中華民國第55屆中小學科學展覽會
作品說明書

國小組　數學科

第二名

080401

青蛙棋棋盤和棋譜的研究

學校名稱：桃園市大溪區仁和國民小學

作者：	指導老師：
小五 王俞量	鄧達鈞
小五 吳立宇	林徹輝
小五 郭庭妤	
小五 張雯棋	
小五 鄧又宸	

關鍵詞：翻轉元素、被跳數、實位

摘要：在暑假數學育樂營的時候，我們接觸了青蛙棋。我們發現關卡應不只卡片上的 40 關，應該還有很多其他的關卡。所以我們用窮舉法列出 2 顆時全部的關卡。後來我們又發現可以用 3333→12 的方法找到 12 顆的破關方式。最後我們利用網路資源找到了電腦運算的程式，經過修改，我們列出了所有關卡的全部破關方式，可是列出結果的解讀又需要費一番功夫。

壹、研究動機

我們在 5 上的時候接觸了一個好玩的遊戲「青蛙棋」。遊戲的目的是透過移動棋子將紅青蛙留到最後，並消滅其他的綠青蛙。

雖然遊戲中有許多關卡，但我們並不滿足於將所有的關卡破關而已。我們想：「是不是還有其他的關卡？12 顆棋子的關卡只有解答這一種嗎？」為了能夠讓這個遊戲有更多的變化，而且找出更多的破關方法，我們開始了這次的研究。

貳、研究目的

一、找出棋盤和棋譜間的規律
二、哪些圖形可以成為關卡？
三、找出破關 12 顆的方法
四、利用電腦運算出全部的破關方式

參、研究設備與器材

白紙、筆、青蛙棋、電腦和 Excel

肆、研究過程與方法

一、遊戲說明
（一）棋盤上有 13 個點，可在棋盤上放置 2～12 隻青蛙，每次都要放紅青蛙
（二）消去綠青蛙的方法：
1. 綠青蛙或紅青蛙跳過綠青蛙，被跳過的綠青蛙即被消去
2. 綠青蛙不能跳過紅青蛙

（三）只能用跳躍移動，不可直接移動到旁邊一格
（四）當棋盤上只剩紅青蛙時就算過關

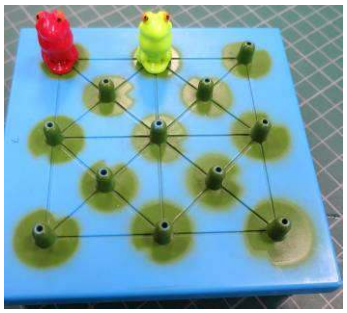
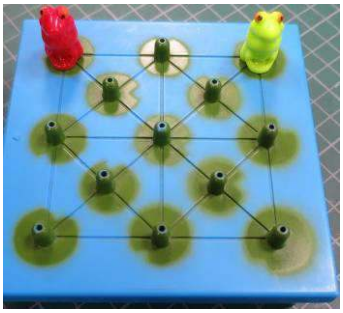
過關：紅青蛙跳過綠青蛙之後，就只剩下一隻紅青蛙。	不能過關：紅青蛙或綠青蛙不能用「走到隔壁一格」的方式移動，當最後 2 顆有間隔，就一定不能破關。
	

表 4-1-1

二、名詞定義

（一）翻轉元素：經過翻和轉可以讓不同的圖形變成相同的圖形。各有 4 種轉法和 4 種翻法。

（二）被跳數和跳數：在棋譜中 a0 跳過 b0 會剩下 a1 的棋子。我們稱為 a0*b0=a1，a0 為跳數，b0 為被跳數。

（三）實位和空位：有放棋子的位置是實位，沒放棋子的位置是空位。

（四）窮舉法：將全部的數字列出來後組合，並確認是否可以成為棋譜的方法。

（五）樹枝法：我們將 2 棵的公式全部列出並製作成道具，再用取代的方法，每次讓 2 顆取代 1 顆。這樣就可以推算出破關的方法。因為最後呈現的結果很像樹枝的形狀，所以稱此方法為樹枝法。

（六）3333→12 表示 3 顆 1 組共 4 組合成為 12 顆，說明可參考表 4-6-6，以此類推 111→4 表示經過 3 次替換成為 4 顆，22→4 表示 2 顆 1 組共 2 組合成為 4 顆，如(a0*b3)*(b2*c2)。

（七）n 階 N 邊形棋盤：3 階 4 邊形棋盤表示有 3 個四邊形在棋盤中，中間那個點也算一個 4 邊形。

三、文獻探討

（一）相關資料分析

出處	題目	內容
臺北市第 42 屆中小學科學展覽會國小組數學	搬新家-一人跳**棋**之毛蟲**棋**探討	移位遊戲的記錄
中華民國第 51 屆中小學科學展覽會國小組數學科	三十六計，『跳』為上	當移動過程符合跳島戰術，將可得到最少移動次數步驟

表 4-3-1

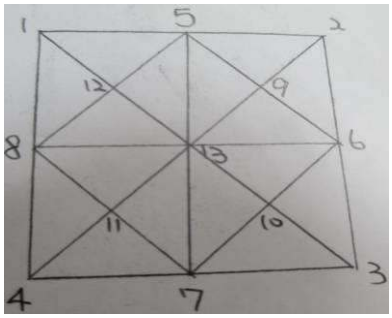
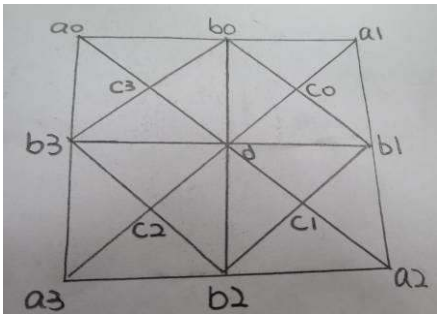
（二）我們的研究特色

1. 找出翻轉元素和整數間的巧妙關係
2. 利用三種方法找出破關的方式
3. 找出最簡單的破關方法
4. 利用電腦將全部的破關方式列出
5. 發現實位和空位的關係

四、研究 1：棋盤和棋譜間的規律

（一）棋盤上各點的命名與棋盤的擴大

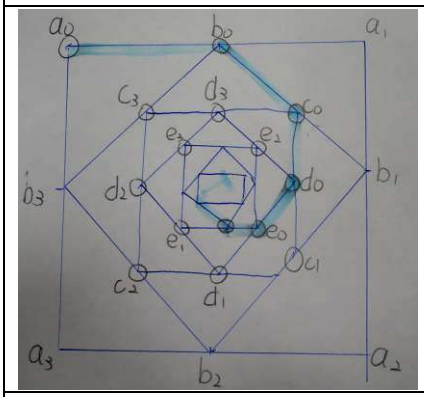
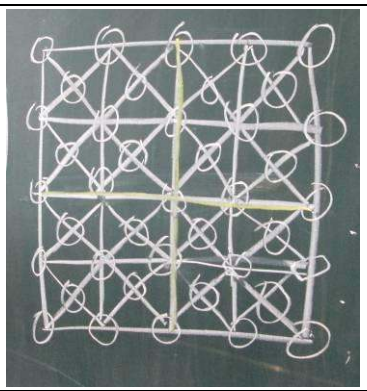
1. 命名：我們將棋盤上的 13 個點用 1～13 和 a0~a3，bo~b3，c0~c3，d 依序命名。下表為表 4-4-1

以數字 1～13 命名	以 a0~a3，bo~b3，c0~c3，d 依序命名
	
1. 最開始的命名 2. 使用窮舉法時較方便，能夠列出全部組合 3. 每 4 個數字一個循環。1->2->3->4->1	1.因使用樹枝法而發展出來 2.可以歸納出 2 顆共 32 組公式 3.方便分類，英文字母 abcd 各一類

2. 棋盤擴大：根據現在看到的規律，我們推估如果要擴大棋盤應該是有兩種情況

（1） 正方形的邊長中點擴大法：每個正方形的第一個點連接起來呈現順時針螺旋狀。

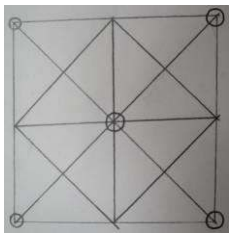
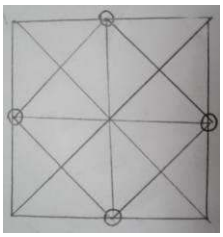
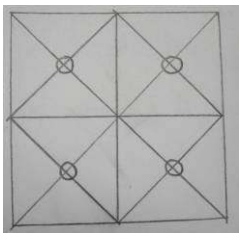
（2） 4 個正方形合併法：每個點間的路徑長不改變

正方形的邊長中點擴大法	4 個正方形合併法
	
.從原本的棋盤可以看到大中小 3 個正方形，每個正方形的邊長中點會發展出更大的正方形。	從原本的棋盤可以看出是 4 個一樣的正方形合併。我們用 4 個一樣的棋盤再合併成更大的棋盤。

上表為 4-4-2：棋盤擴大的方法

（二）棋盤上各點之間的關係：

1. 各點之間呈封閉性：我們發現 3 個集合（1，2，3，4，13）（5，6，7，8）（9，10，11，12），集合間互相不能交流。

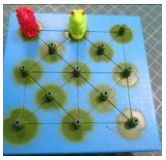
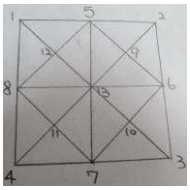
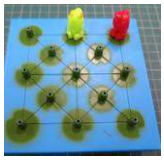
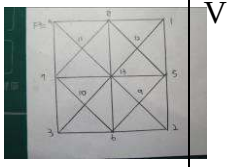
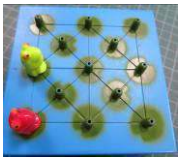
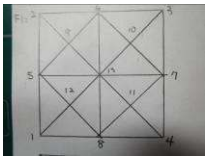
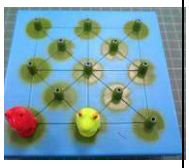
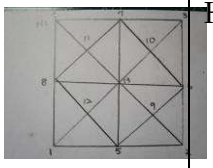
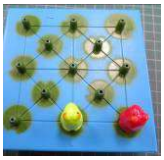
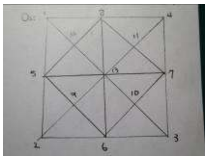
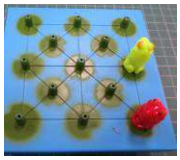
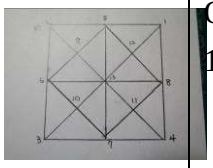
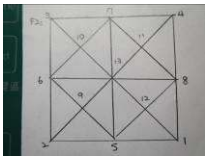
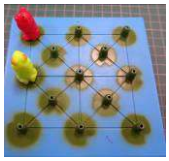
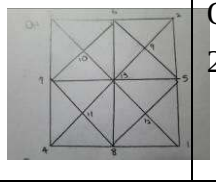
（1，2，3，4，13）	（5，6，7，8）	（9，10，11，12）
		

上表為 4-4-3：共有 3 種棋子位置的集合，各集合之間不能交流。

2. 某些排列出現時，一定不能破關：玩了許多次之後，我們發現出現特定圖形的時候就一定不能破關。如果將這些不能破關的圖形統計起來，在破關時避免出現這種圖形，那就能增加破關機率。

（三）相同棋譜的辨識：我們發現有些圖形看起來不一樣，實際上是一樣的。以 2 顆的圖形為例子：

1. 翻轉元素：我們發現 2 顆圖形有 8 種，根據翻或轉可以形成，統計如下表

翻轉方式	棋盤圖形	數字變化	簡稱	翻轉方式	棋盤圖形	數字變化	簡稱
不翻也不轉動			F0	水平翻			V
逆時針轉 90 度			F1	垂直翻			H
逆時針轉 180 度			F2	左上右下翻			O1
逆時針轉 270 度			F3	右上左下翻			O2

上表為 4-4-4：翻轉元素的呈現

2. 翻轉元素的加法運算：翻轉元素之間可以相加，而且翻轉元素怎麼相加就是那 8 個，讓人覺得非常有趣。

加數＼被加數	F0	F1	F2	F3	V	H	O1	O2
F0	F0	F1	F2	F3	V	H	O1	O2
F1	F1	F2	F3	F0	O2	O1	V	H
F2	F2	F3	F0	F1	H	V	O2	O1
F3	F3	F0	F1	F2	O1	O2	H	V
V	V	O1	H	O2	F0	F2	F1	F3
H	H	O2	V	O1	F2	F0	F3	F1
O1	O1	H	O2	V	F3	F1	F0	F2
O2	O2	V	O1	H	F1	F3	F2	F0

左邊是表 4-4-5：翻種元素的運算。最上列（藍色字）是被加數，最左邊欄位是加數。我們可以看到翻轉元素呈現的結果非常有規律。另外轉元素和翻元素之間沒有交換性，但翻集合和轉集合自己有交換性。

Sec

3. 套用翻轉元素對數字的影響：翻轉元素就像是一個機器，你把東西丟進去，有些出來的產品一樣，有些不一樣。我們把翻轉元素對數字的影響統計在下表，方便以後使用。

（1） 在各翻轉元素下 1～13 的變化：

	F0	F1	F2	F3	V	H	O1	O2
1	+0	+1	+2	+3	+1	+3	+2	+0
2	+0	+1	+2	-1	-1	+1	+0	+2
3	+0	+1	-2	-1	+1	-1	-2	+0
4	+0	-3	-2	-1	-1	-3	+0	-2
5	+0	+1	+2	+3	+0	+2	+1	+3
6	+0	+1	+2	-1	+2	+0	-1	+1
7	+0	+1	-2	-1	+0	-2	+1	-1
8	+0	-3	-2	-1	-2	+0	-1	-3
9	+0	+1	+2	+3	+3	+1	+0	+2
10	+0	+1	+2	-1	+1	-1	+2	+0
11	+0	+1	-2	-1	-1	+1	+0	-2
12	+0	-3	-2	-1	+3	-1	-2	+0
13	0	0	0	0	0	0	0	0

表 4-4-6：不同數字在各翻轉元素下的改變

（2） 在各翻轉元素下 a0～d 的變化

	F0	F1	F2	F3	V	H	O1	O2
a0	a0	a1	a2	a3	a1	a3	a2	a0
a1	a1	a2	a3	a0	a0	a2	a1	a3
a2	a2	a3	a0	a1	a3	a1	a0	a2
a3	a3	a0	a1	a2	a2	a0	a3	a1
b0	b0	b1	b2	b3	b0	b2	b1	b3
b1	b1	b2	b3	b0	b3	b1	b0	b2
b2	b2	b3	b0	b1	b2	b0	b3	b1
b3	b3	b0	b1	b2	b1	b3	b2	b0
c0	c0	c1	c2	c3	c3	c1	c0	c2
c1	c1	c2	c3	c2	c2	c0	c3	c1
c2	c2	c3	c2	c1	c1	c3	c2	c0
c3	c3	C0	c1	c0	c0	c2	c1	c3
d	d	d	d	d	d	d	d	d

表 4-4-7：不同位置在各翻轉元素下的改變

（四）翻轉元素和整數的相似處

1. 封閉性
2. 交換性（翻轉元素部分有）
3. 結合性
4. 都有 1 和倒數

五、研究 2 哪些圖形可以成為關卡：如同上述研究所發現，某些特定圖形是一定可以破關，某些特定圖形出現表示破關失敗，我們想更進一步研究出，剩下少數顆時如何一眼看出這圖形是一定可以破關的。

（一）最多有幾種關卡：

1. 從選擇方法討論：

顆數	原本數量	重複數	關卡數量	經過翻轉後的數量
0	0	1	0	0
1	13	1	13	4
2	156	2	78	5
3	1716	6	286	?
4	17160	24	715	?
5	154440	120	1287	?
6	1235520	720	1716	?
7	8648640	5040	1716	?
8	51891840	40320	1287	?
9	259459200	362880	715	?
10	1037836800	3628800	286	?
11	3113510400	39916800	78	5
12	6227020800	479001600	13	4
13	6227020800	6227020800	0	0

表 4-5-1：各顆數量可能產生的關卡數量

13+78+286+715+1287+1716+1716+1287+715+286+78+13＝8190

2. 除去上面的討論外，我們應該還要考慮到翻轉元素會產生的相同結果。所以可能關卡的數量應該落在 8190÷8（最保守估計，每個關卡都有對應的翻轉關卡）到 8190 之間。

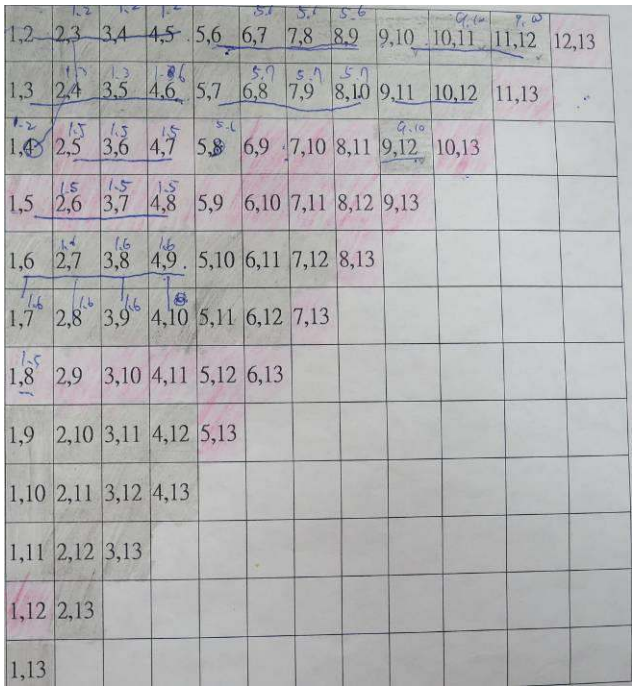
（二）利用窮舉法列出 2 顆和 3 顆的全部情況

1. 2 顆的情況

（1） 在 78 種可能中有 28 種圖形可以成為棋譜

（2） 經過翻轉元素之後我們發現事實上只有 5 種圖形。

我們將所有 2 種的可能列出，並將可以破關的圖形圖上紅色（如下圖）	經過翻轉元素後，我們發現實際只有 5 種圖形（如下圖），同顏色代表同一種圖形



1,2	2,3	3,4	4,5	5,6	6,7	7,8	8,9	9,10	10,11	11,12	12,13
1,3	2,4	3,5	4,6	5,7	6,8	7,9	8,10	9,11	10,12	11,13	
1,4	2,5	3,6	4,7	5,8	6,9	7,10	8,11	9,12	10,13		
1,5	2,6	3,7	4,8	5,9	6,10	7,11	8,12	9,13			
1,6	2,7	3,8	4,9	5,10	6,11	7,12	8,13				
1,7	2,8	3,9	4,10	5,11	6,12	7,13					
1,8	2,9	3,10	4,11	5,12	6,13						
1,9	2,10	3,11	4,12	5,13							
1,10	2,11	3,12	4,13								
1,11	2,12	3,13									
1,12	2,13										
1,13											

表 4-5-2：2 顆棋子的全部組合。左邊紅色部分是可以形成關卡數，右邊可以看出在 2 顆時共有 5 種關卡。

2. 3 顆的情況：我們也列出了 3 顆所有的可能，並將可以形成關卡的圖形都塗上紅色，如下圖。

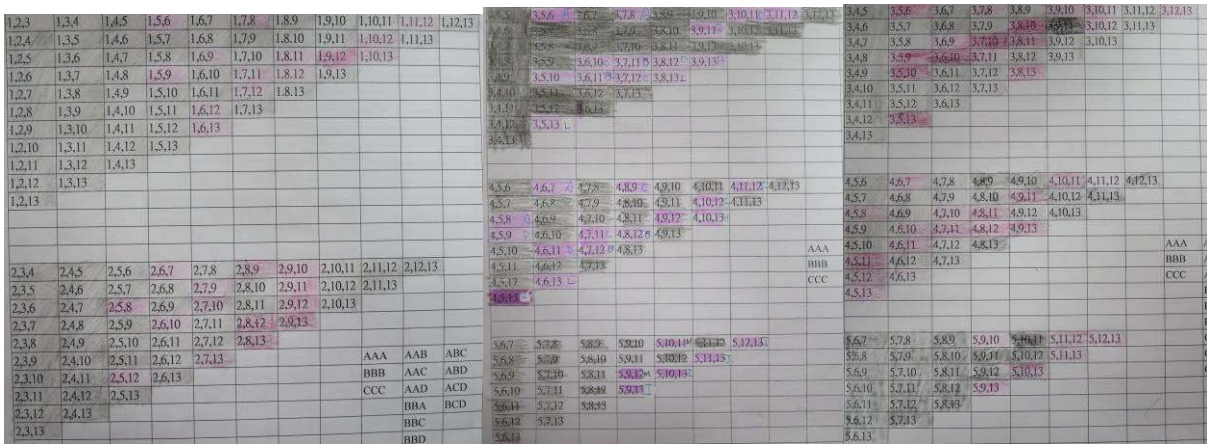


圖 4-5-1：3 顆棋子的全部組合。左邊紅色部分是可以形成關卡數

3. 是否更簡單的方法找出棋譜數量：利用窮舉法找出全部的組合後再一一確認是否可以作為棋譜，然後再經過翻轉元素找出重複的圖形，這個過程實在是太漫長了。所以我們思考是否有更簡單的方法。

（三）利用方向圖示找出 2 顆～4 顆的關卡：我們發現 2 顆的棋子跳法可以用方向來做區分，可以分成 8 種方向。利用方向圖示可以更輕鬆的找出 2 顆～4 顆全部棋譜

1. 棋盤圖形與方向圖示對照表

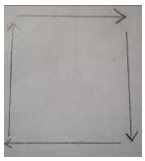
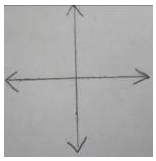
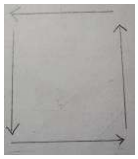
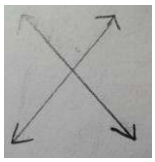
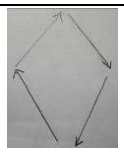
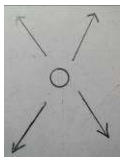
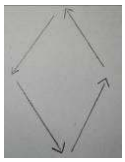
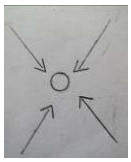
跳棋方向	一般式	棋盤上的結構	方向圖示	跳棋方向	一般式	棋盤上的結構	方向圖示
a0*b0 a1*b1 a2*b2 a3*b3	an*bn =a(n+1)	大正方形（順時針）		b0*d b1*d b2*d b3*d	bn*d =b(n+2)	大十字	
a1*b0 a2*b1 a3*b2 a0*b3	an * b(n-1) =a(n-1)	大正方形（逆時針）		c0*d c1*d c2*d c3*d	cn*d =c(n+2)	小十字	
b0*c0 b1*c1 b2*c2 b3*c3	bn*cn =b(n+1)	小正方形（順時針）		d*c0 d*c1 d*c2 d*c3	d*cn =an	d 開放	
b1*c0 b2*c1 b3*c2 b0*c3	b(n+1)*cn =bn	小正方形（逆時針）		a1*c0 a2*c1 a3*c2 a0*c3	an+1*cn=d	d 收縮	

表 4-5-3：棋盤圖形與方向圖示對照表

六、研究 3 找出破關 12 顆的方法

（一）利用棋盤特性產生的 2 顆公式

終點	a0	b0	c0	d
公式	a1*b0 d*c3 a3*b3	b3*c3 b2*d b1*c0	c2*d	a0*c3 a1*c0 a2*c1 a3*c2
終點	a1	b1	c1	
公式	a2*b1 d*c0 a0*b0	b0*c0 b3*d b2*c1	c3*d	
終點	a2	b2	c2	
公式	a3*b2 d*c1 a1*b1	b1*c1 b0*d b3*c2	c0*d	
終點	a3	b3	c3	
公式	a0*b3 d*c2 a2*b2	b2*c2 b1*d b0*c3	c1*d	

表 4-6-1：2 顆共產生共 32 種公式

（二）樹枝法的形成：我們將上述 32 種公式印出後裁切。並利用取代的方法形成更多顆的公式（破關方法）

1. 製成道具
2. 取代：我們發現所有 2 顆都可以透過取代的方法發展出更多顆數，舉例如下：

（1） a1*bo=(a2*b1)*b0=(a2*b1)* (b2*d)過程如下表

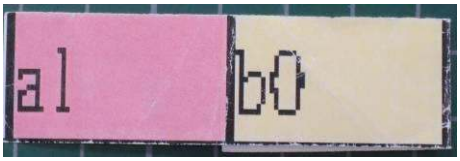
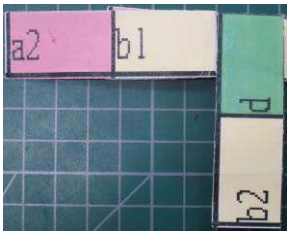
第一步 a1*bo	第二步(a2*b1)取代 a1	第三步(b2*d)取代 b0
		

表 4-6-1：a1*bo=(a2*b1)*b0=(a2*b1)* (b2*d)的產生過程。反過來就是破關方法

（2） b1*c0= b1*(c2*d)= b1*(c2* (a0*c3))= (b2*c1) *(c2* (a0*c3))

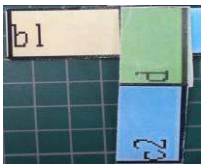
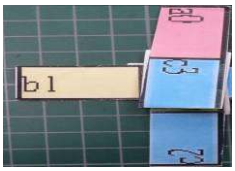
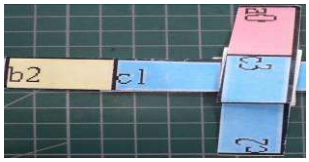
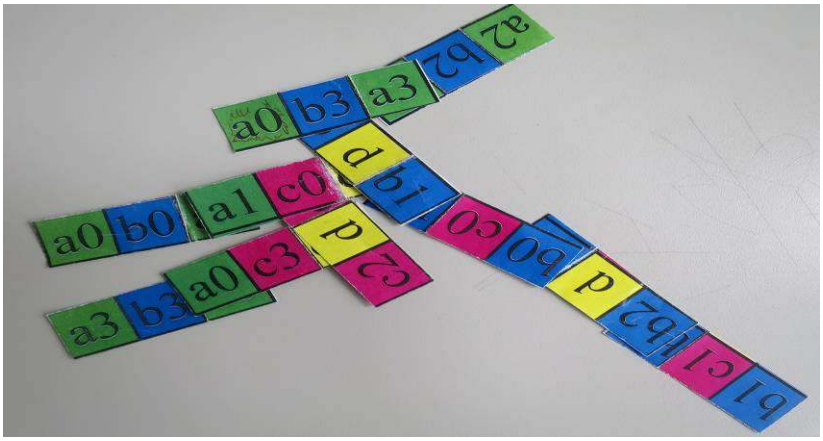
第一步 b1*c0	第二步(c2*d)取代 c0	第三步(a0*c3)取代 d	第四步(b2*c1)取代 b1
			

表 4-6-2：b1*c0= b1*(c2*d)= b1*(c2* (a0*c3))= (b2*c1) *(c2* (a0*c3))的產生過程。

( )外面本應加[ ]，但因後面研究發現括號種類不敷使用，所以我們都用（ ），使用原則和一般計算整數加減時一樣，最裡面的（ ）先消除，反過來說就是最後面建立的（ ）先消除。

3. 樹枝法解答：經過多次的實驗，終於發現樹枝法的解答（如下圖），將解答公式記錄下來：(a2*b2)*((((b1*c1)*d)*c0)*((a0*b0)*(c2*((a3*b3)*c3))))



4. 假設中的破關方法：我們利用樹枝法將全部的四顆可能列出來

（1）4 顆全部公式表

(a0*b0)*(b3*c3)	(a0*b3)*(b0*c3)	a1*((b1*c1)*d)	(a3*c2)*(c1*d)
(a0*b0)*(b1*c0)	(a0*b3)*(b1*d)	a1*((b3*c2)*d)	a1*(b1*(c2*d))
(a0*b0)*(b2*d)	((a0*b3)*c2)*c3	(a2*(b0*d))*b3	(a1*(c2*d))*c3
((a0*b0)*c0)*c3	(d*c2)*(b0*c3)	a3*((b0*d)*c2)	((a0*c3)*c2)*b3
(d*c0)*(b3*c3)	((a1*b0)*c3)*c3	a1*((b2*c2)*c3)	a3*(b1*(a0*c3))
((a1*b1)*b2)*b3	a3*((b2*c1)*d)	a1*((b1*d)*c3)	((a0*c3)*c0)*b0
((a1*b1)*c1)*c3	(a2*(b2*c1))*b0	(a2*b1)*(b3*c3)	a1*(b2*(a0*c3))
((a3*b2)*b1)*b0	a1*((b2*c1)*c0)	a3*((b1*c0)*c3)	((a1*c0)*c2)*b3
((a3*b2)*c1)*c3	a3*((b0*c0)*d)	a3*((b2*d)*c3)	a3*(b1*(a1*c0))
((d*c1)*b2)*b3	(a2*(b3*d))*b0	(a2*b1)*(b2*d)	((a2*c1)*c2)*b3
((d*c1)*b1)*b0	a1*((b3*d)*c0)	a3*(b2*(c0*d))	a3*(b1*(a2*c1))
(a2*b2)*(b0*c3)	(a2*(b1*c1))*b3	(a3*(c0*d))*c3	((a2*c1)*c0)*b0
(a2*b2)*(b1*d)	a3*((b1*c1)*c2)	a3*(b0*(c1*d))	a1*(b2*(a2*c1))
((a2*b2)*c2)*c3	((a3*c2)*c0)*b0	a1*(b3*(c1*d))	
(a0*b3)*(b2*c2)	a1*(b2*(a3*c2))	(a1*c0)*(c1*d)	

表 4-6-3：4 顆全部公式表

（2）111→4 和 22→4：我們發現 4 顆的種類可以分成 2 種。一種是每次都取代 1 顆，取代 2 次就變成 4 顆，例如 a1*(b3*(c1*d))，我們稱為 111→4。還有一種是同時取代前後 2 顆，例如(a0*b3)*(b2*c2)，我們稱為 22→4。

（3）大膽的假設：從 4 顆的全部解我們獲得一個啟示，那就是可以多個公式合併成一個公式。我們假設：大部分的顆數都可以用多顆合併。例如 23→5 或 222→6 又或者 66→12

（三）找出 12 棵的破關方法

1. 3333 →12：根據上面的研究，我們認為可以用 4 種 3 顆的公式合成 12 顆的公式。

（1） 列出全部 3 顆公式：我們將 a0，b0，c0，d 3 顆的公式全部列出，其他則只要經過翻轉元素又可以知道了。

終點:a0	終點:b0	終點:c0	終點:d
a2*b2*b3	b1*c0*d	c2*(a0*c3)	a1*b0*c3
d*c2*b3	b3*c2*d	c2*(a2*c1)	a3*b3*c3
a2*b1*b0	b2*c2*c3	c2*(a1*c0)	a0*b0*c0
d*c0*b0	b1*d*c3		a2*b1*c0
a1*c0*c3	b3*d*c0		a1*b1*c1
a2*c1*c3	b2*c1*c0		a3*b2*c1
a3*c2*c3	b2*(a0*c3)		a0*b3*c2
a3*(b1*d)	b2*(a1*c0)		a2*b2*c2
a3*(b0*c3)	b2*(a2*c1)		a0*(c1*d)
a3*(b2*c2)	b2*(a3*c2)		a1*(c2*d)
a1*(b2*d)	b3*(c1*d)		a2*(c3*d)
a1*(b3*c3)	b1*(c2*d)		a3*(c0*d)
a1*(b1*c0)			

表 4-6-4：3 顆公式統計表

從左表我們可以列出 a0，bo，co，d 的 3 顆公式的合成種類，如下表。如此當我們合成 3333 時就比較方便

a0	b0	c0	d
abb	bcd	acc	abc
bcd	bcc		acd
acc	abc		
abd			
abc			

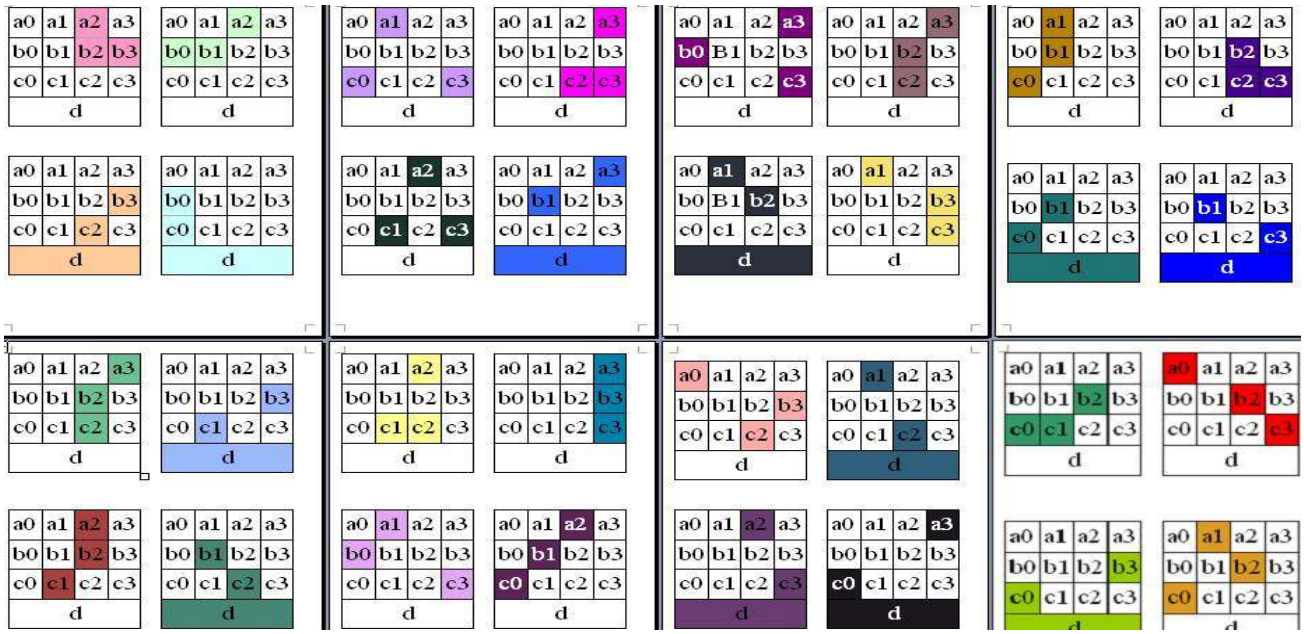
（2） 4 顆的可能：根據表 4-6-3 我們可以列出 4 顆的全部種類，再和 3 顆公式做結合

三同一異	二同二異	四異
abbb	abbc abbd accd aabc abcc bccd bbcd	abcd

表 4-6-5：4 顆種類統計表

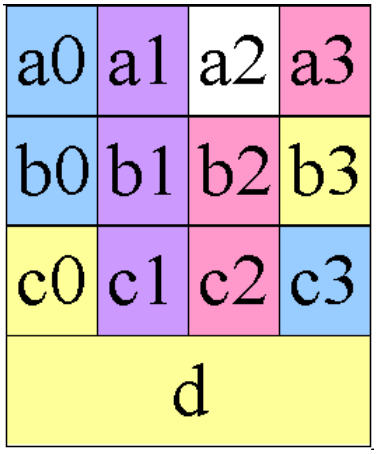
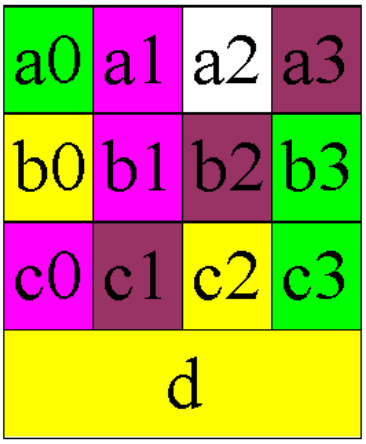
（3） accb 不能用：4 顆的公式裡有 accb 的種類，但如果用 cc 會用掉全部的 c。而剩下的 a 或 b 就不能用 c 了。所以 accb 不能用。

2. 利用區塊道具：我們將 3 顆的公式全部製成區塊圖形如下圖



然後任選其中 4 種配合表 4-6-5 做結合，如能成功湊出 12 顆而沒重複，那我們就成功了。

3. 找到 3333→12 的解答：經過我們許多次的嘗試之後。我們發現了 2 個解答，如下表。

解答區塊		
公式	((a3*(b2*c2))*((b3*d)*c0))* ((b0*(a0*c3))*((a1*b1)*c1))	((a1*(b1*c0))*((b0*d)*c2))* ((b3*(a0*c3))*((a3*b2)*c1))

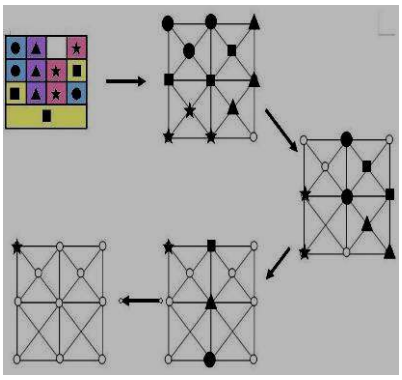
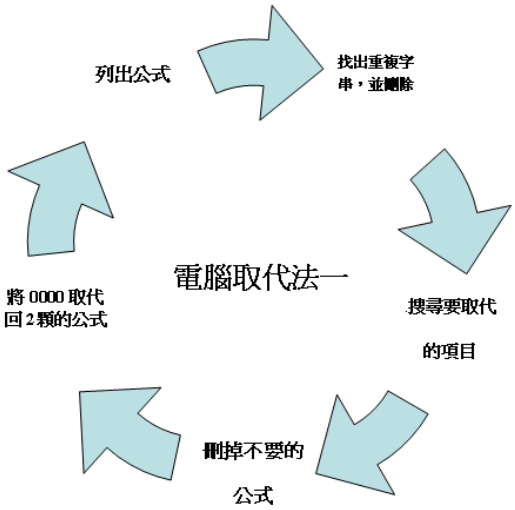


表 4-6-6：3333→12 區塊公式解答，解答過程如右圖

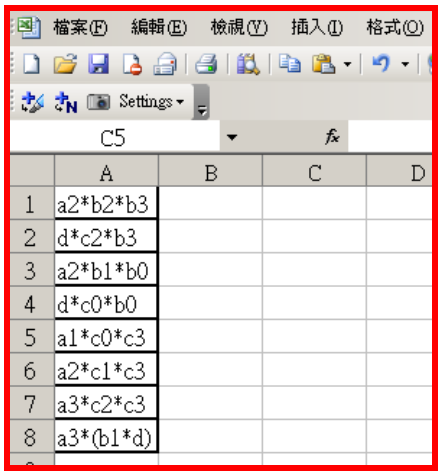
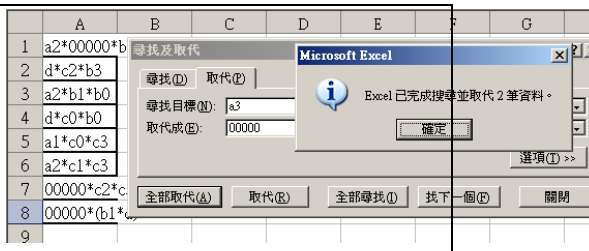
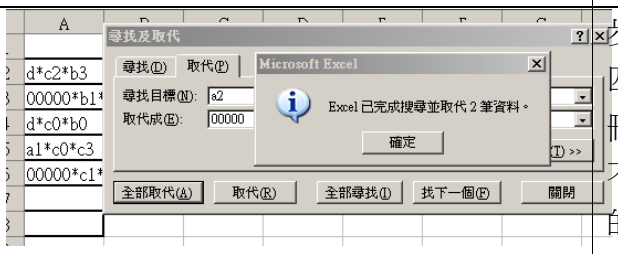
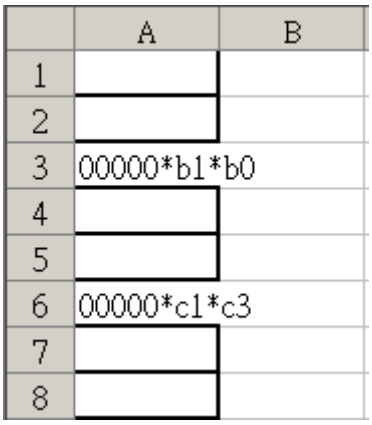
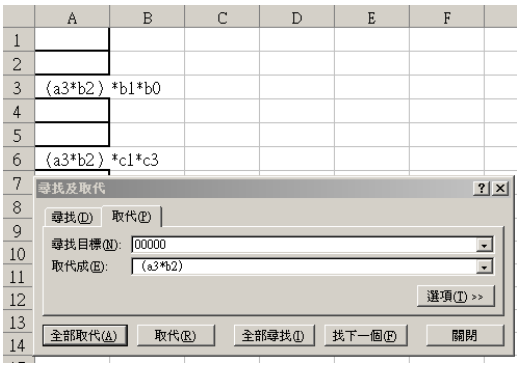
七、研究 4 利用電腦運算出全部的破關方式

（一）簡單而重複的電腦取代法

1.思考邏輯：我們用簡單而原始的方法把 2 顆取代為 4 顆，如右圖



2.取代詳細步驟：如下表，以 a0 的 3 顆->4 顆為例，我們用（a3*b2）取代 a2

步驟一: 列出公式		步驟二：找出重複字串，並刪除	
步驟三：搜尋取代項目（用 00000 取代 1 顆：比較明顯）		步驟四：刪掉不要的公式	
步驟五：將 0000 取代回 2 顆的公式			

（二）上網尋求協助：我們發現上面的方法還是太過複雜，希望能有更簡單快速的方法，所以我們上網查詢是否能將過程再度簡化。

1.先把過程列出：為了能夠更清楚我們需要什麼樣的功能，我們把流程修改後列出：

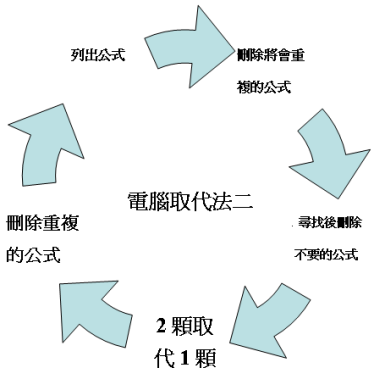
（1）列出公式

（2）刪除將會重複的公式

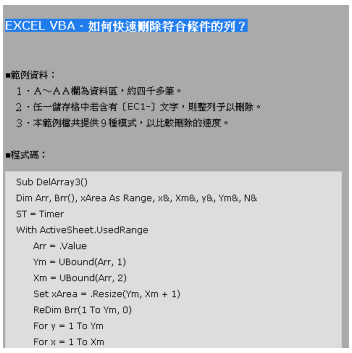
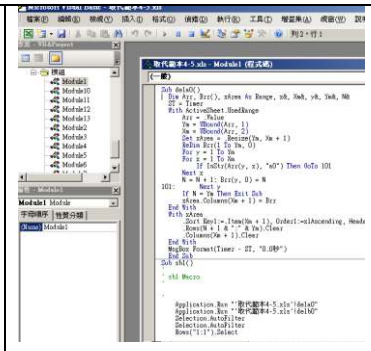
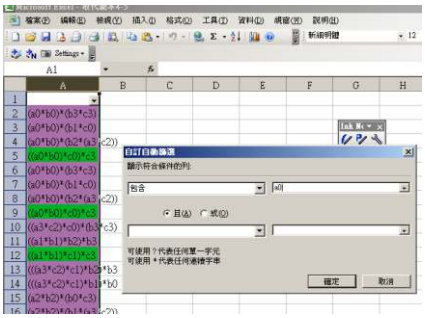
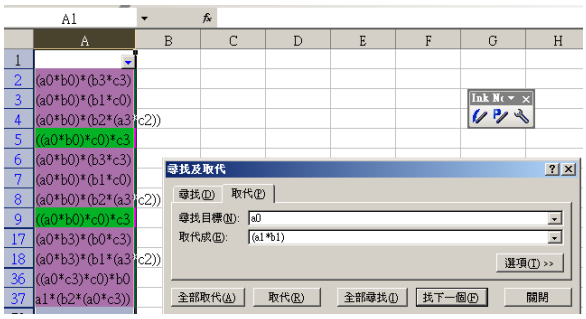
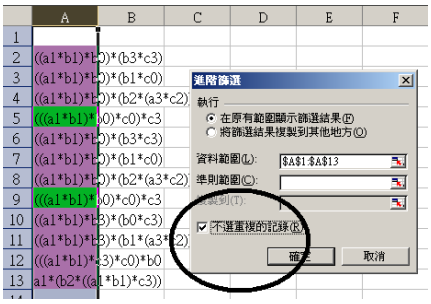
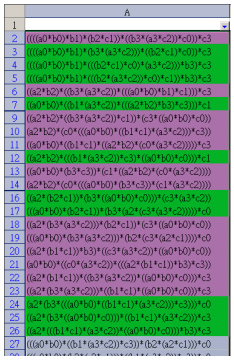
（3）尋找後刪除不要的公式（篩選）

（4）2 顆取代 1 顆

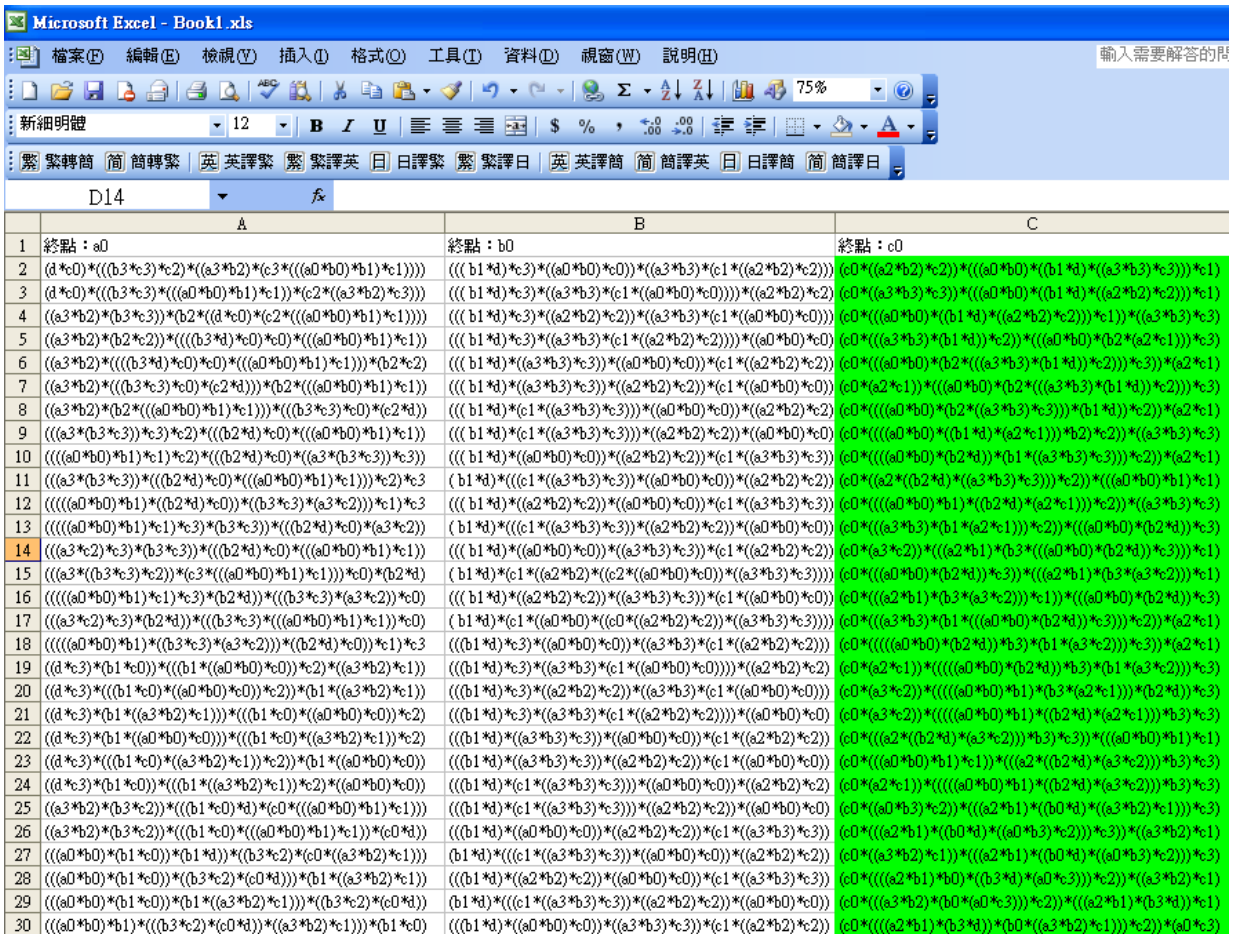
（5）刪除重複的公式



2. 製作巨集與解答詳細步驟

步驟一：尋找編輯巨集的方法		步驟二：編輯巨集，刪除會重複的字串	
步驟三：篩選，留下被取代字串		步驟四：2 顆取代 1 顆	
步驟五：將重複的資料刪除		產生結果	

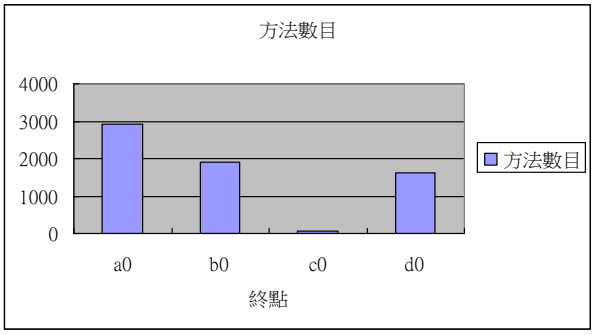
（三）產生結果：根據上面電腦產生的結果，我們發現終點是 a0~a3 的方法最多，終點是 c0~c3 的方法最少（如下表、圖）。



	A	B	C
1	終點：a0	終點：b0	終點：c0
2	(d*c0)*(((b3*c3)*c2)*((a3*b2)*(c3*(((a0*b0)*b1)*c1))))	(((b1*d)*c3)*((a0*b0)*c0))*((a3*b3)*(c1*((a2*b2)*c2)))	(c0*((a2*b2)*c2))*(((a0*b0)*(b1*d)*((a3*b3)*c3)))*c1)
3	(d*c0)*(((b3*c3)*(((a0*b0)*b1)*c1))*(c2*((a3*b2)*c3)))	(((b1*d)*c3)*((a3*b3)*(c1*((a0*b0)*c0))))*((a2*b2)*c2)	(c0*((a3*b3)*c3))*(((a0*b0)*(b1*d)*((a2*b2)*c2)))*c1)
4	((a3*b2)*(b3*c3))*(b2*((d*c0)*(c2*(((a0*b0)*b1)*c1))))	(((b1*d)*c3)*((a2*b2)*c2))*((a3*b3)*(c1*((a0*b0)*c0)))	(c0*(((a0*b0)*(b1*d)*((a2*b2)*c2)))*c1))*((a3*b3)*c3)
5	((a3*b2)*(b2*c2))*((((b3*d)*c0)*c0)*(((a0*b0)*b1)*c1))	(((b1*d)*c3)*((a3*b3)*(c1*((a2*b2)*c2))))*((a0*b0)*c0)	(c0*(((a3*b3)*(b1*d))*c2))*(((a0*b0)*(b2*(a2*c1)))*c3)
6	((a3*b2)*(((b3*d)*c0)*c0)*(((a0*b0)*b1)*c1)))*(b2*c2)	(((b1*d)*((a3*b3)*c3))*((a0*b0)*c0))*(c1*((a2*b2)*c2))	(c0*(((a0*b0)*(b2*(((a3*b3)*(b1*d))*c2)))*c3))*(a2*c1)
7	((a3*b2)*(((b3*c3)*c0)*(c2*d)))*(b2*(((a0*b0)*b1)*c1))	(((b1*d)*((a3*b3)*c3))*((a2*b2)*c2))*(c1*((a0*b0)*c0))	(c0*(a2*c1))*(((a0*b0)*(b2*(((a3*b3)*(b1*d))*c2)))*c3)
8	((a3*b2)*(b2*(((a0*b0)*b1)*c1)))*(((b3*c3)*c0)*(c2*d))	(((b1*d)*(c1*((a3*b3)*c3)))*((a0*b0)*c0))*((a2*b2)*c2)	(c0*((((a0*b0)*(b2*((a3*b3)*c3)))*(b1*d))*c2))*(a2*c1)
9	(((a3*(b3*c3))*c3)*c2)*(((b2*d)*c0)*(((a0*b0)*b1)*c1))	(((b1*d)*(c1*((a3*b3)*c3)))*((a2*b2)*c2))*((a0*b0)*c0)	(c0*((((a0*b0)*(b1*d)*(a2*c1)))*b2)*c2))*((a3*b3)*c3)
10	((((a0*b0)*b1)*c1)*c2)*(((b2*d)*c0)*((a3*(b3*c3))*c3))	(((b1*d)*((a0*b0)*c0))*((a2*b2)*c2))*(c1*((a3*b3)*c3))	(c0*((((a0*b0)*(b2*d))*(b1*((a3*b3)*c3)))*c2))*(a2*c1)
11	(((a3*(b3*c3))*(((b2*d)*c0)*(((a0*b0)*b1)*c1)))*c2)*c3	(b1*d)*(((c1*((a3*b3)*c3))*((a0*b0)*c0))*((a2*b2)*c2))	(c0*((a2*((b2*d)*((a3*b3)*c3)))*c2))*(((a0*b0)*b1)*c1)
12	(((((a0*b0)*b1)*((b2*d)*c0))*((b3*c3)*(a3*c2)))*c1)*c3	(((b1*d)*((a2*b2)*c2))*((a0*b0)*c0))*(c1*((a3*b3)*c3))	(c0*((((a0*b0)*b1)*((b2*d)*(a2*c1)))*c2))*((a3*b3)*c3)
13	(((((a0*b0)*b1)*c1)*c3)*(b3*c3))*(((b2*d)*c0)*(a3*c2))	(b1*d)*(((c1*((a3*b3)*c3))*((a2*b2)*c2))*((a0*b0)*c0))	(c0*(((a3*b3)*(b1*(a2*c1)))*c2))*(((a0*b0)*(b2*d))*c3)
14	(((a3*c2)*c3)*(b3*c3))*(((b2*d)*c0)*(((a0*b0)*b1)*c1))	(((b1*d)*((a0*b0)*c0))*((a3*b3)*c3))*(c1*((a2*b2)*c2))	(c0*(a3*c2))*(((a2*b1)*(b3*(((a0*b0)*(b2*d))*c3)))*c1)
15	(((a3*((b3*c3)*c2))*(c3*(((a0*b0)*b1)*c1)))*c0)*(b2*d)	(b1*d)*(c1*((a2*b2)*((c2*((a0*b0)*c0))*((a3*b3)*c3))))	(c0*(((a0*b0)*(b2*d))*c3))*(((a2*b1)*(b3*(a3*c2)))*c1)
16	(((((a0*b0)*b1)*c1)*c3)*(b2*d))*(((b3*c3)*(a3*c2))*c0)	(((b1*d)*((a2*b2)*c2))*((a3*b3)*c3))*(c1*((a0*b0)*c0))	(c0*(((a2*b1)*(b3*(a3*c2)))*c1))*(((a0*b0)*(b2*d))*c3)
17	(((a3*c2)*c3)*(b2*d))*(((b3*c3)*(((a0*b0)*b1)*c1))*c0)	(b1*d)*(c1*((a0*b0)*((c0*((a2*b2)*c2))*((a3*b3)*c3))))	(c0*(((a3*b3)*(b1*(((a0*b0)*(b2*d))*c3)))*c2))*(a2*c1)
18	(((((a0*b0)*b1)*((b3*c3)*(a3*c2)))*(b2*d)*c0))*c1)*c3	(((b1*d)*c3)*((a0*b0)*c0))*((a3*b3)*(c1*((a2*b2)*c2)))	(c0*(((((a0*b0)*(b2*d))*b3)*(b1*(a3*c2)))*c3))*(a2*c1)
19	((d*c3)*(b1*c0))*(((b1*((a0*b0)*c0))*c2)*((a3*b2)*c1))	(((b1*d)*c3)*((a3*b3)*(c1*((a0*b0)*c0))))*((a2*b2)*c2)	(c0*(a2*c1))*(((((a0*b0)*(b2*d))*b3)*(b1*(a3*c2)))*c3)
20	((d*c3)*(((b1*c0)*((a0*b0)*c0))*c2))*(b1*((a3*b2)*c1))	(((b1*d)*c3)*((a2*b2)*c2))*((a3*b3)*(c1*((a0*b0)*c0)))	(c0*(a3*c2))*(((((a0*b0)*b1)*(b3*(a2*c1)))*(b2*d))*c3)
21	((d*c3)*(b1*((a3*b2)*c1)))*(((b1*c0)*((a0*b0)*c0))*c2)	(((b1*d)*c3)*((a3*b3)*(c1*((a2*b2)*c2))))*((a0*b0)*c0)	(c0*(a3*c2))*(((((a0*b0)*b1)*((b2*d)*(a2*c1)))*b3)*c3)
22	((d*c3)*(b1*((a0*b0)*c0)))*(((b1*c0)*((a3*b2)*c1))*c2)	(((b1*d)*((a3*b3)*c3))*((a0*b0)*c0))*(c1*((a2*b2)*c2))	(c0*(((a2*((b2*d)*(a3*c2)))*b3)*c3))*(((a0*b0)*b1)*c1)
23	((d*c3)*(((b1*c0)*((a3*b2)*c1))*c2))*(b1*((a0*b0)*c0))	(((b1*d)*((a3*b3)*c3))*((a2*b2)*c2))*(c1*((a0*b0)*c0))	(c0*(((a0*b0)*b1)*c1))*(((a2*((b2*d)*(a3*c2)))*b3)*c3)
24	((d*c3)*(b1*c0))*(((b1*((a3*b2)*c1))*c2)*((a0*b0)*c0))	(((b1*d)*c1*((a3*b3)*c3)))*((a0*b0)*c0))*((a2*b2)*c2)	(c0*(a2*c1))*(((((a0*b0)*b1)*((b2*d)*(a3*c2)))*b3)*c3)
25	((a3*b2)*(b3*c2))*(((b1*c0)*d)*(c0*(((a0*b0)*b1)*c1)))	(((b1*d)*(c1*((a3*b3)*c3)))*((a2*b2)*c2))*((a0*b0)*c0)	(c0*((a0*b3)*c2))*(((a2*b1)*((b0*d)*((a3*b2)*c1)))*c3)
26	((a3*b2)*(b3*c2))*(((b1*c0)*(((a0*b0)*b1)*c1))*(c0*d))	(((b1*d)*((a0*b0)*c0))*((a2*b2)*c2))*(c1*((a3*b3)*c3))	(c0*(((a2*b1)*((b0*d)*((a0*b3)*c2)))*c3))*((a3*b2)*c1)
27	(((a0*b0)*(b1*c0))*(b1*d))*((b3*c2)*(c0*((a3*b2)*c1)))	(b1*d)*(((c1*((a3*b3)*c3))*((a0*b0)*c0))*((a2*b2)*c2))	(c0*((a3*b2)*c1))*(((a2*b1)*((b0*d)*((a0*b3)*c2)))*c3)
28	(((a0*b0)*(b1*c0))*(b3*c2)*(c0*d)))*(b1*((a3*b2)*c1))	(((b1*d)*((a2*b2)*c2))*((a0*b0)*c0))*(c1*((a3*b3)*c3))	(c0*((((a2*b1)*b0)*((b3*d)*(a0*c3)))*c2))*((a3*b2)*c1)
29	(((a0*b0)*(b1*c0))*(b1*((a3*b2)*c1)))*(b3*c2)*(c0*d))	(b1*d)*(((c1*((a3*b3)*c3))*((a2*b2)*c2))*((a0*b0)*c0))	(c0*(((a3*b2)*(b0*(a0*c3)))*c2))*(((a2*b1)*(b3*d))*c1)
30	(((a0*b0)*b1)*(((b3*c2)*(c0*d))*(a3*b2)*c1)))*(b1*c0)	(((b1*d)*((a0*b0)*c0))*((a3*b3)*c3))*(c1*((a2*b2)*c2))	(c0*((((a2*b1)*(b3*d))*(b0*((a3*b2)*c1)))*c2))*(a0*c3)

上圖是電腦跑出來之後的結果

終點	a0	b0	c0
方法數目	2918	1902	64



a0 是終點方法數量最多

→因為消除 **a** 的公式最少

→以 **a** 為終點最容易破關

→在其他棋盤中只要 **a** 不能成為終點，則該圖形不能成為棋盤（推測）

伍、研究結果

一、我們發現翻轉元素和整數很相似。以乘法而言，F0 就相當於整數的 1。而翻轉元素也有乘法中的倒數。比較值得一提的是 F2、V、H、O1、O2 都是自己的倒數，這在整數中比較少見。

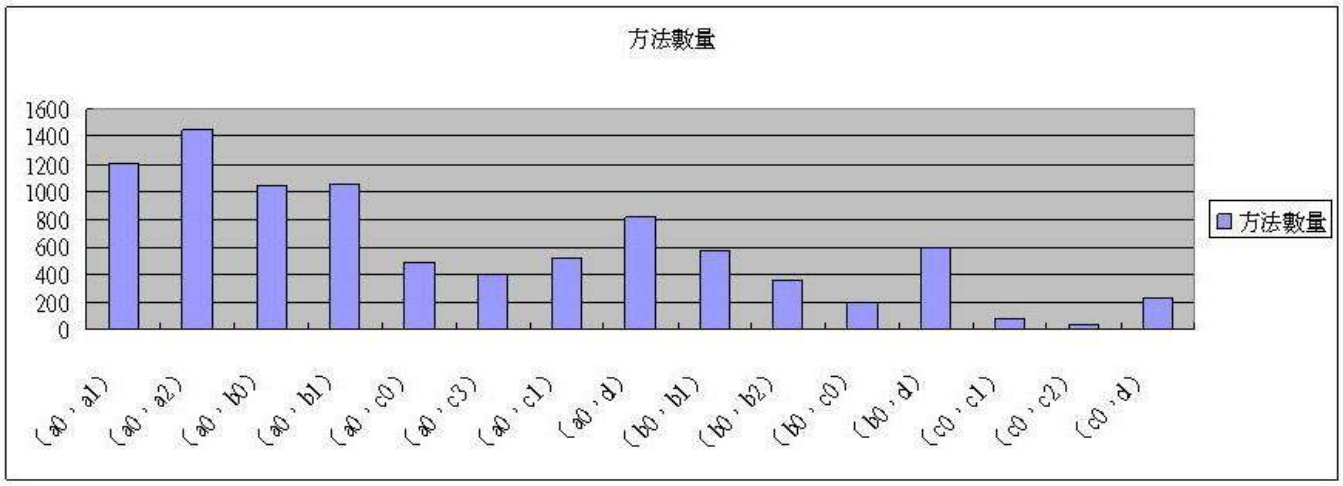
二、哪些圖形可以成為關卡

（一）根據窮舉法的研究，我們發現 2 顆時有 5 種圖形可以當成關卡。

（二）根據電腦運算的結果，我們發現在 12 顆時空位不管在那個點，都可以破關。

（三）我們將 a0 的結果使用電腦篩選，想知道 2 顆空位時有哪些是不能成為關卡的，結果我們發現在有 2 顆空位時，所有的圖形都可以成為關卡，終點都可以走到 a0。（如下表和下圖）

圖形	(a0，a1)	(a0，a2)	(a0，b0)	(a0，b1)	(a0，c0)	(a0，c3)	(a0，c1)	(a0，d)	(b0，b1)	(b0，b2)	(b0，c0)	(b0，d)	(c0，c1)	(c0，c2)	(c0，d)
方法數量	1208	1446	1050	1055	492	408	521	813	572	357	192	597	73	30	230



三、棋譜上有 12 棵棋子時要怎麼破關

（一）樹枝法：我們發現可以用 3333->12 的方式用 4 種不同的 3 科排列去組合成 12 顆。如此按照順序可以破關。不過目前也只有發現 2 種合併方法。另外 66->12 也是一種方法。444->12 也很快就可以成功。

（二）電腦運算法：經過電腦運算，我們發現只要終點目標是 a0，不管空為在哪裡都可以破關。我們對電腦運算法的整理如下：

以(c0*((a2*b2)*c2))*(((a0*b0)*((b1*d)*((a3*b3)*c3)))*c1)＝0 這個公式為例。

我們可以看到有許多（ ），乍看之下很難分辨出怎麼跳。但只要把握一個原則：「由左而右，後取代的先跳，裡面的先解決」就可以很容易破關了。作法如下

(c0*((a2*b2)*c2))*(((a0*b0)*((b1*d)*((a3*b3)*c3)))*c1)
(c0*((a2*b2)*c2))*((a1*((b1*d)*((a3*b3)*c3)))*c1)
(c0*((a2*b2)*c2))*((a1*((b1*d)*(a0*c3)))*c1)
(c0*((a2*b2)*c2))*((a1*(b3*(a0*c3)))*c1)
(c0*((a2*b2)*c2))*((a1*(b3*d))*c1)
(c0*((a2*b2)*c2))*((a1*b1)*c1)
(c0*(a3*c2))*((a1*b1)*c1)
(c0*(a3*c2))*(a2*c1)
(c0*d)*(a2*c1)
C2*(a2*c1)
C2*d
C0

左圖中我們將同組的（）圖上相同顏色，這樣就可以方便找出哪一顆棋子要先跳，和接下來跳的方向。

陸、討論

一、翻轉元素的探討：我們利用 excel 錄製巨集的功能，將翻轉元素放入 excel 中，這樣我們就可以很快的知道各顆組合，每一個圖形有多少種數量。

（一）翻轉元素巨集：

1. 思考邏輯：為了不讓電腦把 1 和 11 弄混，所以我們必須先將數字對應到英文字母然後按照 a~m 及 1~13 的順序將英文字母改回數字。

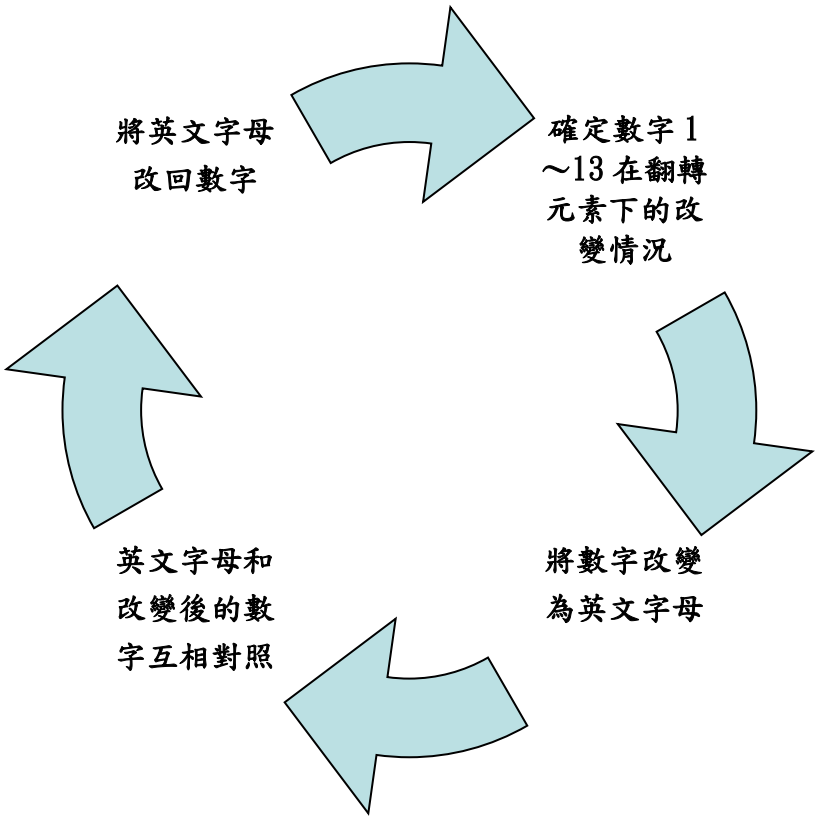


圖 6-1-1：翻轉元素巨集邏輯

2. 巨集：根據錄製後的巨集，我們更改巨集以合適用，結果如下

圖 6-1-2：翻轉巨集

（二）產生結果：有了翻轉巨集後，我們就可以很快速的找到各圖形在各顆下所產生的數量。

表 6-1-1：2 顆時各種圖形數量表（紅色代表可成為棋譜）

圖形	（1，2）	（1，3）	（1，5）	（1，6）
數量	4	2	8	8
圖形	（1，9）	（2，9）	（4，9）	（5，6）
數量	8	4	4	4
圖形	（5，7）	（5，9）	（7，9）	（9，10）
數量	2	8	8	4
圖形	（9，11）	（5，13）	（9，13）	（3，11）
數量	2	4	4	4

表 6-1-2：3 顆時各種圖形數量表

圖形	（1，2，3）	（1，2，4）	（1，2，5）	（1，2，6）
數量	2	1	4	4
圖形	（1，2，7）	（1，2，8）	（1，2，9）	（1，3，2）
數量	4	4	4	1
圖形	（1，3，5）	（1，3，6）	（1，3，9）	（1，4，9）
數量	4	4	4	4
圖形	（1，5，6）	（1，5，7）	（1，5，8）	（1，5，9）
數量	8	4	4	8
圖形	（1，6，7）	（1，6，8）	（1，6，9）	（1，7，9）
數量	4	4	8	8

圖形	（1，8，9）	（2，1，9）	（2，4，9）	（2，5，9）
數量	8	4	2	8
圖形	（2，7，9）	（4，1，9）	（4，2，9）	（4，5，9）
數量	8	4	2	8
圖形	（4，7，9）	（5，6，7）	（5，6，8）	（5，6，9）
數量	8	3	1	4
圖形	（5，7，9）	（5，8，9）	（6，7，9）	（7，8，9）
數量	8	4	4	4
圖形	（1，9，10）	（1，9，11）	（1，9，12）	（1， 2，13）
數量	8	4	8	4
圖形	（1，3，13）	（1，5，13）	（1，6，13）	（1，9，13）
數量	2	8	8	8
圖形	（2，1，19）	（2，9，11）	（2，9，13）	（4， 9，13）
數量	2	2	4	4
圖形	（5，9，10）	（5，9，11）	（5，9，12）	（5， 6，13）
數量	4	4	4	4
圖形	（5，7，13）	（5，9，10）	（5，9，11）	（5， 9，13）
數量	2	4	4	8
圖形	（7，12，9）	（7，9，13）	（9，10，11）	
數量	4	8	4	
圖形	（9，10，13）	（9，11，13）		
數量	4	2		
圖形	（9，10，11）	（9， 10，12）		
數量	1	3		

（9，10，11）和（9， 10，12）應該要改成（9，10，11）共 4 個

（三）發現：經過上面的研究，我們發現「所有圖形的數量都是 8 的因數。」所以我們思考是否用 F0～F3 做翻轉元素，做出來的圖形的數量會是 4 的因數，結果如下：

表 6-1-3：2 顆時各種圖形數量表只有 F0～F3（錯誤）

圖形	（1，2）	（1，3）	（1，4）	（1，5）
數量	3	2	1	4
圖形	（1，6）	（1，7）	（1，8）	（1，9）
數量	4	4	4	4
圖形	（2，9）	（3，9）	（4，9）	（5， 6）
數量	4	4	4	3

圖形	（5，7）	（5，8）	（5， 9）	（6，9）
數量	2	1	4	4
圖形	（7，9）	（8，9）	（9，10）	（3， 11）
數量	4	4	1	4
圖形	（9，11）	（9，12）	（5， 13）	（9，13）
數量	2	3	4	4

從表 6-1-3 我們發現有些圖形的數量不是 4 的因數，但那些圖形是可以合併到別的圖形計算的。訂正結果如下：

表 6-1-4：2 顆時各種圖形數量表只有 F0～F3（訂正）

圖形	（1，2）	（1，3）	（1，5）	（1，6）
數量	4	2	4	4
圖形	（1，7）	（1，8）	（1，9）	（2，9）
數量	4	4	4	4
圖形	（3，9）	（4，9）	（5， 6）	（5，7）
數量	4	4	4	2
圖形	（5， 9）	（6，9）	（7，9）	（8，9）
數量	4	4	4	4
圖形	（3， 11）	（9，11）	（9，12）	（5， 13）
數量	4	2	4	4
圖形	（9，13）			
數量	4			

根據上面的結果，我們推測：「各種圖形的數量會是翻轉元素數目的因數」

（四）其他圖形的翻轉元素：我們思考上面的推測在改成其他圖形和翻轉元素的時候，不知是否會成立

1. 三角形

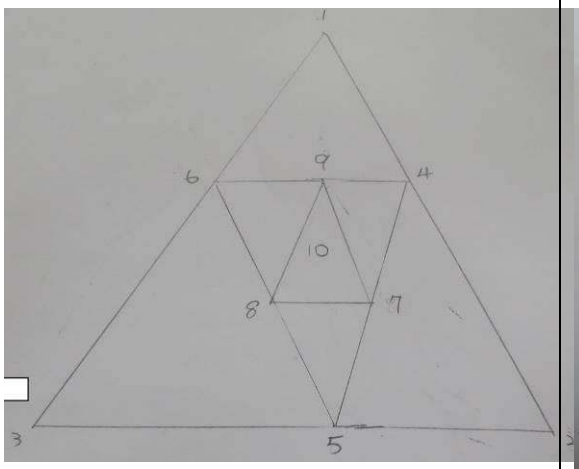
三角形棋盤	翻轉元素
	

表 6-1-5：我們試著用三角形來做數字組合，翻轉元素作用如圖中。結果如下

表 6-1-6：2 顆時在三角形中各種圖形數量表

圖形	（1，2）	（1，3）	（1，5）	（1，7）
數量	3	6	6	6
圖形	（1，8）	（1，9）	（5， 6）	（5，9）
數量	6	6	3	6
圖形	（3，11）	（9，12）	（5，13）	（9，13）
數量	6	3	6	6

從表 6-1-6 我們可以看到各種圖形的數量是 6 的因數

2. 五邊形

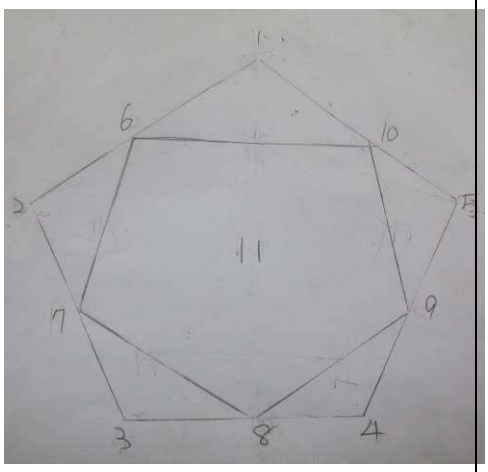
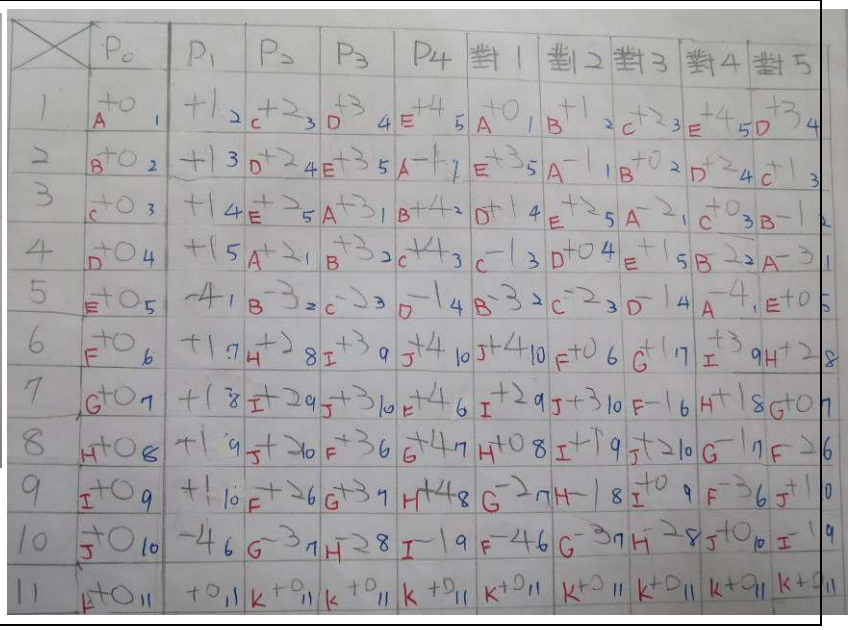
三角形棋盤	翻轉元素
	

表 6-1-7：我們試著用五邊形來做數字組合，翻轉元素作用如圖中。結果如下

表 6-1-8：2 顆時在五邊形中各種圖形數量表

圖形	（1，2）	（1，3）	（1，6）	（1，7）
數量	5	5	10	10
圖形	（1，8）	（6，7）	（6，8）	（1，11）
數量	5	5	5	5
圖形	（6，11）			
數量	5			

從表 6-1-8 我們可以看到各種圖形的數量是 10 的因數

根據上面的研究，我們推測：「在 N 邊形中，經過翻轉元素之後的圖形數量會是 2N 的因素。」

二、電腦取代法三：我們認為電腦取代法二需要用到 32 個活頁，重複的動作太多，應該可以縮減，將 32 個活頁的動作都放置於一頁・

(一)上網尋求支援：為了能夠分項刪除或保留，我們上網發問，希望可以得到解答・結果如下：

圖 6-2-1 上網詢問分項刪除的巨集

(二)電腦取代法三的邏輯：可以分項刪除和保留指定欄位後，我們就可以製作電腦取代法三了，邏輯如下・

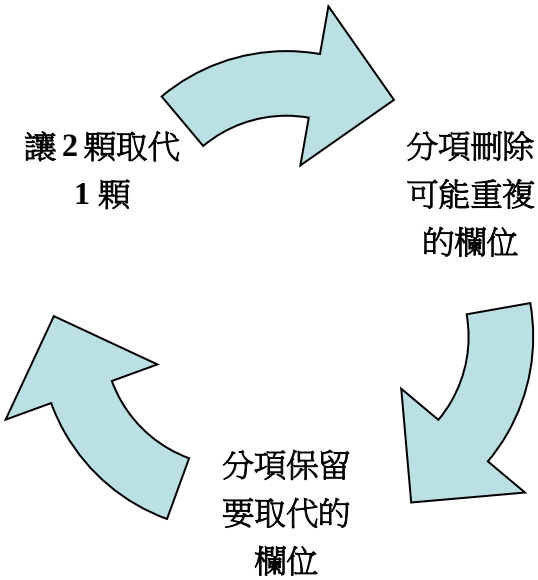


圖 6-2-2：電腦取代法 3 的邏輯

(三)括號解決方案：在電腦取代法二中，我們發現括號都一樣會不容易解讀公式・原本我們想要用顏色來區分次數，但後來發現無法如此，所以我們用不同的符號取代括號，方便解讀公式，不同括號定義如下：

顆數	1	2	3	4	5	6	7	8	9	10	11	12
括號	（）	[]	{}	~ ~	! !	@@	# #	$ $	&&	《》	、、	【】

表 6-2-1：不同括號分類表

因此我們可以得到容易解答的公式，以 8 顆公式為例：(後取代先走)

([{#a1*b0#*~c1*!@a2*b1@*c0!~}*c2]*b3)

第一步：#a1*b0#

第二步：@a2*b1@

第三步：!@a2*b1@*c0!

第四步：~c1*!@a2*b1@*c0!~

第五步：{#a1*b0#*~c1*!@a2*b1@*c0!~}

第六步：[{#a1*b0#*~c1*!@a2*b1@*c0!~}*c2]

第七步：([{#a1*b0#*~c1*!@a2*b1@*c0!~}*c2]*b3)

三、擴大與變形

（一）擴大結果：我們嘗試了擴大的棋盤，棋盤如下：

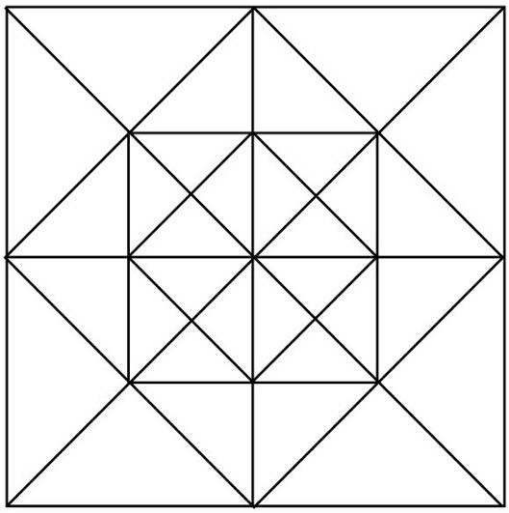


圖 6-3-1：5 階正方形棋盤

我們發現 3 階正方形棋盤無法留下只剩一顆(破關)，但 4 階、5 階正方形棋盤都可以破關。

（二）形狀變形：我們思考除了正方形之外，是不是還有其他形狀的棋盤，所以做了嘗試如下：

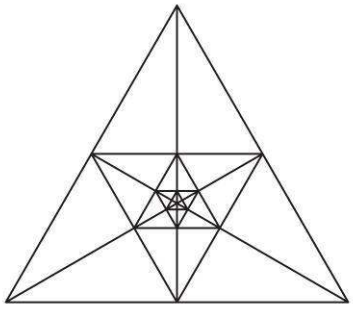
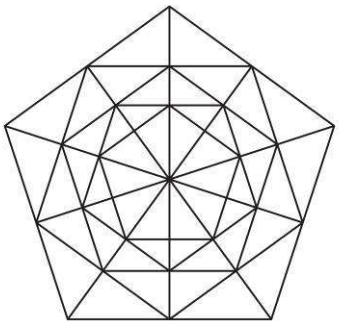
5 階三角形棋盤	5 階五邊形棋盤
	

表 6-3-1：各種形狀棋盤

我們發現 3 階三角形棋盤可以破關， 4 階 5 邊形棋盤也可以找到破關方法，所以『當 N 為奇數時，4 階的 N 邊形棋盤都可以有破關方法』。

（三）推論與較簡單的證明方法

1.我們發現 3 階三角形棋盤、4 階正方形棋盤、5 階五邊形棋盤可以破關。所以我們推論：「n 階的 N 邊形棋盤可以有破關方法」

2. 2 階三角形棋盤沒有破關方法證明：我們將棋盤上的點分為 a0~a2 和 b 兩種目前且 a0~a2 和 b 皆無法跳動，所以不能破關。

3. 3 階正方形棋盤沒有破關方法證明：我們將已知條件表列如下

破關條件	a1+a2+a3+b0+b1+b2+b3+c=a0 或 b0 或 c a0+a1+a2+a3+b1+b2+b3+c= a0 或 b0 或 c a0+a1+a2+a3+b0+b1+b2+b3=a0 或 b0 或 c
已知條件	a1+b0=a0　a3+b3=a0　b0+c=b2 a2+b1=a1　a0+b0=a1　b1+c=b3 a3+b2=a2　a1+b1=a2　b2+c=b0 a0+b3=a3　a2+b2=a3　b3+c=b1

表 6-3-2：已知條件表

如果能將已知條件代入破關條件，並得到破關答案就可破關・可惜根據窮舉法的結果我們發現 3 階正方形是沒有破關方法的

（四）重疊：我們還思考是否有其他有趣的棋盤・我們想到一種可能的棋盤

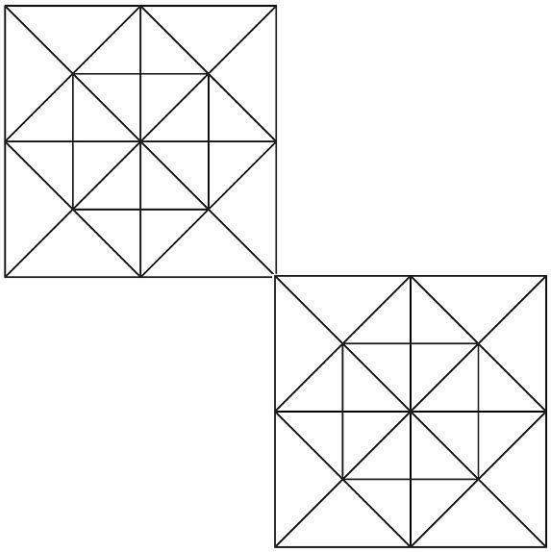
棋盤	破關方式
	1.空位：a0 2.中點：左上正方形剩 1 顆在 d 右下正方形全滿 3.右下正方形 c3 跳過 a0，之後左上 d 跳過 c1 4.右下一個空位 c3

表 6-3-3：兩個正方形棋盤重疊成一個新的棋盤

四、青蛙環：我們發現破關方法之間有特殊規律

(一)空位和實位：我們發現任一破關方法的空位會有一破關方法的實位對應・例如破關方法 A 的第一步空位可對應破關方法 B 的第 12 步實位，如下圖。

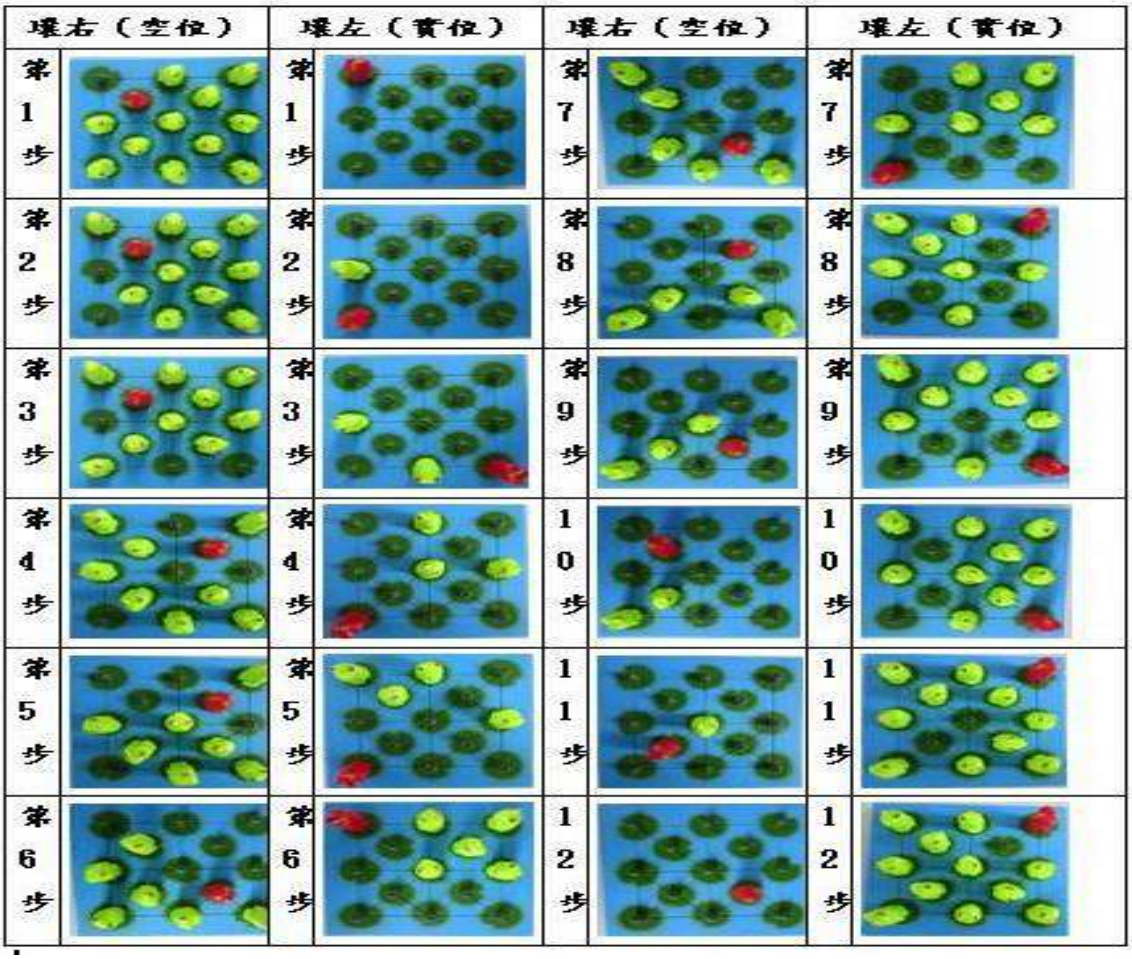


圖 6-4-1：空位和實位對應圖

(二)青蛙環的結構：根據上面的發現，我們認為破關方法的實位和空位是可以對應且有規律的，如下圖。

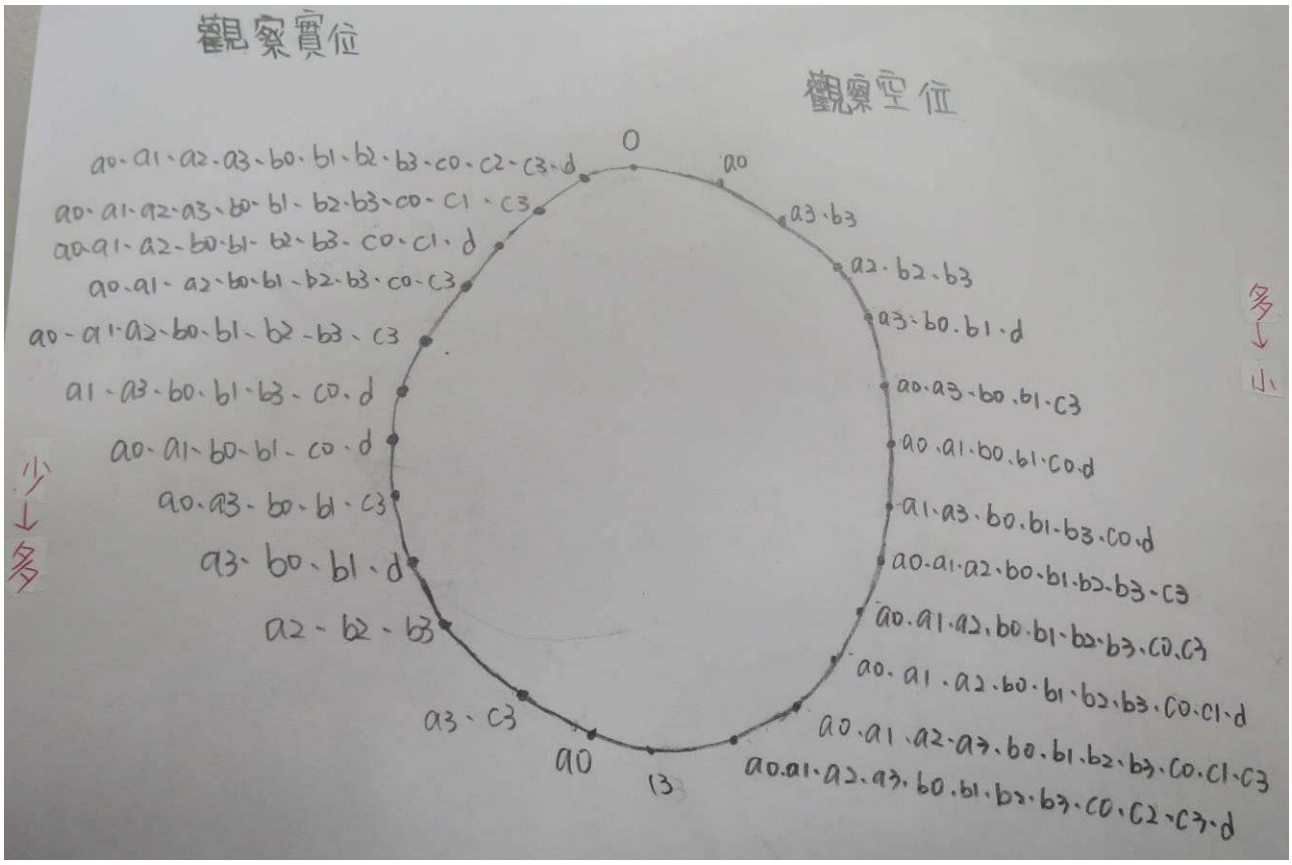


圖 6-4-2：青蛙環

從圖 6-4-2 我們可以看到青蛙環是由 2 個破關方式構成。右邊的破關方式是青蛙棋由多變少，左邊破關方式用取代的方式由少變多。右邊的空位=左邊的實位，對角線的空位會等於對角線的實位。

柒、結論

一、翻轉元素相當有趣，我們打算上網搜尋整數的特性和翻轉元素一一比對，看整數和翻轉元素除了目前了解的之外，還有甚麼相同的地方。

二、樹枝法和區塊法雖然有趣，但只找到 2 個 12 顆的解答，希望往後可以找到更多的解答。

三、我們用電腦運算將全部的 2~12 顆破關方法算出來了。但結果還需另外解讀，應該有更快更迅速的方法。另外雖然用電腦運算快了很多，但也花了我們整個寒假的時間，往後我們會多學習 Excel 希望可以改進目前的做法。

四、我們證明了 2 階三角形棋盤、3 階正方形棋盤是沒有破關方法的，但我們認為應該有更簡單的方式可以證明「n 階 N 邊形有破關方法。」

五、青蛙環在 4 階正方形棋盤中可以見到，但希望可以找到環左和環右完全相同的青蛙環。

六、在經過 n 邊形翻轉元素產生的圖形數量會是 2n 的因數。

捌、參考資料

一、臺北市第 42 屆中小學科學展覽會國小組數學--搬新家-一人跳棋之毛蟲棋探討

二、中華民國第 51 屆中小學科學展覽會國小組數學科--三十六計，『跳』為上

三、http://blog.xuite.net/smile1000mile/blog/95241202-EXCEL-- EXCEL VBA．如何快速刪除符合條件的列？

四、http://www.techbang.com/posts/10605-- Excel 自動刪除表格重複資料：篩選重複資料，二步驟快速刪除

【評語】080401

本件作品主要探討青蛙棋譜及相關問題。主題相當具有趣味。問題的討論分析也具有相當的深度。並且利用電腦程式進一步研究更複雜的情況，這是一件相當不錯的作品。