稱為 $m = 3$ 的最大長度序列， $m = 3$ 表示移位暫存器有3個，它所能產生的序列週期如果是 $n = 2^m - 1$，則稱為最大長度序列。讀者應該已經注意到1 1 1 0 1 0 0的出現，只要尾巴再補上一個0，就可以將8個 0 或 1 排成一圈, 使得讀三個相鄰位元剛好出現 000, 001, ..., 111 各一次。

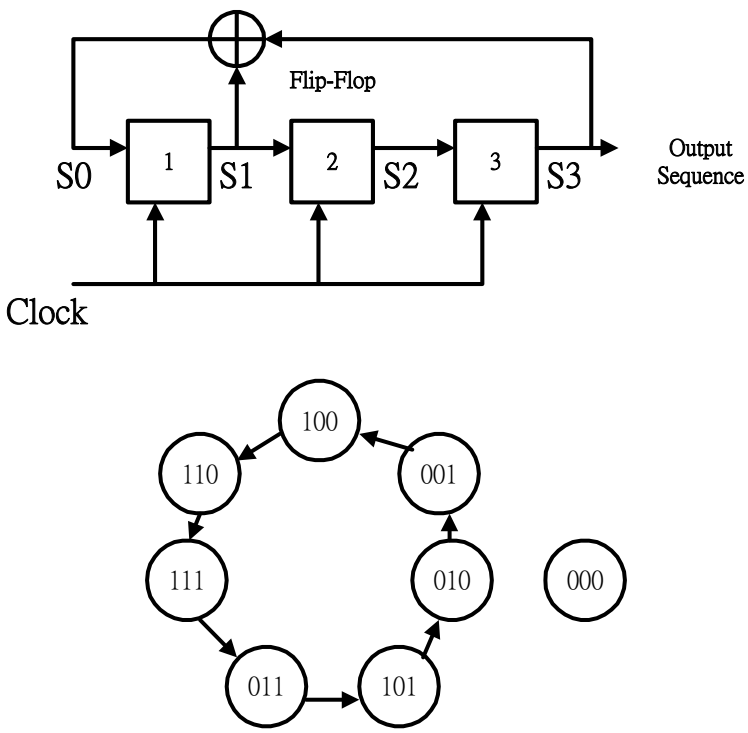

圖 1 ：一個 SSRG 的例子：(a) 簡單的移位暫存器電路，(b) 產生的狀態圖

參考圖2，假設我們將每一個移位暫存器的輸出設為 x_i，其中i為暫存器的編號，如果回授函數 $f(x_1, x_2, ..., x_m)$ 可以被表示為二進制的加法，則我們說SSRG是線性的(linear)。線性回授移位暫存器(linear feedback shift register, LFSR)通常由互斥邏輯閘(XOR)和一組移位暫存器所組成的電路來實現，架構中包含一串1位元的儲存裝置，每一個裝置有一條輸出線和輸入線，輸出線表示目前儲存的數值。在稱為時序(clock times)的離散時間瞬間，輸入線的數值取代儲存裝置內的數值。整個LFSR由時序控制同步動作，使得全部暫存器移1個位元。圖1(a)就是一個線性回授移位暫存器的例子，其中 $S0 = f(x_1, x_2, x_3) = x_1 \oplus x_3$。

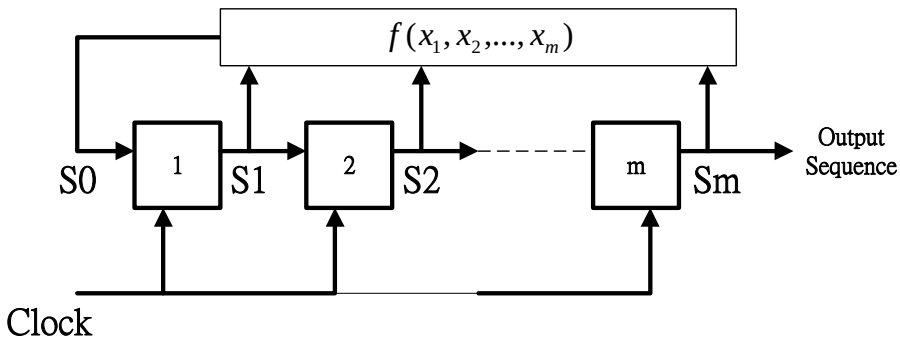

圖 2：二位元回授移位暫存器之序列產生器

圖 3 為線性 SSRG 的基本模型，其中的 c_i 可以是 0 或 1，稱為回授加權(feedback weight)。回授序列元(feedback sequence element) $a_k (= a(k))$可以表示成如下的 m 階線性遞迴關係(linear recursion relation)：

$$a(k) = c_1 a(k-1) + c_2 a(k-2) + ... + c_n a(k-m) = \sum_{i=1}^{m} c_i a(k-i) \quad k \geq m \tag{2}$$

其中 $a(0), a(-1), ..., a(-m+1)$ 稱為初始條件，$a(k)$ 中的 k 對應到第 k 個時脈(clock)。電路實現的簡單例子如圖 1(a)所示，它是下面式子的一個 3 位元 LFSR。

$$a(k) = a(k-1) + a(k-3) \tag{3}$$

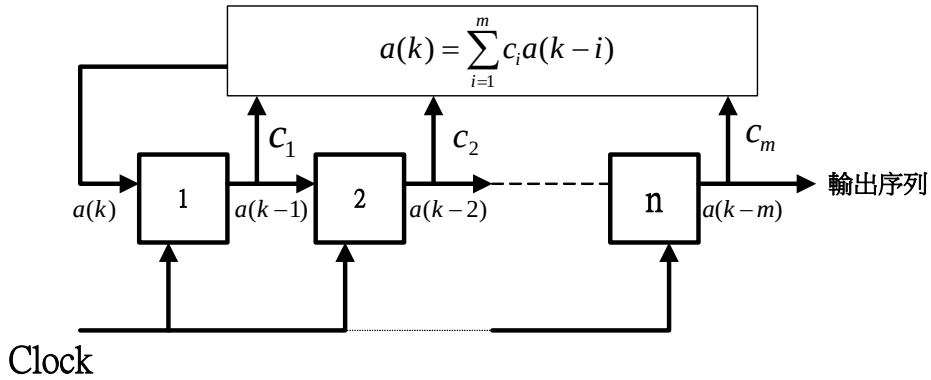

圖 3：二位元線性回授移位暫存器之序列產生器

為了解釋圖 2 的回授函數 $f(x_1, x_2, ..., x_m)$，我們先考慮一個係數由 0 和 1 組成的多項式，例如 $1+x^7$，把它作因式分解的結果如下：

$$1+x^7=(1+x)(1+x^2+x^3)(1+x+x^3) \tag{4}$$

一個具有 0 或 1 的係數之 m 次多項式，如果不能被分解，或者不能被另一個次數比 m 低的多項式整除，我們稱為不可分解的(irreducible)，例如：

$$1+x+x^5=(1+x^2+x^3)(1+x+x^2) \tag{5}$$

即是可被分解的。然而像(4)式右邊的多項式 $1+x+x^3$ 則不能被分解。

介紹一個名詞稱為特徵多項式(characteristic polynomial)，它跟線性回授移位暫存器有關。考慮圖 1 的例子，連續輸出狀態為 100, 110, 111, …，它對應到轉移矩陣(特徵矩陣)如下：

$$A=\begin{bmatrix} c_1 & c_2 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}=\begin{bmatrix} 1 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \tag{6}$$

轉移矩陣的作用是把前一個狀態轉移到下一個狀態。例如如果知道前一個狀態是 100，則經過一個時脈得到下一個狀態是 110，矩陣運算如下：

$$\begin{bmatrix}1 & 0 & 1\\ 1 & 0 & 0\\ 0 & 1 & 0\end{bmatrix}\begin{bmatrix}1\\ 0\\ 0\end{bmatrix}=\begin{bmatrix}1\\ 1\\ 0\end{bmatrix} \tag{7}$$

同理，如果知道前一個狀態是 110，則下一個狀態是

$$\begin{bmatrix}1 & 0 & 1\\ 1 & 0 & 0\\ 0 & 1 & 0\end{bmatrix}\begin{bmatrix}1\\ 1\\ 0\end{bmatrix}=\begin{bmatrix}1\\ 1\\ 1\end{bmatrix} \tag{8}$$

設定 $|A-\lambda I|=0$ 可以得到特徵方程式，方程式中等式的右邊是 0，等式的左邊即是特徵多項式。特徵多項式的寫法如下：

$$f(x)=\sum_{k=0}^{m}c_k x^k,\quad \text{其中} c_0=1 \text{且} c_m=1 \tag{9}$$

一個特徵多項式會決定圖 3 的箭頭是否要斷開，例如若 $c_2=0$，則圖 3 中對應到 c_2 的箭頭可以取消，一個特徵多項式會對應到一串輸出序列 $\{a_n\}$，例如 $1+x+x^3$ 的多項式(參考圖 1(a))對應到如下的輸出序列：

$$\begin{array}{cccccccccc} a_0, & a_1, & a_2, & a_3, & a_4, & a_5, & a_6, & a_7, & a_8, & \cdots\\ 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & \cdots \end{array} \tag{10}$$

底下有幾個定理皆來自教科書(Holmes, 1982)，為了避免敘述過於繁複，證明都一律省略，有興趣的讀者可翻閱參考文獻(Holmes, 1982)。

定理 1：如果序列 $\{a_n\}$ 的週期為 p，則 $f(x)$ 可以整除 $1+x^p$，亦可表示成 $f(x)|1+x^p$。

例如(10)式是週期序列，週期為 $p=7$，所以 $1+x+x^3$ 可以整除 $1+x^7$。

定理 2：如果 $f(x)|1+x^p$ 且 p 是使 $f(x)|1+x^p$ 成立的最小正整數，則序列 $\{a_n\}$ 在初始條件 $0,0,0,\cdots,0,1$ 的情況下有週期 p。

定理 3：如果 SSRG 序列$\{a_n\}$有最大長度，則它的特徵多項式 $f(x)$ 是不可分解的。(對最大長度序列是必要但不是充分的條件)

特別強調，如果一特徵多項式是不可分解的，則它的序列不一定是最大長度。例如，考慮下面這個不可分解的多項式

$$f(x)=1+x^3+x^6 \tag{11}$$

由於 $f(x)|1+x^9$，我們從定理 2 中看到存在一個週期為 9 的序列，但它不是 $2^6-1=63$，所以無法產生最大長度序列。

目前已被證明每一個次數 $m>1$ 的不可分解的多項式都可以被多項式 $1+x^{2^m-1}$ 整除，因此我們可以寫出下列的定理：

定理 4：已知特徵多項式是不可分解的且次數為 m 階，則 SSRG 序列的週期會是 2^m-1 的因數。

以(11)式為例，9 是 63 的一個因數。對這結果作推論，我們得到：若 2^m-1 是質數(prime)，則每一個 m 次不可分解的特徵多項式會對應到一組最大長度序列。再舉個例子，假設 $m=5$，因為 $2^5-1=31$ 是質數，所以 5 次不可分解的特徵多項式像是 $1+x^2+x^5$ 就會對應到一組最大長度序列。

反過來說，對每一個 m 而言，如果我們需要一個最大長度序列，我們必須限制我們的特徵多項式是原生的 (primitive)。一個 m 次不可分解的多項式稱為原生的若且為若對 $m<n<2^m-1$，此多項式都無法整除 $1+x^n$。例如 $m=4$ 的不可分解多項式 $1+x+x^4$ 都無法整除 $1+x^5$，$1+x^6\cdots$到 $1+x^{14}$，此時我們說 $1+x+x^4$ 是原生的多項式。

由上面說明知如果特徵多項式是原生的，則產生的序列為最大長度序列，也稱為 m-序列，表 2 列出其中的一些例子。以 GPS 採用的 C/A 碼為例，他結合兩個特別挑出來的 $m=10$ 的最大長度序列，稱為優先選擇對(preferred pair)，由於是 Gold 最早提出來的概念，因此也稱作金氏碼(Gold code)， C/A 碼的長度(序列週期)為 $2^{10}-1=1023$ 位元。當讀者採用手機做定位時，你已經接受了金氏碼的服務。前面提到作者從事展頻通訊時間超過 6 年，展頻過程也是使用金氏碼，其優先選擇對是兩個 $m=11$ 的最大長度序列，特徵多項式分別是 $f_1(x)=1+x^2+x^{11}$ 及 $f_2(x)=1+x^2+x^5+x^8+x^{11}$。

表 2：最大長度序列的例子

暫存器數目	序列長度	原生的多項式例子
3	7	$1+x+x^3$
4	15	$1+x+x^4$
5	31	$1+x^2+x^5$
6	63	$1+x+x^6$
7	127	$1+x+x^7$
8	255	$1+x+x^5+x^6+x^8$
9	511	$1+x^4+x^9$
10	1023	$1+x^3+x^{10}$
11	2047	$1+x^2+x^{11}$

m-序列有幾種良好的特性對於展頻通訊及各種運用深具吸引力。假設移位暫存器的個數為 m，則其重複週期為 $N=2^m-1$ 個位元，如同上文的討論，我們整理並說明如下 (Holmes, 1982)：

特性 1. 平衡特性(balance property)：在每一個序列週期內，0 與 1 的個數相差一個，即一個 m-序列有 2^{m-1}個 1 和 $2^{m-1}-1$個 0。

以 $f(x)=1+x+x^4$ (m=4) 為例，輸出序列{0 0 0 1 0 0 1 1 0 1 0 1 1 1 1}中有 $2^{4-1}=8$ 個 1 及 $2^{4-1}-1=7$ 個 0，即 1 的個數比 0 的個數多一個。

特性 2. 沿著輸出序列平移長度 m 的視窗 N 次($N=2^m-1$)，除了全部是 0 的序列以外，視窗所見 m 位元的所有組合各只出現一次。以 $f(x)=1+x+x^4$ (m=4) 為例，可以看到 {0001}，{0010}，{0100}，{1001}…，{1111}，{1110}，{1100}，{1000}，總共 15 個 4 位元狀態各只出現一次。這也回答了游森棚教授在文章中提到的習題 18 的解，只要再補上{0000}即可成為習題 18 的 1 個解。

特性 3. 串的特性(run property)：所謂的串是序列中相同位元凝結在一起稱為一串，例如 00 或 111。在每一個序列週期內，位元串長度 (run length) 為 1 的出現機率是 1/2，位元串長度為 2 的出現機率是 1/4，依此類推。反過來說，在每一個序列週期內共有 2^{m-1} 個位元串，其中有 1 個長度 m 的全 1 序列(run of ones)；有一個長度(m－1)的全 0 序列(run of zeros)；長度為 $m-2$ 的全 1 序列和全 0 序列各有 1 個；長度為 $m-3$ 的全 1 序列和全 0 序列各有 2 個，依此類推，通常有長度為 1 的全 1 序列和全 0 序列各 2^{m-3} 個。

以 $f(x)=1+x+x^4$ 為例，輸出序列{000100110101111}中共有 $2^{m-1}=2^3=8$ 個位元串，有一個長度 4 的全 1 序列{1111}；有一個長度 3 的全 0 序列{000}；長度為 2 的全 1 序列和全 0 序列各有 1 個；長度為 1 的全 1 序列和全 0 序列各有 2 個；位元串長度為 1 的出現機率是 4/8=1/2，位元串長度為 2 的出現機率是 2/8=1/4，依此類推。

還有其他特性像是自相關(auto-correlation)特性及擬隨機(pseudo-random)特性等等，我們在此先告一個段落。

討論至此我們來關心一下最大長度特徵多項式的數量(Holmes, 1982)，根據因數唯一分解定理(unique factorization theorem of arithmetic)，每一個大於 1 的正整數 n 可以表示成相異質數的次方(power)的乘積，如下式：

$$n=\prod_{i=1}^{k} p_i^{\alpha_i} \tag{12}$$

其中 $p_1, p_2, \cdots p_k$ 是相異的質數，而 $\alpha_1, \alpha_2, \cdots \alpha_k$ 都是正整數，且 $k \geq 1$。

例如，我們可以把 63 表示成

$$63=3^2 \times 7 \tag{13}$$

對應到式(12)，可以得到 $p_1=3$，　$p_2=7$，$\alpha_1=2$ 及 $\alpha_2=1$ 。

尤拉(Euler)函數 $\phi(n)$ 被定義為

$$\phi(n)=\begin{cases} 1 & n=1 \\ \prod_{i=1}^{k} p_i^{\alpha_i-1}(p_i-1) & n>1 \\ p-1 & \text{如果 } n=p \text{ 是質數} \end{cases} \tag{14}$$

Holms 在書中(Holmes, 1982, p339)提到 Zieler 已經證明最大長度序列的個數可以(15)式表示：

$$N_M(m) = \frac{\phi(2^m - 1)}{m} \tag{15}$$

例如，考慮 m=6，我們得到

$$N_M(n = 6) = \frac{\phi(2^6 - 1)}{6} = \frac{\phi(3^2 \times 7)}{6} = \frac{3 \times 2 \times 1 \times 6}{6} = 6 \tag{16}$$

因此針對 m=6 這個例子，我們知道有 6 個最大長度序列。表 3 列出一部份的資料，包括碼長度($N = 2^m - 1$)，不可分解多項式的個數 N_I，和最大冪數(exponent)多項式的個數 N_M。

特別注意到表 3 中當 N 是質數時，$N_M = N_I$，因為此時所有的不可分解多項式都是原生的多項式，包括 $m = 3, 5, 7$，反之，像 $N = 2^{10} - 1 = 1023 = 3 \times 11 \times 31$ 或 $N = 2^{11} - 1 = 2047 = 23 \times 89$ 都不是質數，因此有一部分的不可分解多項式不是原生的，因而產生 $N_M < N_I$ 的結果。

表 3：最大長度(原生的)序列個數和不可分解多項式的個數

m	$N = 2^m - 1$	N_M	N_I
3	7	2	2
4	15	2	3
5	31	6	6
6	63	6	9
7	127	18	18
8	255	16	30
9	511	48	56
10	1023	60	99
11	2047	176	186

介紹完最大長度序列之後，我們舉一個簡單的應用。底下是節錄並濃縮自書本(翁萬德等，2015）的一段文章:

…漢明(Hamming)回憶起使用機械式繼電器電腦的懊惱，他僅能在週末使用這部電腦，常常隔週進來發現東西都被停擺(dumped)，於是他說：「該死，要是機器可以偵測到一個錯誤就不會停擺了，為何機器不能確定出錯誤的位置並把他更正?」就是這個問題引導著漢明發現第一個二進制錯誤更正碼。

考慮一個最簡單的線性方塊碼(linear block code)[4]，例如 $(n,k)=(7,4)$ 的漢明碼，輸入 $k=4$ 個信息位元(message bit)，輸出 $n=7$ 個編碼位元。把(4)式重寫如下：

$$1+x^7=(1+x)(1+x^2+x^3)(1+x+x^3)=(1+x+x^2+x^4)(1+x+x^3)=h(x)g(x) \tag{17}$$

其中 $g(x)=1+x+x^3$ 稱為生成多項式(generator polynomial)， $h(x)=1+x+x^2+x^4$ 稱為同位檢查多項式(parity-check polynomial)，讀者應該又注意到 $h(x)$ 的係數以 7 個位元表示正是 1110100，故事的主角又出現了。 $g(x)=1+x+x^3$ 對應到底下(18)式的循環形式的生成矩陣，第一列就是 $g(x)=1+x+x^3$ 的係數，第二列以循環的形式往右邊移動一格，其餘各列依此類推。

$$G'=\begin{bmatrix}1&1&0&1&0&0&0\\0&1&1&0&1&0&0\\0&0&1&1&0&1&0\\0&0&0&1&1&0&1\end{bmatrix} \tag{18}$$

如果我們希望信息位元(message bit)在編碼後依然會出現在編碼完成的字元裡，可以把(18)式作列運算，得到 (19) 式，由於 G 矩陣中後半部的 4×4 元素形成單位矩陣，這樣的處理我們稱為系統編碼(systematic code)。

$$G=\begin{bmatrix}1&1&0&1&0&0&0\\0&1&1&0&1&0&0\\1&1&1&0&0&1&0\\1&0&1&0&0&0&1\end{bmatrix}=\begin{bmatrix}P & \vdots & I_k\end{bmatrix} \tag{19}$$

利用相同的處理技巧也可以把同位檢查多項式 $h(x)$ 轉成同位檢查矩陣 H ，由於已經有(19)式，透過 $HG^T = 0$ 的關係可以更快的直接寫出如下式：

$$H = \left[I_{n-k} \quad \vdots \quad P^T \right] = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix} \tag{20}$$

其中 T 表示矩陣的轉置，特別的是每一列都是 1110100 的一個逆循環。

改錯碼的最短漢明距離(Hamming distance)決定了可以改幾個位置的錯誤，要如何求得最短漢明距離？其中一種做法是先計算 H 矩陣最少需幾個行向量相加才能等於 0？透過觀察知道是 3 個，例如把第 1，2，4 行相加得

$$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{21}$$

我們說最短的漢明距離 $d_{\min} = 3$，一個線性方塊碼可以改 t 個位置的錯誤由下式決定，

$$t \le fix[\frac{1}{2}(d_{\min} - 1)] \tag{22}$$

其中 $fix[\bullet]$ 表示取整數的運算，由於 $t \le fix[\frac{1}{2}(3-1)] = 1$ ，因此 $(n,k) = (7,4)$ 的漢明碼可以修正一個位元的錯誤。

底下我們藉由一個實際的例子來說明改錯的過程。假設信息位元 $\vec{m} = [1\,0\,1\,0]$ ，編碼結果如下：

$$\vec{c} = \vec{m}G = [1 \quad 0 \quad 1 \quad 0] \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} = [0 \quad 0 \quad 1 \quad 1 \quad 0 \quad 1 \quad 0] \tag{23}$$

用通俗的說法，$\vec{m} = [1\,0\,1\,0]$是重要人物，我們另外派$[0\,0\,1]$做前導來保護他們。假設經由通訊過程沒有發生錯誤，即收到的碼 $\vec{r} = \vec{c}$，我們可以計算它的病徵(syndrome)向量如下：

$$\vec{s} = H\vec{r}^{\,T} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{24}$$

由於病徵向量 $\vec{s}$ 為 0 向量，這時候我們可以說接收到的信息很大的機率是對的。假設收到的信息錯了一個位元，例如 $\vec{r} = [0 \quad 0 \quad 1 \quad 1 \quad 0 \quad 1 \quad 1]$，同樣計算它的病徵向量如下：

$$\vec{s} = H\vec{r}^{\,T} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} \tag{25}$$

比較病徵向量 $\vec{s}$ 與同位檢查矩陣 $H = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}$，發現病徵向量與同位檢查矩陣的第 7 行向量相同，因此我們知道收到信息的第 7 個位元是錯的，經過改正後得知正確的信息是 $\vec{m} = [1\,0\,1\,0]$。

最後我們再回到游森棚教授提到的益智遊戲，把它改成：能否將 $n = 2^m$ ($m \in N$)個 0 或 1 排成一圈，使得讀 m 個相鄰位元剛好出現 $000\cdots0, 000\cdots1, ..., 111\cdots1$ 各一次? 答案是可以的，因為不管 m 是多少都會有最大長度序列，最後補一個 0 再圍成一圈就完成任務了。這類問題與圖論中的尤拉圖形有關，有興趣的讀者可參考 Wilson 的著作(Wilson, et al. , 1976)關於鼓輪問題(rotating drum problem)即可獲得滿意的解答。

文後特別感謝審稿專家提出文章敘述的缺陷，費心修改並提供適當的參考文獻。

參考文獻

翁萬德、江松茶、翁健二（編譯）(2015)。**通訊系統**（國際版第五版）。全華書局。
游森棚（2018）。從數學競賽到數學研究—從 2002-C6 說起。**數學傳播**，**42**(4)，32-45。
Haykin, S. (2000). *Communication systems*. John Wiley & Sons.
Holmes, J. K. (1982). *Coherent spread spectrum systems*. John Wiley & Sons.
Wilson, R. J., & Watkins, J. J. (1976). *Graphs, an introductory approach*. John Wiley & Sons.