中華民國第 57 屆中小學科學展覽會

作品說明書

高級中等學校組　數學科

最佳團隊合作獎

050407

The Magic Power of Sequencing Polyominoes

學校名稱：國立宜蘭高級中學

作者：	指導老師：
高三 陳佑昕 高三 陳韋綸 高三 黃奕鈞	沈碧照

關鍵詞：polyomino、SAW、sequence

摘要

本研究係用「數列命名的方法」作為骨幹，解決或簡化多方塊(polyominoes,也稱作 連方圖形)的相關問題。「數列命名的方法」 係指將多方塊的邊長連續寫成數列。為解決使用「數列命名的方法」遇到的問題，目前建立了一套系統、技巧，可將 Polygon，旋轉、鏡射、合併等等，在此專門用以處理多方塊相關問題。 研究發現此方法可以解一些多方塊的問題，例如多方塊的種類、任意的多個相同的多方塊是否可以填滿(嵌滿)矩形(平面)。對於計算多方塊數量問題，可分為兩部分探討，一是多方塊的形的個數，二是單一多方塊的形其中含有多少多方塊，本研究主要探討此二部分的特性，並簡化運算結構和找出個數範圍。

壹、 研究動機

計算多方塊的種類是複雜的，在相對應的數論問題上，解決多方塊的計數問題，相當於完成質數分布或質數計算函數的更精確技巧(並且延伸)，因而在此問題上，目前僅能尋求相對緊密的函數上界和下界，並藉由處理上下界問題，試圖找出多方塊種類隨著連方數增加而變化的趨勢。

同時，多方塊在歐幾里得平面上的排列性質亦是困難的，然而利用 SAW(Self Avoiding Walking)的思維可以用代數模型描述多方塊在平面上的特性，透過模型定義，也容易嚴格證明一些直觀容易理解卻不易解答的問題，獲得這些知識後，進而引起研究與延伸探討的動機。

貳、 研究目的

一、找出一個足以描述多方塊(one-sided polyomino)的數學模型
二、簡化該模型的規則
三、透過建立數學模型以理解多方塊在歐幾里得平面上的性質
四、找出多方塊種類函數的範圍
五、縮小多方塊種類函數的上界與下界距離
六、整理相同多方塊在平面上組成矩形的條件

參、 研究內容與過程

一、將多方塊轉化為一數列

(一). 假設有一個圖形 Fig.1，可以任意選定起點，命往右或往上為正，往左或往下為負，先左右後上下，連續取數列。範例中令 A 為起點，利用數列(1,1,1,2,-2,-3)作為該圖形的命名。
(陳佑昕(1)陳韋綸(2)李昕(3)[17])

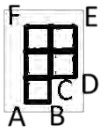


Fig.1

(二). 判斷一個數列是否為連方：

假設存在一數列(a_1 ,a_2 ,a_3 ,…,a_{n-1} ,a_n)，該數列代表一個多方塊的必要條件包括：

1. $\sum_{i=1}^{\frac{n}{2}} a_{2i-1} = 0 \wedge \sum_{i=1}^{\frac{n}{2}} a_{2i} = 0 \ , a_i \in Z$

2. 任意取數列中連續的數，假設是數列中的第 g 項到第 g+2x+3 項，x 為適當自然數：

$$\text{When } (a_{g+3}+a_{g+5}+a_{g+7}\ldots+a_{g+2x+3})\times(a_{g+1})<0\wedge$$
$$(a_{g+2}+a_{g+4}+a_{g+6}\ldots+a_{g+2x+2})\times(a_g)<0$$
$$\text{if } |a_{g+3}+a_{g+5}+a_{g+7}\ldots+a_{g+2x+3}|\geq|a_{g+1}|$$
$$then|a_{g+2}+a_{g+4}+a_{g+6}\ldots+a_{g+2x+2}|>|a_g|$$

(三). 多方塊的數列運算

1. 判斷兩個數列是否代表同一圖形：

在平面座標系上，一個圖形可以被旋轉、鏡射，而在判斷多方塊數量時，就必須判斷多個圖形是否代表的是同一種圖形。透過命名後的數列判斷的方式：

(1) 數列中的每一個數取絕對值，將該數列排成一個環，判斷比較環是否相等
如上圖，若選定 A 為起點(1,1,1,2,-2,-3)則環如 Fig.2
若選定 F 為起點(2,-2,-1,-1,-1,3)則環亦相等
此兩環相等，表同一圖形(判斷方式一)。

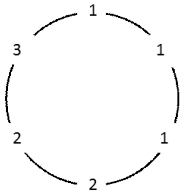


Fig.2

2. 判斷一個數列所代表的圖形為幾連方：

判斷一個圖形為 t 連方，相當於在平面座標系上圖形所圍成的面積為 t。
計算方法同樣可隨意取圖形一頂點作為原點，並分別以奇數項、偶數項代表平面座標系上的 x 軸、y 軸：

可由下式 $\sum_{i=1}^{\frac{n}{2}}[a_{2i}\times(\sum_{k=1}^{i}a_{2k-1})]=$ t計算數列代表的連方數 t。

3. 圖形的旋轉與鏡射

(1) 若兩數列的奇數項彼此互為相反數，表示兩數列所代表的圖形為對 y 軸做鏡射的圖形；若兩數列的偶數項彼此互為相反數，表示兩數列所代表的圖形為對 x 軸做鏡射的圖形。

(2) 若兩數列分別為 $A(a_1,a_2,a_3,\ldots,a_{n-1},a_n)$與 $B(a_2,a_3,\ldots,a_{n-1},a_n,a_1)$，則圖形 B 為圖形 A 順時針旋轉 90 度且 x 軸鏡射(定義原數列奇數項為 x 軸)後的圖形，或圖形 A 逆時針旋轉 90 度且 y 軸鏡射(原數列偶數項)後的圖形。

因此，若要將圖形順時針旋轉 90 度，可將數列中的奇數項均改成其相反數 $-a_{2i-1}\rightarrow a'_{2i-1}, i\in\mathbb{Z}$，再將數列(假設共 n 項)中的各個數往前放置$a_i\rightarrow a'_{i-1}; a_1\rightarrow a'_n$，$(2\leq i\leq n, i\in\mathbb{N})$。同理，若要將逆時針旋轉 90 度，可將數列中的偶數項均改成其相反數，其餘步驟相同。

4. 合併圖形

(1) 存在數列 $A(a_1,a_2,a_3,\ldots,a_{n-2},a_{n-1},a_n)$, $B(b_1,b_2,b_3,\ldots,b_{n-2},b_{n-1},b_n)$

(2) 假設兩數列 A 數列中有子數列 $A_t(a_m,a_{m+1},\ldots,a_{m+x-2},a_{m+x-1},a_{m+x})$與 B 數列中有子數列 $B_u(b_l,b_{l+1},\ldots,b_{l+x-2},b_{l+x-1},b_{l+x})$為一一對應的數，即 $(b_l,b_{l+1},\ldots,b_{l+x-2},b_{l+x-1},b_{l+x})=\pm(a_{m+x},a_{m+x-1},\ldots,a_{m+2},a_{m+1},a_m)$ 則代表兩圖形中有構形一樣(可以合併)的部分，僅因合併時相對位置之故，如 Fig.3 兩圖形相交之片段。

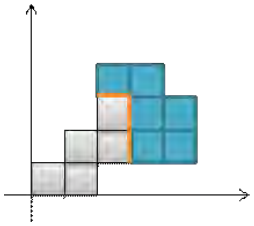


Fig.3

(3) 將兩數列分別拆為兩不完整的子數列(相接部分及其他)，圖形相加之後的數列，去掉兩數列相同的部分，新數列可表示為 $C(a_1,a_2,\ldots,a_{m-1}+b_{2l+x},b_{l+x+2},b_{l+x+3},\ldots,b_{l+x+1}+a_{m+x+1},a_{m+x+2},a_{m+x+3},\ldots,a_{n-2},a_{n-1},a_n)$。

(4) 反圖形:

反圖形可視為在座標平面上位於相反象限(一與三象限、二與四象限)的圖形(原點相同，考慮其前兩項正負)，或言反圖形互為旋轉 180 度後的圖形，如 Fig.4。加 A 圖形的反圖形，相當於減 A 圖形。

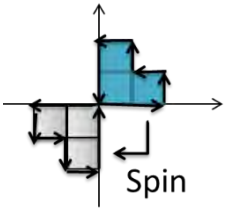


Fig.4

(5) 順(逆)時針:

從起點開始隨數列值行進有其順序，如 Fig.5 可以箭頭表示(順(反)時針):

在合併圖形時應將兩圖形轉為順逆時針相同的圖形再行相加。(亦可以順逆相反的圖形相減)

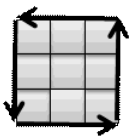


Fig.5

(6) 由於一個圖形的數列實際上可以無限循環，即 $A(a_1,a_2,a_3,\ldots,a_{n-2},a_{n-1},a_n)\equiv A(a_4,a_5\ldots,a_{n-2},a_{n-1},a_n,a_1,a_2,a_3)$，因此在進行圖形的合併時，定義上一個數列可以有無限多項。而要簡化計算，可將數列 $B(b_1,b_2,b_3,\ldots,b_{n-2},b_{n-1},b_n)$改成 $B'(b_{l+x+1},b_{l+x+2},b_{l+x+3},\ldots b_{n-1},b_n,b_1,b_2,\ldots,b_{l+x-2},b_{l+x-1},b_{l+x})$。修改後，B'的起點就在合併圖形的重疊部分的端點上，Fig.6 上的點即為圖片右側圖形的起點。

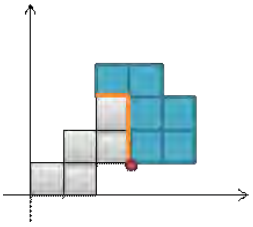


Fig.6

(7) 合併時，將修改過的數列相加，併消去相加後其值為 0 的項，例：(3,1,-1,1,-1,-1,-1,-1)和(2,-2,-1,1,-1,1)的其中一種合併(粗體為合併邊(也就是 At 和 Bu))，將兩個做欲合併處的對應

→ (**3, 1,-1, 1,-1**,-1,-1,-1)

(2,**-2,-1, 1,-1,1**)

上下任取一數列將底線部分左右順序互換

(底線部分為合併時重疊到的邊，不一定完全等於粗體的構形相同處)。

最後再將左右互換的數列的剩餘項放入兩□之間。

(□是因為合併而長度有所變化的邊) 。

(3, **1,-1, 1**,-1,-1,-1,-1)
+) (2,1,**-1, 1,-1**,-2)

=(4,2,-3,-1,-1,-1)

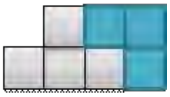


Fig.7

(8) 若欲將兩圖形不完美的合併(如 Fig.8)，則合併的計算需使用廣義的合併處理，因此消去的數列長度(x+1)可奇亦可偶，且消去的數列其他片段(除 b_l,b_{l+x},a_m,a_{m+x})限制仍相同(為一一對應的相反數)，唯偶數時$|b_l| \neq |a_m| \vee |b_{l+x}| \neq |a_{m+x}|$。

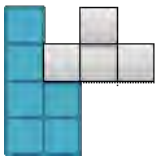


Fig.8

計算過程以 Fig.8 為例

(1 / 1 /0, 0/-1 / 0, -1 , -4 , 2 , 2)
+)(1 , -1 , 1 , -1 , -1 , -1 , 0 / -1/0 ,0/ 1 / 2)

(2 , 1 , -1 , 1 , -1 , -1 , -1 , 1 , -1 , -4 , 2 , 2)

(9) 不完美的合併規則和完美的合併規則本質上相同，

都是將重疊項相加，最後去掉其值為 0 的項，只是□應為重疊項的頭或尾項。例如(上圖)：(**3**,1,-1,1,-1,-1,-1,**-1**)和(2,2,**-1,2**,-1,-4)的其中一種合併，將(**-1,3**,1,-1,1,-1,-1,-1)和(-1,-4,2,2,**-1,2**)做欲合併處的對應

→ (**-1, 3**, 1,-1,1,-1,-1,-1)

(-1,-4,2,2,**-1, 2**)

上下任取一數列將底線部分左右順序互換。

最後再將左右互換的數列的剩餘項放入兩□之間。

(**-1, 3**, 1,-1,1,-1,-1,-1)
+) (-1,-4, 2, 2, 2,-1)

= (**1**,-1,-4, 2, 2, **2**, 1,-1, 1,-1,-1,-1)

(10) 若只接觸一邊(廣義合併原理):

如(**1**, 1,-1,-1)與(3,1,**-3**,-1)的其中一種合併

→ (**1**, 1,-1,-1)

(-1, 3, 1,**-3**)

接著，將兩數列合併邊的數字依照原來數列的排序方向拆成三個整數，分別代表重疊部分的長度及兩側分別剩餘的長度。(拆成的三個數不可異號，換言之此三整數絕對值之和必須等於原整數之絕對值。)

(0 / 1/ 0, 1,-1,-1)

+) (-1, 3, 1, 0/-1/-2)

再來，按照前面提到的合併規則(p.5-(7))，將底線部分左右順序互換後再相加。

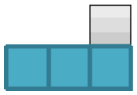


Fig.9

(0/ 1/ 0, 1,-1,-1)

+) (-1, 3,-2/-1/ 0, 1)

= (-2,-1, 3, 2,-1,-1)

二、以圖形命名後的數列討論與窮舉多方塊。

(一). 多方塊的數列規則與運算

1. 由於代表多方塊的數列必要的規則可以整合成一個不定方程，故可以嘗試用不定方程窮舉數列：

(1) 當數列長度=4:

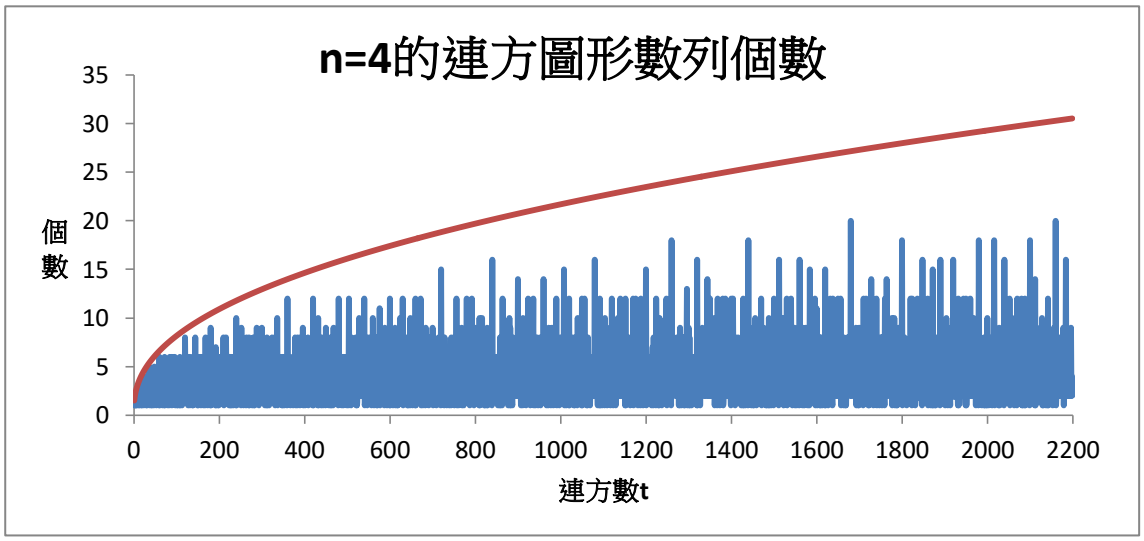
$$\because a_1 + a_3 = 0 \ \wedge \ a_2 + a_4 = 0 \wedge \ a_2(a_1) + a_4(a_1 + a_3) = t$$

$$\therefore a_1 \times a_2 = t$$

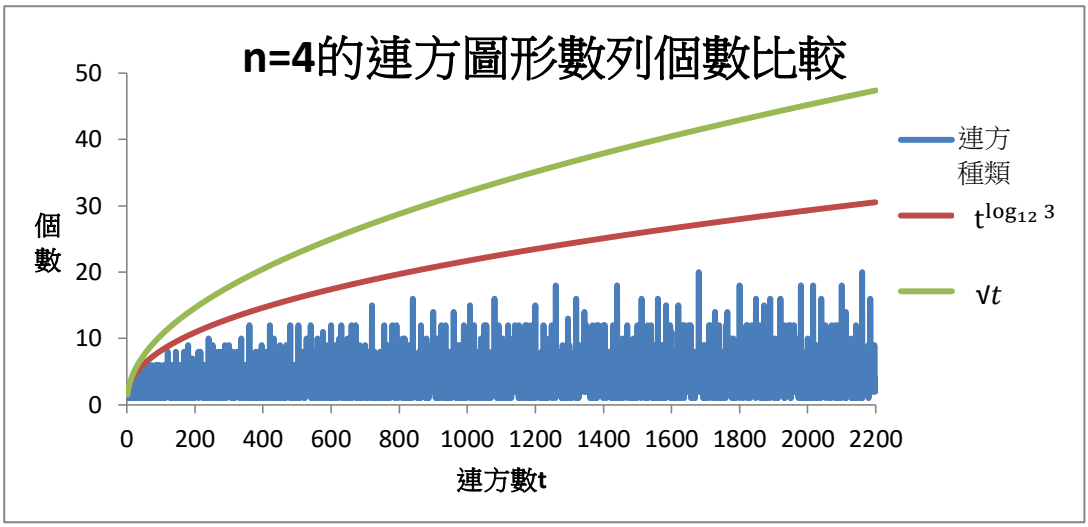
先填式子中的 a_1 和 a_2，再由奇數項合為零，且偶數項和為零算出 a_3 和 a_4。

t	1	2	3	4	4	5	6	6	7
a_1	1	1	1	1	2	1	1	2	1
a_2	1	2	3	4	2	5	6	3	7
a_3	-1	-1	-1	-1	-2	-1	-1	-2	-1
a_4	-1	-2	-3	-4	-2	-5	-6	-3	-7

以程式演算出各連方數長度為 4 的數列總數得到以下關係圖:



(2) 數列長度=4 的種類函數上界和正整數的正因數個數:



已知正整數 t 的正因數個數必小於$2\sqrt{t}$，故 n=4 的連方種類數必小於$\sqrt{t}+0.5$。基於這個事實，猜測 n=4 的連方種類函數上界亦為 t 的冪次。以電腦比對資料後發現可能為$t^{\log_{12} 3}$。

(3) 當數列長度=6:

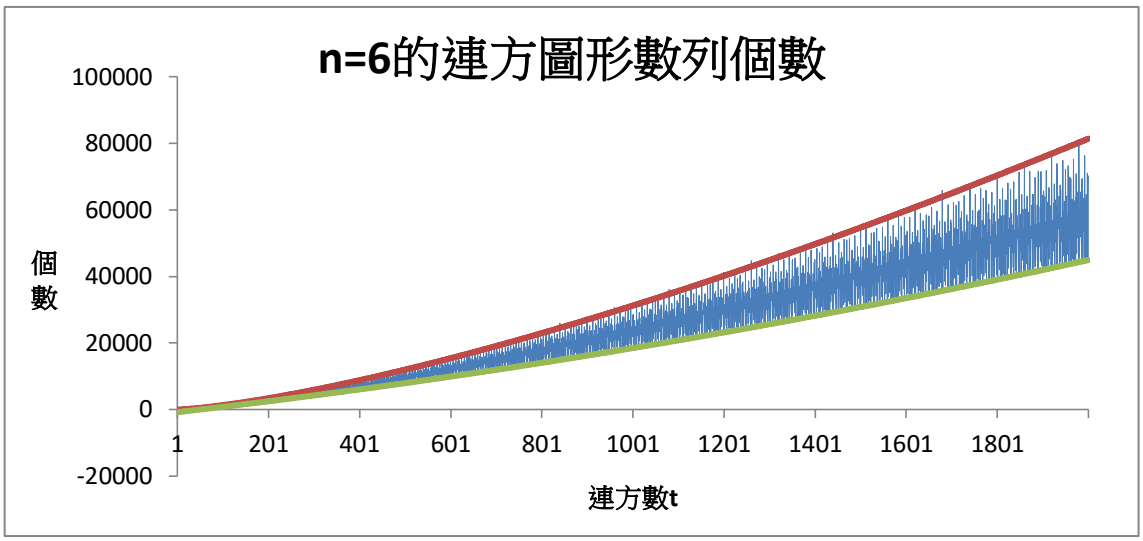
$$\because a_1+a_3+a_5=0 \wedge a_2+a_4+a_6=0$$
$$\wedge a_2(a_1)+a_4(a_1+a_3)+a_6(a_1+a_3+a_5)=t$$
$$\therefore a_1\times a_2-a_4\times a_5=t$$

先填式子中的 a_1、a_2、a_4、a_5再由奇數項合為零，且偶數項和為零算出 a_3 和 a_6。

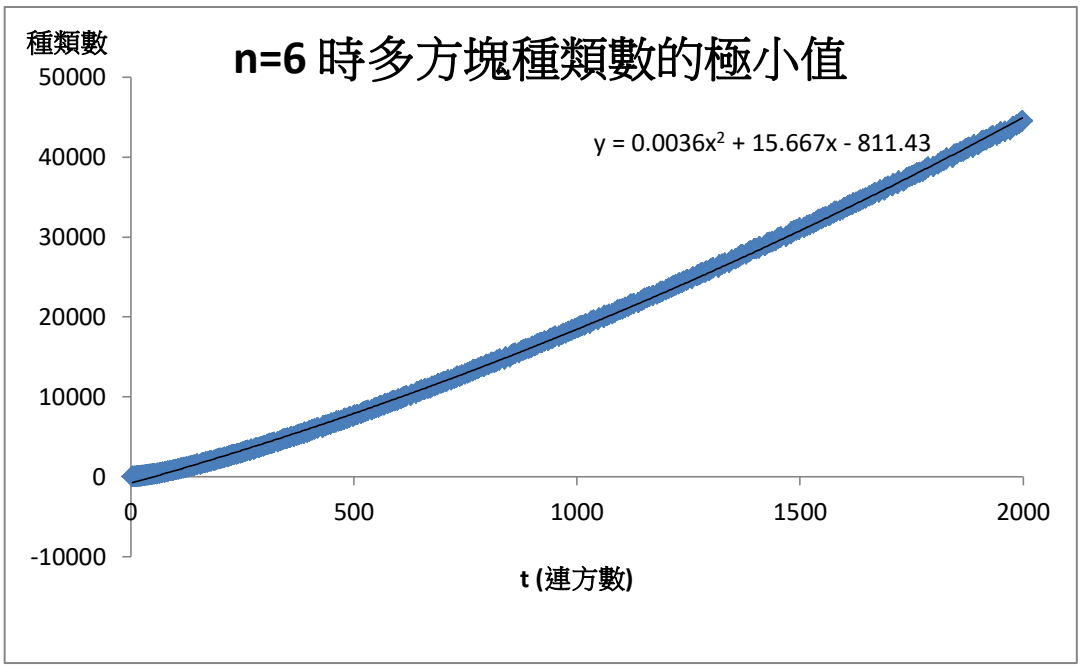
t	3	4	4	5	5	5	5	5	6	6	6	6
a_1	1	1	1	1	1	1	2	1	1	1	2	1
a_2	1	1	2	1	1	2	1	2	1	2	1	2
a_3	1	2	1	3	1	2	1	-4	4	3	2	1
a_4	1	1	1	1	2	1	1	-1	1	1	1	2
a_5	-2	-3	-2	-4	-2	-3	-3	3	-5	-4	-4	-2
a_6	-2	-2	-3	-2	-3	-3	-2	-1	-2	-3	-2	-4

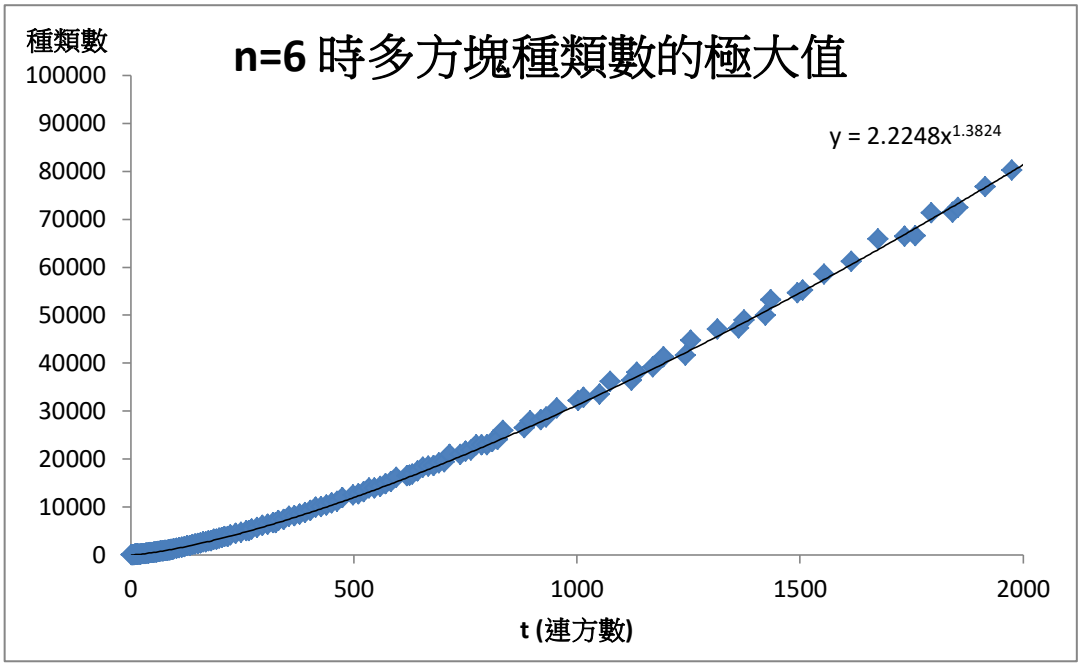
t	6	6	7	7	7	7	7	7	7	7	7
a_1	1	3	1	1	1	1	2	1	1	3	3
a_2	3	1	1	1	1	2	1	3	3	1	1
a_3	2	-2	5	2	1	4	-1	3	1	-2	-1
a_4	1	3	1	2	3	1	5	1	2	4	2
a_5	-3	-1	-6	-3	-2	-5	-1	-4	-2	-1	-2
a_6	-4	-4	-2	-3	-4	-3	-6	-4	-5	-5	-3

以程式演算出各連方數長度為 6 的數列總數得到以下關係圖:



(4) 數列長度=6 的種類函數上界和正整數的正因數個數:





利用程式搜尋 n=6 的連方種類函數的極值並找出趨勢線，發現極大值偏向冪次函數$\mathrm{P}(6,\mathrm{t}) \leq 2.2248t^{1.3824}$。

而極小值偏向多項式函數$\mathrm{P}(6,\mathrm{t}) \geq 0.0036t^2 + 15.667t - 811.43$。

三、關於多方塊形狀的討論。

(一).形：

1. 可以理解為不給定所有邊長的多方塊形狀。本文中將以「多方塊的形」或「形」指稱這些模糊的形狀。
2. 在數列中，形，即是正負號的排列，代表每條邊的方向。如 Fig.10 的數列為(1,1,2,2,-1,-1,-2,-2)。它代表的形可以表示為 (+ + + + - - - -)，習慣上可以寫成$\begin{bmatrix} + & + & - & - \\ + & + & - & - \end{bmatrix}$用以分開奇數項和偶數項。

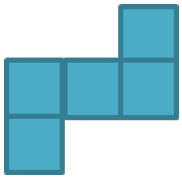


Fig.10

3. 觀察<以圖形命名後的數列討論與窮舉多方塊>(p.5)，可以發現計算多方塊種類數包含兩個部分：

(1) 數列長度 n 的多方塊有幾種形。

(2) 單就一種形狀而言，t 連方有多少種類。

前述<數列長度為 4>(p.6)和<數列長度為 6>(p.6)均只有一種形，因此引這兩者為例計算 t 連方種類數，而此節則討論數列長度為 n 的多方塊有幾種形。

(二).形的運算：

1. $\begin{bmatrix} + & - \\ + & - \end{bmatrix}$為最基本單位，代表一個封閉的圖形。
2. 從「路徑」的方面討論形的描述方式，整理如下表

單元	路徑	相當於
$\begin{bmatrix}+ & +\\ + & \end{bmatrix}$		$\begin{bmatrix}+\end{bmatrix}$
$\begin{bmatrix}+ & +\\ - & \end{bmatrix}$		$\begin{bmatrix}+\end{bmatrix}$
$\begin{bmatrix}+ & -\\ + & \end{bmatrix}$		
$\begin{bmatrix}- & +\\ + & \end{bmatrix}$		
$\begin{bmatrix}- & +\\ - & \end{bmatrix}$		
$\begin{bmatrix}+ & -\\ - & \end{bmatrix}$		
$\begin{bmatrix}- & -\\ + & \end{bmatrix}$		$\begin{bmatrix}-\end{bmatrix}$
$\begin{bmatrix}- & -\\ - & \end{bmatrix}$		$\begin{bmatrix}-\end{bmatrix}$

由上表可知，有四種單元可以簡化。

當同列相鄰的兩個符號同號，即可簡化為一個該符號。

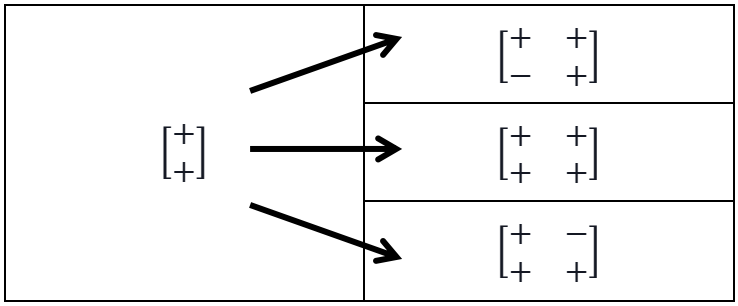
3. 透過這樣的簡化和觀察可知，並非所有的正負號排列都是多方塊的形，只有那些可以簡化為基本單位$\begin{bmatrix}+ & -\\ + & -\end{bmatrix}$的排列方式才能作為多方塊的形。

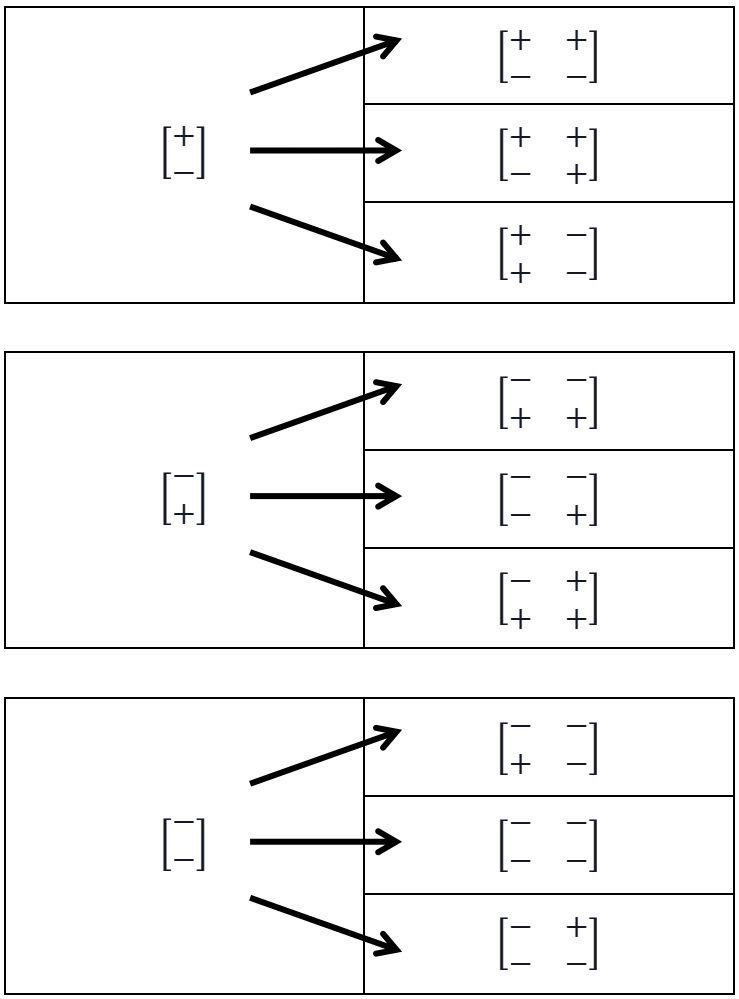
以$\begin{bmatrix}+ & + & + & -\\ + & - & + & -\end{bmatrix}$為例，簡化由箭頭表示，則：

$$\begin{bmatrix}+ & + & + & -\\ + & - & + & -\end{bmatrix} \rightarrow \begin{bmatrix}+ & + & -\\ + & + & -\end{bmatrix} \rightarrow \begin{bmatrix}+ & -\\ + & -\end{bmatrix}$$

(三). 形的種類

1. 若是由簡化繁，就能知道圖形最多有多少形(雖然仍相當寬鬆)





2. 根據上述，歸納可知由一行增為兩行時最多有三種新的組合。

令 M(n)為一函數，其值為數列長 $\mathrm{n}\{\mathrm{n}|\frac{n}{2}\in \mathrm{N}\ \wedge\ \mathrm{n}\geq 4\}$的多方塊的形總數。

則因為寫成奇偶數分列的形式，且(不妨令形的首、尾行為$\begin{bmatrix}+ & \cdots & -\\ + & \cdots & -\end{bmatrix}$)倒數第二行的兩個元可由簡化繁為三種可能(如上表)

$$有\ \mathrm{M(n+2)}\leq 3(\frac{n}{2}-2)M(n)$$

$$\because\ \mathrm{M(n+2)}\leq 3(\frac{n}{2}-2)M(n)=(3)^{\frac{n-4}{2}}\left(\frac{n-4}{2}\right)!\,\mathrm{M}(4)$$

$$又\because \mathrm{M}(4)=1$$

$$\therefore \mathrm{M(n)}\leq (3)^{\frac{n-6}{2}}\left(\frac{n-6}{2}\right)!$$

3. <圖形的旋轉與鏡射>(p.2)中提到，當數列中全部奇數項或全部偶數項同時變號，並將數列中的排序提前($a'_i=a_{i+1}$)，代表圖形經過旋轉。根據 one-sided polyomino 的定義，旋轉後仍和旋轉前的圖形為同一種圖形，因此不同的正負號排序也有可能代表相同的形，但遵守以下關係：

$$\begin{bmatrix} s_1 & s_3 & s_5 & \cdots & s_{n-3} & s_{n-1} \\ s_2 & s_4 & s_6 & \cdots & s_{n-2} & s_n \end{bmatrix} \overset{\text{def}}{=\!=} \begin{bmatrix} s_2 & s_4 & s_6 & \cdots & s_{n-2} & s_n \\ -s_3 & -s_5 & -s_7 & \cdots & -s_{n-1} & -s_1 \end{bmatrix}$$

四、 多方塊種類函數與範圍

(一). 多方塊種類函數

1. **【定義 1】**本文中將定義多方塊種類函數為 P(n , t)， n 為數列長度，t 為連方數，且 P(n , t)的定義域為$\{(n,t)|\ \frac{n}{2} \in N, n \geq 4, t \in N,\ t \neq 0\}$

【定義 2】$\sigma_0(t)$為除數函數，表正整數 t 的正因數個數

(二). 固定形，t 連方種類的討論

下文中，中括號表高斯符號。

1. 數列長度為 4：

數列長度為 4 時，只有一種形$\begin{bmatrix} + & - \\ + & - \end{bmatrix}$，又有$a_1 \times a_2 = t$(見 p.5)，故 P(4,t)即為正整數 t 有多少兩個正因數乘積的組合。

對於所有質數 p，整數p^m有 m+1 個因數，故知p^m連方在數列長度為 4 時有$\mathrm{P}(4, p^m) = [\frac{m+2}{2}]$種

【引理 1】 $\mathrm{P}(4,\mathrm{t}) = [\frac{\sigma_0(t)+1}{2}]$

證明 n=4 時，a_1與 a_3、a_2與 a_4為相反數，又$a_1 \times a_2 = t$故多方塊數列種類數為正整數 t 的因數個數的一半(取高斯符號)。

已知$\sigma_0(t) \leq 2\sqrt{t}$，故上界不妨寫作 $\mathrm{P}(4,\mathrm{t}) \leq \left[\frac{2\sqrt{t}+1}{2}\right]$

2. 數列長度為 6：

數列長度為 6 時，由數列長度 4 的形$\begin{bmatrix} + & - \\ + & - \end{bmatrix}$只可延展為$\begin{bmatrix} + & + & - \\ + & + & - \end{bmatrix}$、$\begin{bmatrix} + & - & - \\ + & + & - \end{bmatrix}$，但因圖形的旋轉規則(見報告 p.2)$\begin{bmatrix} + & - & - \\ + & + & - \end{bmatrix} \equiv \begin{bmatrix} + & + & - \\ + & + & - \end{bmatrix}$，故知數列長度為 6 時，也只有一種形。又有$a_1 \times a_2 - a_4 \times a_5 = t$(見 p.6)，且$a_1 + a_3 + a_5 = 0$，$a_3 \neq 0$，故$|a_1| \neq |a_5|$

故知 P(6,t)即是當正整數 t 拆成 t'和(t-t')二正整數時，t'和(t-t')有多少組因數乘積的總和。

【引理 2】$P(6,t)=\frac{1}{2}\sum_{i=1}^{t-1}[\sigma_0(i)\times\sigma_0(t-i)-\sigma_0(\gcd(i,t-i))]$

證明 n=6 時，$a_1\times a_2-a_4\times a_5=t$，但$\because|a_1|\neq|a_5|$
$\therefore P(6,t)$為二正整數因數個數乘積，再扣掉最大公因數的因數個數。

【引理 3】$\sum_{i=0}^{t}\sqrt{ti-i^2}\leq\frac{\pi}{8}t^2$

$$\because\int_0^t\sqrt{ti-i^2}\,di=\int_0^t\sqrt{\left(\frac{t}{2}\right)^2-\left(i-\frac{t}{2}\right)^2}\,di$$

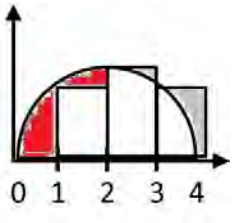
Let $i=\frac{t}{2}(1+\sin\theta)$

$$\therefore\int_0^t\sqrt{\left(\frac{t}{2}\right)^2-\left(i-\frac{t}{2}\right)^2}\,di=\frac{t^2}{4}\int_{-\frac{\pi}{2}}^{\frac{\pi}{2}}\cos^2\theta\,d\theta=\frac{t^2}{4}\int_{-\frac{\pi}{2}}^{\frac{\pi}{2}}\frac{\cos 2\theta+1}{2}\,d\theta$$

$$=\frac{t^2}{4}\left[\frac{\sin 2\theta}{4}+\frac{\theta}{2}\right]_{-\frac{\pi}{2}}^{\frac{\pi}{2}}=\frac{\pi}{8}t^2$$

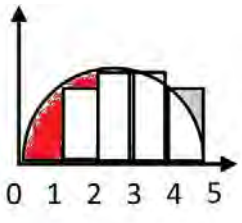
註：$\sum_{i=0}^{t}\sqrt{ti-i^2}\leq\frac{\pi}{8}t^2$的圖形說明

(1) 當 t 為偶數時，如 t=4 時



$$\sum_{i=0}^{4}\sqrt{i(4-i)}$$
$$=\sqrt{3}+\sqrt{2\times 2}+\sqrt{3}$$
$$=\textit{圖形中三條長方形面積}$$
$$<\textit{半圓面積}=\frac{4^2\pi}{8}=2\pi$$

(2) 當 t 為奇數時，如 t=5 時



$$\sum_{i=0}^{5}\sqrt{i(5-i)}$$
$$=\sqrt{4}+\sqrt{2\times 3}+\sqrt{2\times 3}+\sqrt{4}$$
$$=\textit{圖形中四條長方形面積}$$
$$<\textit{半圓面積}=\frac{5^2\pi}{8}=\frac{25\pi}{8}$$

由圖形的對稱性可以說明紅色區域面積大於黑色區域面積，並可以推廣到一般自然數 t 都成立，所以$\sum_{i=0}^{t}\sqrt{ti-i^2}\leq\frac{\pi}{8}t^2$恆成立。

並且當 t 愈大左式與右式愈接近，
兩者偏差在$t>165$後小於千分之一。

【定理 1】

$$\mathrm{P}(6,\mathrm{t}) < \frac{\pi}{4}t^2 - \left\{ C_1 + \frac{2}{3}\left(\frac{t}{2}\right)^{\frac{3}{2}} + \frac{1}{2}\sqrt{\frac{t}{2}} + \frac{1}{24}\sqrt{\frac{2}{t}} - \frac{1}{24}\left\{ \frac{\left[\sqrt{\frac{t}{2}}+\sqrt{\frac{t}{2}+1}\right]^{-5}}{\sqrt{\frac{t}{2}(\frac{t}{2}+1)}} + \frac{\left[\sqrt{\frac{t}{2}+1}+\sqrt{\frac{t}{2}+2}\right]^{-5}}{\sqrt{(\frac{t}{2}+1)(\frac{t}{2}+2)}} + \cdots \right\} \right\}$$

其中$C_1 = \frac{1}{4\pi}\left(\frac{1}{1\sqrt{1}} + \frac{1}{2\sqrt{2}} + \frac{1}{3\sqrt{3}} + \cdots\right) \simeq 0.159$

證明　根據【引理 2】:

$$\because \mathrm{P}(6,\mathrm{t}) = \frac{1}{2}\sum_{i=1}^{t-1}[\sigma_0(i) \times \sigma_0(t-i) - \sigma_0(\gcd(i,t-i))]$$

，又已知$\sigma_0(t) \le 2\sqrt{t}$

∴將函數$\sigma_0(t)$都代換為$\sqrt{t}$

$\therefore \mathrm{P}(6,\mathrm{t}) < 2\sum_{i=0}^{t}\left(\sqrt{i} \times \sqrt{t-i}\right) - \sum_{i=1}^{\left[\frac{t}{2}\right]}\sqrt{i}$　(此式之中中括號表高斯符號)

由【引理 3】證明$\sum_{i=0}^{t}\sqrt{ti-i^2} \le \frac{\pi}{8}t^2$，

由 Srinivas Ramanujan 在 1915 年得到的結論—(Srinivas Ramanujan[9])

$$\sum_{i=1}^{\left[\frac{t}{2}\right]}\sqrt{i} = C_1 + \frac{2}{3}\left(\frac{t}{2}\right)^{\frac{3}{2}} + \frac{1}{2}\sqrt{\frac{t}{2}} + \frac{1}{24}\sqrt{\frac{2}{t}} - \frac{1}{24}\left\{ \frac{\left[\sqrt{\frac{t}{2}}+\sqrt{\frac{t}{2}+1}\right]^{-5}}{\sqrt{\frac{t}{2}(\frac{t}{2}+1)}} + \frac{\left[\sqrt{\frac{t}{2}+1}+\sqrt{\frac{t}{2}+2}\right]^{-5}}{\sqrt{(\frac{t}{2}+1)(\frac{t}{2}+2)}} + \cdots \right\}$$

並且在 t 足夠大時此式近似為

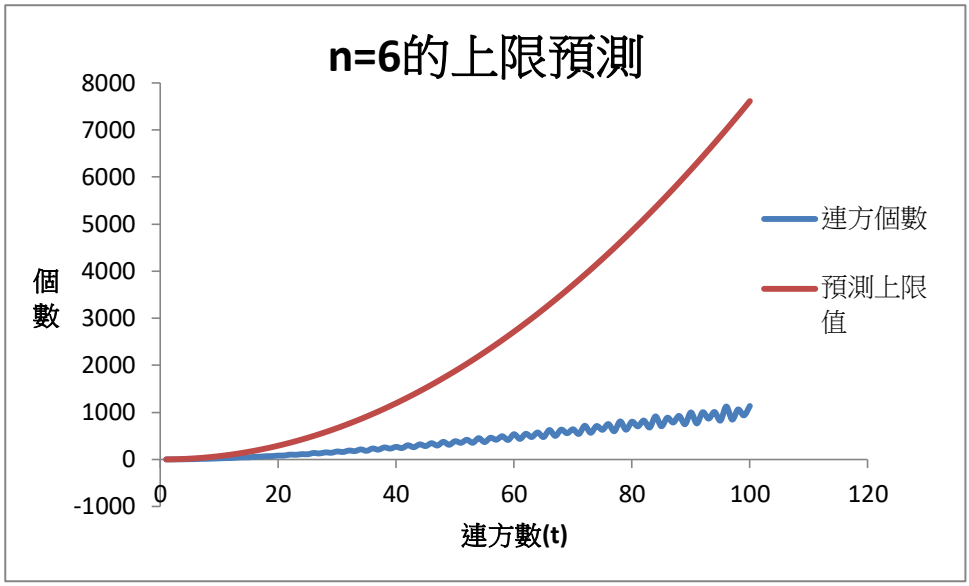
$$C_1 + \frac{2}{3}\left(\frac{t}{2}\right)^{\frac{3}{2}} + \frac{1}{2}\sqrt{\frac{t}{2}} + \sqrt{\frac{2}{t}}\left(\frac{1}{24} - \frac{1}{480t^2} + \frac{1}{576t^4} - \cdots\right)$$

$\therefore \mathrm{P}(6,\mathrm{t}) <$

$$\frac{\pi}{4}t^2 - \left\{ C_1 + \frac{2}{3}\left(\frac{t}{2}\right)^{\frac{3}{2}} + \frac{1}{2}\sqrt{\frac{t}{2}} + \frac{1}{24}\sqrt{\frac{2}{t}} - \frac{1}{24}\left\{ \frac{\left[\sqrt{\frac{t}{2}}+\sqrt{\frac{t}{2}+1}\right]^{-5}}{\sqrt{\frac{t}{2}(\frac{t}{2}+1)}} + \frac{\left[\sqrt{\frac{t}{2}+1}+\sqrt{\frac{t}{2}+2}\right]^{-5}}{\sqrt{(\frac{t}{2}+1)(\frac{t}{2}+2)}} + \cdots \right\} \right\}$$

3. 數列長度為 6 上限預測：

由於$\sqrt{t}$與$\sigma_0(t)$的差距隨 t 增加而增加，故 n=6 與代入$\sqrt{t}$預測的個數上限隨 t 增加而平方增加。



4. 數列長度為 8：

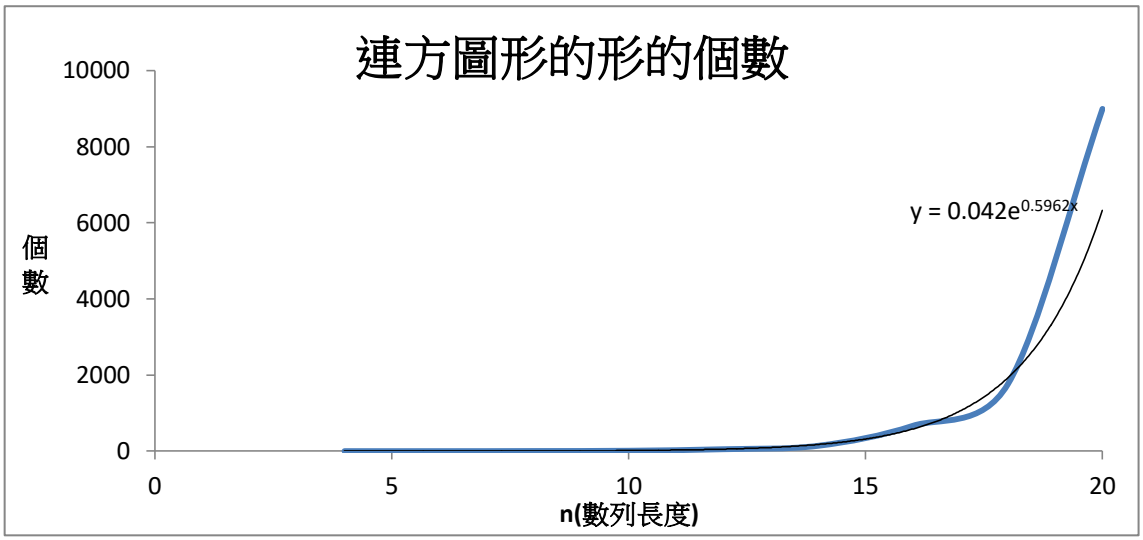
數列長度為 8 時，由數列長度 6 的形$\begin{bmatrix}+ & + & -\\ + & + & -\end{bmatrix}$只可延展為$\begin{bmatrix}+ & + & + & -\\ + & + & + & -\end{bmatrix}$、$\begin{bmatrix}+ & + & - & -\\ + & + & + & -\end{bmatrix}$、$\begin{bmatrix}+ & + & + & -\\ + & - & + & -\end{bmatrix}$、$\begin{bmatrix}+ & - & + & -\\ + & + & + & -\end{bmatrix}$、$\begin{bmatrix}+ & + & - & -\\ + & + & - & -\end{bmatrix}$，但因圖形的旋轉規則$\begin{bmatrix}+ & - & + & -\\ + & + & + & -\end{bmatrix} \equiv \begin{bmatrix}+ & + & + & -\\ + & - & + & -\end{bmatrix}$，故知數列長度為 8 時，只有 4 種形。又有 $a_1 \times a_8 + a_3 \times (a_4 + a_6) + a_5 \times a_6 = t$。

(三). 形的種類數

1.

形的種類數					
n (數列長度)	4	6	8	10	12
種類數	1	1	4	12	52
n (數列長度)	14	16	18	20	
種類數	142	668	1757	8996	

2. 形的種類數計算



所有的形，都可表達為一組二進位數，例如 n=6 的形$\begin{bmatrix}+ & + & -\\ + & + & -\end{bmatrix}$即是 111100，十進位表示為 60，依照旋轉的規則，奇數位變號(二進位制)，每一位提前，最高位數變為最後一位。

例如，$111100 \equiv 101100 \equiv 001100 \equiv 001101 \equiv 001111 \equiv 001011 \equiv 000011 \equiv 010011 \equiv 110011 \equiv 110010 \equiv 110000 \equiv 110100$，十進位表示為$60 \equiv 44 \equiv 12 \equiv 13 \equiv 15 \equiv 11 \equiv 3 \equiv 19 \equiv 51 \equiv 50 \equiv 48 \equiv 52$。並根據形定義，n=6 時首兩位為 1(正)，末兩位為 0(負)，所以以十進位表示，只有$2^{\frac{n}{2}-1} + 2^{\frac{n}{2}-2} = 3 \times 2^{\frac{n}{2}-2}$到$\sum_{i=0}^{\frac{n}{2}-1} 2^i = 2^{\frac{n}{2}} - 1$的整數才在計算的範圍內，共$2^{\frac{n}{2}-2} + 2$個數。

上述例子中的等價關係表示重複， 而代表形的二進位數，需滿足的條件和以正負號表示的方式相同，透過縮減規則，若第 i 項與 i+2 項相同，可將 i、i+1、i+2 項縮減，只留下第 i 項，多次縮減到無法縮減時，剩下 1100 的形滿足條件。

找尋形的種類數，按照上述的程序，以 C 語言運算，程式如附錄三。

3. 形的種類數討論

對於數列長度 n 的連方圖形，滿足條件的整數最多只有$2^{\frac{n}{2}-2} + 2$個，扣除重複和不滿足條件的整數後，形的個數函數 M(n)的趨勢線約可表示為$M(n) \approx C_1 e^{C_2 n}$其中$C_1$、$C_2$為常數。以現有資料整理出的趨勢約為$0.042e^{0.5962n}$

五、以數列討論有限矩形方格的填滿。

(一). 數個任意圖形填滿方型平面方格問題討論

1. 有限矩形方格，可以存在許多小的子圖形，而其中最大的子圖形等於此有限矩形方格。
2. 判斷最後圖形的來源及整理產生最後等於此矩形方格的圖形，可以直接判斷小於平面方格的圖形及找相同的圖形。

(二). 討論數個相同圖形是否可以填滿矩形方格

1. 觀察對稱軸 a 的連方組成性質，可知 0 列的方格數必須對於 a 對稱軸的兩側平分，而 1 列的方格數也必須於兩側平分，以此類推。對於對稱軸 b 也有此性質(ㄅ行方格數於 b 軸兩側平分…)。

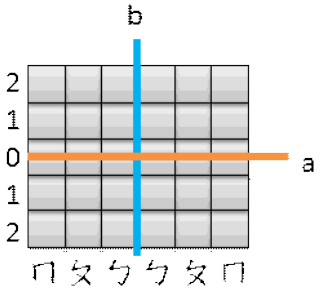


Fig.13

2. 一矩形只會有兩對稱軸平行方格介面(如 a,b)。
3. 若一圖形可組成最小矩形為 S，則此圖形必可以拼成矩形 nS。
4. 由 1、2 兩點可知能被同一種非矩形多方塊組成的矩形方格必為奇數邊×偶數邊，或者是偶數邊×偶數邊。
5. 奇數邊×偶數邊的矩形方格必是由 2 倍的最大相同非矩形連方組成，否則將須使用含有 1/2 方格的非矩形連方來組合。而偶數邊×偶數邊的矩形方格則可以 2 或 4 個的最大相同非矩形連方組成。簡言之，一非矩形多方塊若不能以 2 或 4 個同樣的圖形拼成一矩形，則此圖形必不能以自身拼成一矩形。
6. 從這些性質可以得知，若相同的兩個圖形可以拼成一個完整矩形，必須在數列中找到對稱的一連續子數列，且該子數列僅比原數列少三項；若相同的四個圖形可以拼成一個完整矩形，則是兩個該圖形先合併後產生的圖形，必須在這個新數列中找到對稱的一連續子數列，且該子數列僅比這個新數列少三項。

(三). 命題、任意一種非矩形多方塊皆無法以相同三個，旋轉合併拼成一矩形

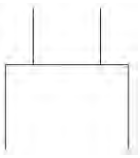
1. 證明：

將此非矩形多方塊表示為數列 $A(a_1,a_2,a_3,\ldots,a_n)$

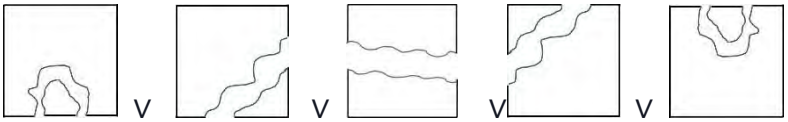
且不妨使此圖形第一次合併中，兩個 A 分別從 a_2 項和 a_{t+1} 項開始，共重疊刪去 x 項。將圖形先兩兩合併再合併第三個，討論過程中各項的關係及合理性，再分開討論圖形有無旋轉的情況。

若欲將三個 A 合併為矩形，可能滿足 a、b 或 c 三種情況:
(其中 x,y 分別代表第一次及第二次合併中重疊的項數。)

a. 第一次兩個 A 合併的數列長度 n'= $2n$(第一次合併時只有接觸到一項且沒有延續其他邊)且第二次再合入一個 A 的數列長度



$n'' = n + n' - 2y - 2 = 4$
$\therefore$ n'' $= 4 =$ 3n − 2y − 2
$\Rightarrow$ 3n − 2y = 6又{[$y = n - 5$] ∨ [$y = n - 4$] ∨ [$y = n - 3$] ∨ [$y = n - 2$] ∨ [$y = n - 1$]}如下圖。



將 y 帶入前式，得n = −4 ∨ −2 ∨ 0 ∨ 2 ∨ 4，由於與 n ≥ 6 矛盾，故此種合併方法不可能拼成矩形。

b. 第一次兩個 A 合併的數列長度 n'= $2n - 2x - 2$且第二次再合入一個 A 的數列長度$n'' = n + n' = 4$ (第二次合併時只有接觸到一項且沒有延續其他邊)
$\therefore$ n'' = 4 = 3n − 2x − 2 $\Rightarrow$ 3n − 2x = 6 $\Rightarrow$ 3n = 2x + 6又{[$x = n - 5$] ∨ [$x = n - 4$] ∨ [$x = n - 3$] ∨ [$x = n - 2$] ∨ [$x = n - 1$]}將 x 帶入前式，得 n = −4 ∨ −2 ∨ 0 ∨ 2 ∨ 4，由於與n ≥ 6 矛盾，故此種合併方法不可能拼成矩形。

c. 第一次兩個 A 合併的數列長度 n'= $2n - 2x - 2$且第二次再合入一個 A 的數列長度$n'' = n + n' - 2y - 2 = 4$
$\therefore$ n'' = 4 = 3n − 2x − 2y − 4 $\Rightarrow$ 3n − 2x − 2y = 8
$\because$ 3n − 2x − 2y = 8 ∧ {[$y = n - 4$] ∨ [$y = n - 3$] ∨ [$y = n - 2$] ∨ [$y = n - 1$]} (根據合併原理)
將 y 帶入前式，得知若要進行三個相同圖形的合併，並使合併後新數列的長度等於 4(表示該圖形為矩形)，第一次合併的消去部分長度 x 只可能滿足:

[2x = n]∨[2x = n − 2]∨[2x = n − 6]

針對這四種可能性討論:

(1) 當[2x = n]:
若兩圖形合併時沒有經過旋轉，在此種合併中
第一次合併:
(為了避免代號過於複雜，故以此例子說明)

+) ($a_{g+x+2},\ldots,a_n$, a_1, $a_2,\ldots$, a_{g-1},a_{g+x+1}, a_{g+x}, …, a_{g+1}, a_g)

= ([a_1+a_{g+x+1}], $a_{g+x+2}\ldots$, a_n, a_1, $a_2,\ldots$, a_{g-1}, [a_{x+2}+a_g],a_{x+3},a_{x+4},…a_n)

∵ 在合併中 $\begin{Bmatrix} a_2 = -a_{g+x} \\ a_3 = -a_{g+x-1} \\ \cdots\cdots \\ a_{x+1} = -a_{g+1} \end{Bmatrix}$，即 $a_{2+u} = -a_{g+x-u}$ ，u ∈ {0,1,2,……,x-1}

且∵ [2x=n]，而(以此合併為例) |1-(g+x+1)| 及 |(x+2)-g| 皆必小於 n(即 2x)。

∴ ∃u 使$(a_{2+u}, a_{g+x-u}) = (a_{x+g+1}, a_1) \vee (a_g, a_{x+2})$，

即$a_{X+g+1} = -a_1 \vee a_g = -a_{x+2}$，其中一個□的長度不存在，

與合併原則矛盾

而當2 | (x + g)時，更有

$a_{\frac{(2+x+g)}{2}} \vee a_{\frac{(2+x+g+n)}{2}} = a_{2+u} = -a_{x+g-u} = -a_{2+u}$的矛盾

由此可知兩相同的圖形在沒有旋轉的情況下，不可能以[2x=n]的形式合併。

若圖形合併時有旋轉

當2 ∤ x，則第一次的合併式子可改寫為

(-a_1 , a_2 , -a_3 , … , a_{x+1}, -a_{x+2}, $a_{x+3}\ldots a_n$)

+) ($a_{g+x+2}\ldots$, a_n, a_1, $a_2,\ldots$, a_{g-1}, a_{g+x+1}, a_{g+x}, …,a_{g+2}, a_{g+1}, a_g)

= ([-a_1+a_{g+x+1}], $a_{g+x+2}\ldots$, a_n, a_1, $a_2,\ldots$, a_{g-1}, [-a_{x+2}+a_g],a_{x+3},-a_{x+4},…a_n)

當4 ∤ n ∧ 2 | g可由合併原理知：

令 u 為適當正整數，

$\begin{Bmatrix} a_2 = -a_{g+x} \\ a_3 = a_{g+x-1} \\ a_4 = -a_{g+x-2} \\ a_5 = a_{g+x-3} \\ \cdots\cdots \\ a_{x+1} = a_{g+1} \end{Bmatrix}$

∵ when 2 | 2 + u , 2 ∤ (g + x − u) ; when 2 ∤ 2 + u , 2 | (g + x − u)

∴ $\begin{Bmatrix} a_{2+u} = -a_{g+x-u} \\ \wedge \\ a_{2+u} = a_{g+x-u} \end{Bmatrix}$

當4 ∤ n ∧ 2 ∤ g時，可由合併原理知：

∵ $\begin{Bmatrix} a_2 = -a_{g+x} \\ a_3 = a_{g+x-1} \\ a_4 = -a_{g+x-2} \\ a_5 = a_{g+x-3} \\ \cdots\cdots \\ a_{x+1} = a_{g+1} \end{Bmatrix}$

∴ $2 \mid \frac{2+g+x}{2}$ ∴ $a_{\frac{2+g+x}{2}} = -a_{\frac{2+g+x}{2}}$，與定義矛盾。

當$4 \mid n \wedge 2 \mid g$或當$4 \nmid n \wedge 2 \nmid g$則可由合併原理知：

若圖形合併時有旋轉，且當$4 \mid n \wedge 2 \nmid g$或當$4 \mid n \wedge 2 \mid g$，則第一次的合併式子可改寫為

($-a_1$, a_2 , $-a_3$, …,$-a_{x+1}$, a_{x+2}, a_{x+3}…a_n)

+) (a_{g+x+2}…, a_n, a_1, a_2,…, a_{g-1}, a_{g+x+1}, a_{g+x}, …,a_{g+2}, a_{g+1}, a_g)

= ($[-a_1+a_{g+x+1}]$, a_{g+x+2} ,…, a_n, a_1, a_2,…, a_{g-1}, $[a_{x+2}+a_g]$,$-a_{x+3}$,a_{x+4},…a_n)

當$4 \mid n \wedge 2 \nmid g$可由合併原理知：

令 u 為適當正整數，

$$\left\{\begin{array}{c} a_2 = -a_{g+x} \\ a_3 = a_{g+x-1} \\ a_4 = -a_{g+x-2} \\ a_5 = a_{g+x-3} \\ \cdots\cdots \\ a_{x+1} = a_{g+1} \end{array}\right\}$$

∵ when $2 \mid 2+u$, $2 \nmid (g+x-u)$; when $2 \nmid 2+u$, $2 \mid (g+x-u)$

$$\therefore \left\{\begin{array}{c} a_{2+u} = -a_{g+x-u} \\ \wedge \\ a_{2+u} = a_{g+x-u} \end{array}\right\}$$

當$4 \mid n \wedge 2 \mid g$可由合併原理知：

$$\left\{\begin{array}{c} a_2 = -a_{g+x} \\ a_3 = a_{g+x-1} \\ a_4 = -a_{g+x-2} \\ a_5 = a_{g+x-3} \\ \cdots\cdots \\ a_{x+1} = a_{g+1} \end{array}\right\}$$

並且當圖形進行第二次合併時可表示為

(a_w , a_{w+1} , a_{w+2} , a_{w+3} , … , $-a_{x+1}$, a_{x+2} , $-a_{x+3}$,…, a_n , a_1 , … , a_{w-1})

(a_n , …,$-a_{x+3}$, $[a_{x+2}+a_g]$, …, a_2 , a_1 , a_n ,……, a_{g+x+2}, $[-a_1+a_{g+x+1}]$)

= ($[a_{w+2}+a_n]$, a_w , a_{w+1} , $[a_{w-1}+-a_1+a_{g+x+1}]$)

令 u 為適當正整數，

∵ $2x = n$

因為有 $\left[\frac{x}{2}\right]$ 項新數列與 $\left[\frac{x}{2}\right]$ 項原數列同值異號，

另有 $\left[\frac{x}{2}\right]$ 項新數列與另外 $\left[\frac{x}{2}\right]$ 項原數列同值同號

∴ $a_{w-2} = -a_{g+x+2} = a_{n-g-x+u} = \cdots = -a_{w-2}$

∴ 在符合合併原理的情況下，必產生$a_u = -a_u$ 與定義矛盾。

綜合上面三種 x 的可能性，知此種合併方法不可能拼成矩形。

(2) $[2x = n - 2]$、 $[2x = n - 4]$及$[2x = n - 6]$

依同樣方法亦可證得。

(3) 總結以上，可得知任意三個相同的非矩形多方塊合併無法拼成矩形。

肆、 結論

一、計算多方塊的形與種類數的規則和限制

二、主要針對特定形的連方種類數研究，可推得含除數函數之個數公式，得用以簡化運算程序，並且預測函數範圍。

三、由以知條件$\sigma_0(t) \le 2\sqrt{t}$，可推得

$$P(6,t) < \frac{\pi}{4}t^2 - \left\{ C_1 + \frac{2}{3}\left(\frac{t}{2}\right)^{\frac{3}{2}} + \frac{1}{2}\sqrt{\frac{t}{2}} + \frac{1}{24}\sqrt{\frac{2}{t}} - \frac{1}{24}\left\{ \frac{\left[\sqrt{\frac{t}{2}}+\sqrt{\frac{t}{2}+1}\right]^{-5}}{\sqrt{\frac{t}{2}(\frac{t}{2}+1)}} + \frac{\left[\sqrt{\frac{t}{2}+1}+\sqrt{\frac{t}{2}+2}\right]^{-5}}{\sqrt{(\frac{t}{2}+1)(\frac{t}{2}+2)}} + \cdots \right\} \right\}$$

四、推得多方塊形的種類數的上界$M(n) \le (3)^{\frac{n-6}{2}}\left(\frac{n-6}{2}\right)!$

五、多方塊的形的數量約隨著指數增加

六、以連方數列運算證明任意一種非矩形多方塊皆無法以相同三個，旋轉合併拼成一矩形

伍、 未完成研究

一、完成函數對於 n 為正整數的所有函數 P(n , t)相當困難(考慮正因數個數問題)，但分別做出有限個函數卻可行，提升此正整數 n 的大小，可能是可以完成的問題。

二、該圖形命名方法若可以從不同的正多邊形、不同維度方面延伸，也許能夠處理更貼近現實生活問題的應用。

三、能夠填滿矩形的多方塊在所有多方塊裡的比例高低是 Polygon 在平面上或 Polycube 在空間上填滿特定範圍的相似問題。假設填滿的狀況使空間分布均勻與穩定，則了解這樣的多方塊比例就能初步了解條件下選定範圍的穩定性。

陸、 參考資料

1.【摘要資料】Weisstein, Eric W. "Polyomino." From MathWorld--A Wolfram Web Resource. http://mathworld.wolfram.com/Polyomino.html

2.【學位論文】D. Hugh REDELMEIER . (1981). COUNTING POLYOMINOES: YET ANOTHER ATI'ACK. Discrete Mathematics ,36 ,191-203
3.【學位論文】D. Beauquier (1)M. Nivat (2).(1991-12). On translating one polyomino to tile the plane. L.I.T.P., Université Paris VI, 2 Place Jussieu, 75251, Paris Cédex 05, France
4.【學位論文】Xiangqian Zhou(1) Heping Zhang(2).(2016-6-19). A minimax result for perfect matchings of a polyomino graph. Discrete Applied MathematicsVolume 206, 19 June 2016, Pages 165–171
5.【期刊文章】JACOB TURNER.(2016-2-19). ON SUBPACKINGS OF POLYOMINO PACKINGS
6.【期刊文章】Svjetlan Feretic'. (1998). A new way of counting the column-convex polyominoes by perimeter
7.【期刊文章】Svjetlan Feretic'. (2000). An alternative method for q-counting directed column-convex polyominoes
8.【期刊文章】Mireille Bousquet-M´elou1 and Richard Brak2(2008-11-26). Exactly solved models of polyominoes and polygons
9.【期刊文章】Srinivas Ramanujan. (1915). On the sum of the square roots of the first n natural numbers
10.【網頁資料】http://mathworld.wolfram.com/Monomino.html /WolframMathWorld
11.【網頁資料】http://mathworld.wolfram.com/Domino.html /WolframMathWorld
12.【網頁資料】http://mathworld.wolfram.com/Triomino.html /WolframMathWorld
13.【網頁資料】http://mathworld.wolfram.com/Tetromino.html /WolframMathWorld
14.【網頁資料】http://mathworld.wolfram.com/Pentomino.html /WolframMathWorld
15.【網頁資料】http://mathworld.wolfram.com/Hexomino.html /WolframMathWorld
16.【網頁資料】http://mathworld.wolfram.com/Heptomino.html /WolframMathWorld
17.【中小學科展】陳佑昕(1)陳韋綸(2)李昕(3).(2016).PODO123.(polyomino do math)

柒、　附錄:

一、數列長度為 4 的連方圖形種類計算程式(部分)：

```
# include <cstdio>
main()
{
    int t=250;
    for(int i=1990;i<2200;i++)
    {
        int count=0;
        for(int j=1;j*j<=i;j++)
        {
            if(i%j==0)    count+=2;
            if(j*j==i)  count--;
        }
        printf("%d\n",count);
    }
}
```

二、數列長度為 6 的連方圖形種類計算程式(部分)：

```
#include <cstdio>
#include <algorithm>
using namespace std;

const int SIZE = 2000+10;
int A[SIZE];
int dp[SIZE];


int main (void)
{

	freopen("fuck.txt","r",stdin);

	for(int i=1;i<=2000;i++){
		int x;
		scanf("%d",&x);
		//printf("%d\n",x);
		A[i] = x;
	}

	dp[1] = 0;dp[2] = 0;

	int MX	= 0;
	for(int t=3;t<=2000;t++){
		for(int i=1;i<=t-1;i++){

			dp[t] += A[i]*A[t-i] - A[__gcd(i,t-i)] ;
		}

		if(dp[t]>MX)
			MX = dp[t];
		else dp[t] = 0;
	}
	for(int i=1095;i<=1300;i++)
		printf("%d\n",dp[i]);
	return 0;
}
```

三、形的種類數計算

```
# include <stdio.h>
# include <stdlib.h>

main()
{
	int B[2];
	int a,b,i,j,k,l=0,m=0,n=0,o=0,p=0,q=0,r,s,t,u;


	for(a=2;a<=11;a++)
	{


		b=1;
	for(o=1;o<=2*a;o++)
	{
		b=2*b;
	}
	b=b/16;

	int A[2*a],N[2][b];

	for(i=0;i<=(b-1);i++)
	{
		N[0][i]=(b*12)+(4*i);
	}

	for(j=0;j<=b-1;j++)
	{

		if(N[0][j]%4==0)
		{
			l=N[0][j];
			for(k=0;k<=(2*a)-1;k++)
			{
				A[(2*a)-1-k]=l%2;
				l/=2;
			}
```

```
for(o=1;o<=a-2;o++)
{
	for(m=0;m<=a-2;m++)
	{
		if(A[2*m]!=4)
		{
			for(n=0;A[2*(m+n)+2]==4;n++)
			{
			}

			if(A[2*m]==A[2*(m+n+1)])
			{
				A[2*m]=4;
				A[2*m+1]=4;
			}
		}

	}
	for(m=0;m<=a-1;m++)
	{
		if(A[2*m+1]!=4)
		{
			for(n=0;A[2*(m+n)+3]==4;n++)
			{
			}

			if(A[2*m+1]==A[2*(m+n)+3])
			{
				A[2*(m+n)+2]=4;
				A[2*(m+n)+3]=4;
			}
		}
	}
}
p=0;
q=0;
t=0;
for(m=0;m<=a-1;m++)
{
```

```
                if(A[2*m]==A[2*m+1]&&A[2*m]==1)
                {
                    p++;
                }

                if(A[2*m]==A[2*m+1]&&A[2*m]==0)
                {
                    q++;
                }
                if(A[2*m]==4 )
                {
                    t++;
                }
                if(A[2*m+1]==4 )
                {
                    t++;
                }


}

if(p==q&&p==1)
{
    if(t==(2*a)-4)
    {
        N[1][j]=1;
    }
    else
    {
        N[1][j]=0;
    }
}
else
{
    N[1][j]=0;
}

}
```

```
        else
        {
                N[1][j]=0;
        }



}

for(j=0;j<=b-1;j++)
{
        if(N[1][j]==1)
        {

                l=N[0][j];
                for(k=0;k<=(2*a)-1;k++)
                {
                        A[(2*a)-1-k]=l%2;
                        l/=2;
                }
                m=0;
                n=1;
                o=0;
                p=0;
                q=0;
                for(k=0;k<=(2*a)-1;k++)
                {
                        m=m+(n*A[(2*a)-1-k]);
                        n=n*2;
                }

                for(o=0;p!=m;o++)
                {

                        B[0]=A[0];
                        B[1]=A[(2*a)-1];
                        for(q=0;q<=a-2;q++)
                        {
                                A[2*q]=A[2*q+1] ;
```

```
                A[2*q+1]=1-A[2*(q+1)];
            }
            A[(2*a)-2]=B[1];
            A[(2*a)-1]=1-B[0];
            n=1;
            p=0;
            for(k=0;k<=(2*a)-1;k++)
            {
                p=p+(n*A[(2*a)-1-k]);
                n=n*2;
            }
            if(p<=b*12+(4*(b-1)+3)&&p>=b*12)
            {
                if((p-(b*12))/4>j&&p%4==0)
                {
                    N[1][(p-(b*12))/4]=0;
                }
            }


        }
    }

}
s=0;
for(j=0;j<=b-1;j++)
{
s=s+N[1][j];
}
printf("%d    %d    \n",(2*a),s);

}

}
```

【評語】050407

優點：

1. 本研究係用「數列命名的方法」作為骨幹，解決或簡化多方塊（polyominoes，也稱作連方圖形）的相關問題，其結果有趣。
2. 研究方法由多方塊種類，化為數列，在以形的方式呈現，再探討形的種類，有不錯的結果。
3. 本研究引用不少學位論文及期刊文章，足見研究者用心探討本問題。

建議：

1. 需作文獻探討，說明研究方向及方法需交代哪些是自創，哪些是參考自文獻？
2. 研究題材多方塊式及有趣問題，但研究方法部分不完整，且結論不完整。
3. 形的種類數計算主要以電腦得到結果，須說明使用何種軟體求得？
4. 研究目的一及二的數學模式需說明如何建立過程及修改、驗證其正確性。
5. 多方塊種類函數定義不明確，參考文獻列舉方式不合常規，出處不明。

作品海報

壹、摘要

本研究係用「數列命名的方法」(詳見報告p.1)作爲骨幹，解決或簡化Polyominoes的相關問題，例如多方塊的種類、任意的多個相同的多方塊是否可以填滿(嵌滿)矩形(平面)。對於計算多方塊數量問題，可分爲兩部分探討，一是多方塊的形的個數，二是單一多方塊的形其中含有多少多方塊，本研究主要探討此二部分的特性，並簡化運算結構和找出個數範圍。

貳、研究目的

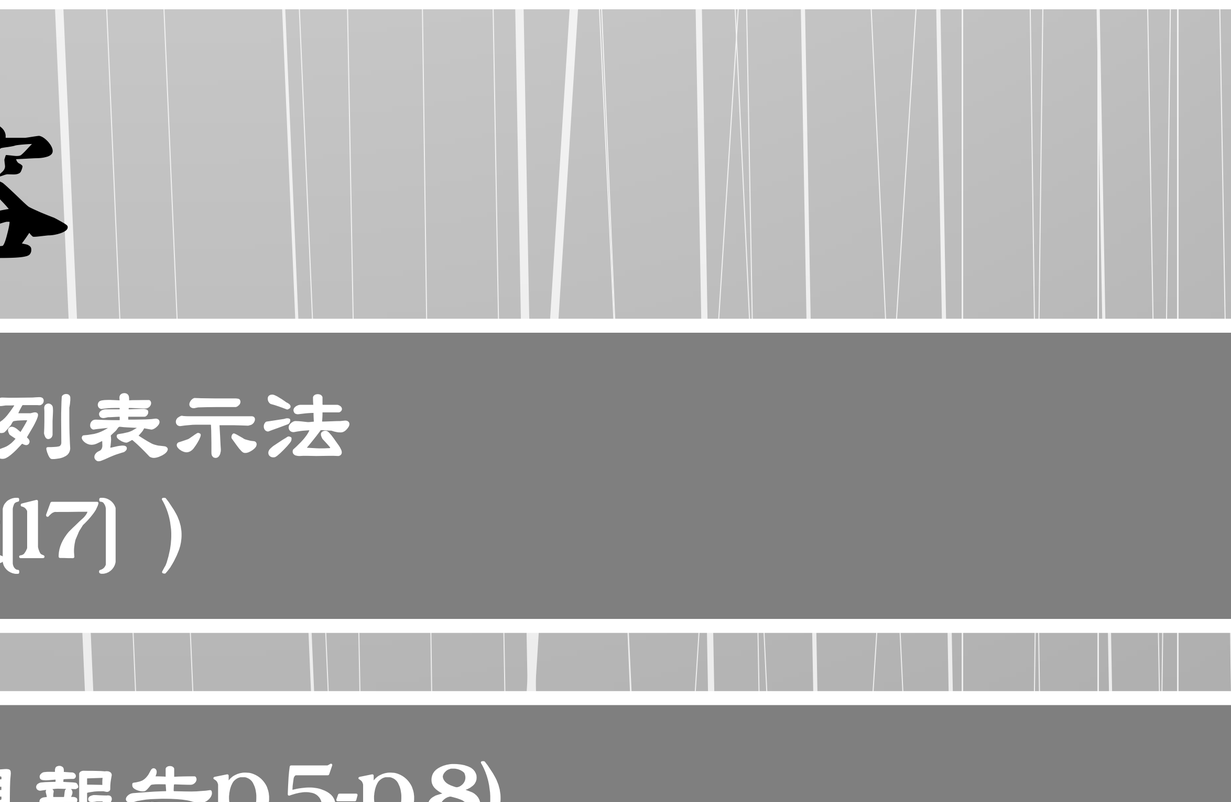
一、找出足以描述polyominoes的數學模型
二、簡化該模型的規則
三、找出polyominoes種類面數的範圍
四、整理相同多方塊在平面上組成矩形的條件

參、研究內容

一、polyominoes 的數列表示法
(見報告p.1-p.4及參考文獻[17])

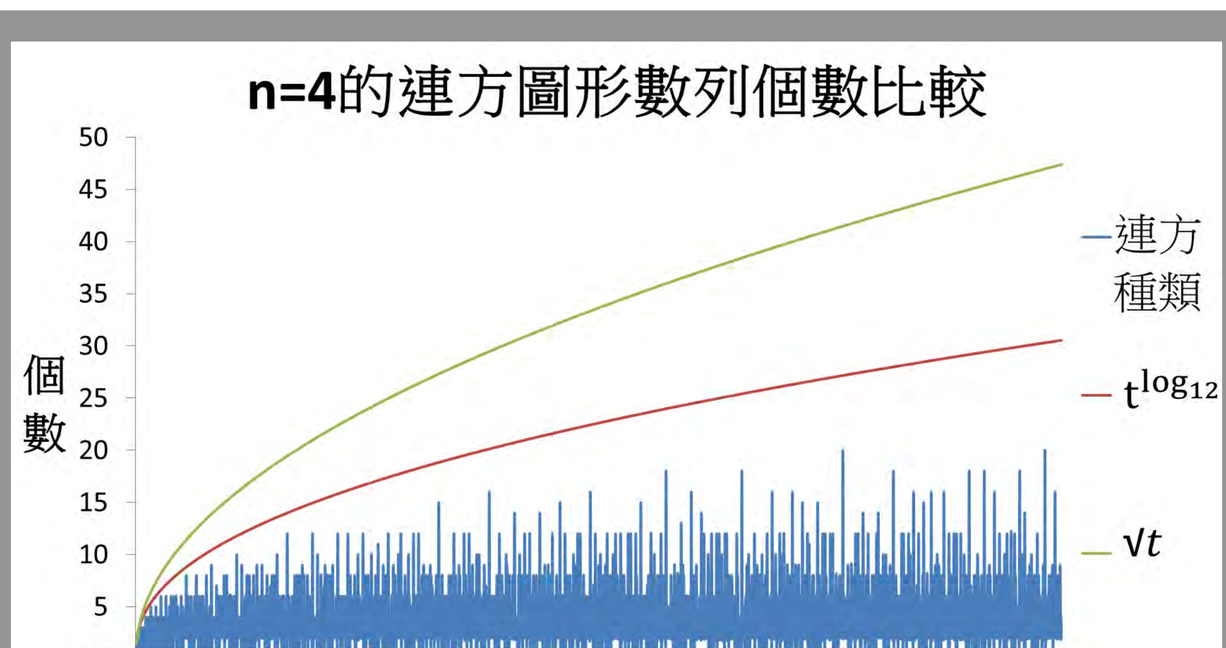
二、以數列窮舉多方塊(見報告p.5-p.8)

1.將圖形轉換爲數列後以程式搜尋所有的連方圖形，可得右圖的結果。



2.透過程式及已知數論上的性質，找出連方圖形個數的範圍。

3.連方圖形個數的範圍顯示，數列長度n增加後，範圍的成長由對數面數增加爲t的冪次數



三、polyominoes形狀的性質(見報告p.8-p.10)

1.形：

- 本研究以「形」指稱polyominoes的輪廓，即是代表多方塊的數列中的正負號序列。例：(1,1,2,2,-1,-1,-2,-2)代表的圖形如右，而此圖形的模爲(+ + + + - - - -)，習慣上可以寫成$\begin{bmatrix} + & + & - & - \\ + & + & - & - \end{bmatrix}$用以分開奇數項和偶數項。

- 多方塊的計數問題包含兩個層面：一是數列長度n的多方塊有幾種形；二是單就一種「形」而言，t連方有多少種類。
- 研究形的特性是爲了瞭解多方塊的種類。

2.形的運算：

- $\begin{bmatrix} + & - \\ + & - \end{bmatrix}$爲最基本單位，代表一個封閉的圖形。
- 當同列相鄰的兩個符號同號，即可簡化爲一個該符號。例：$\begin{bmatrix} + & + \\ + & \end{bmatrix}$可簡化爲$\begin{bmatrix} + \\ + \end{bmatrix}$，而$\begin{bmatrix} + & + \\ - & \end{bmatrix}$亦可簡化爲$\begin{bmatrix} + \\ - \end{bmatrix}$。
- 由已知的兩個元(一行)增加爲四個元(兩行)時，有3種可能。
- M(n)爲一函數，其值爲數列長n$\{n|\frac{n}{2} \in N \wedge n \geq 4\}$的多方塊的形總數。
- 根據形的性質，可以推導出$M(n) \leq (3)^{\frac{n-6}{2}}\left(\frac{n-6}{2}\right)!$

四、多方塊種類函數與範圍(p.11-p.15)

1.多方塊種類函數：

- 【定義1】多方塊種類函數爲P(n，t)， n爲數列長度，t爲連方數，且P(n，t)的定義域爲$\{(n,t)|\ \frac{n}{2} \in N, n \geq 4, t \in N,\ t \neq 0\}$
- 【定義2】$\sigma_0(t)$爲冪次爲0的除數函數，表正整數t的正因數個數

2.固定形，t連方種類的討論：

- 【引理1】 $P(4,t) = [\frac{\sigma_0(t)+1}{2}]$

  證明 n=4時，a1與a3、a2與a4爲相反數，又$a_1 \times a_2 = t$故多方塊數列種類數爲正整數t的因數個數的一半(取高斯符號)。
- 【引理2】$P(6,t) = \frac{1}{2}\sum_{i=1}^{t-1}[\sigma_0(i) \times \sigma_0(t-i) - \sigma_0(\gcd(i,t-i))]$

  證明 n=6時，$a_1 \times a_2 - a_4 \times a_5 = t$，但$\because |a_1| \neq |a_5|$

  $\therefore$P(6,t)爲二正整數因數個數乘積扣掉最大公因數的因數個數。

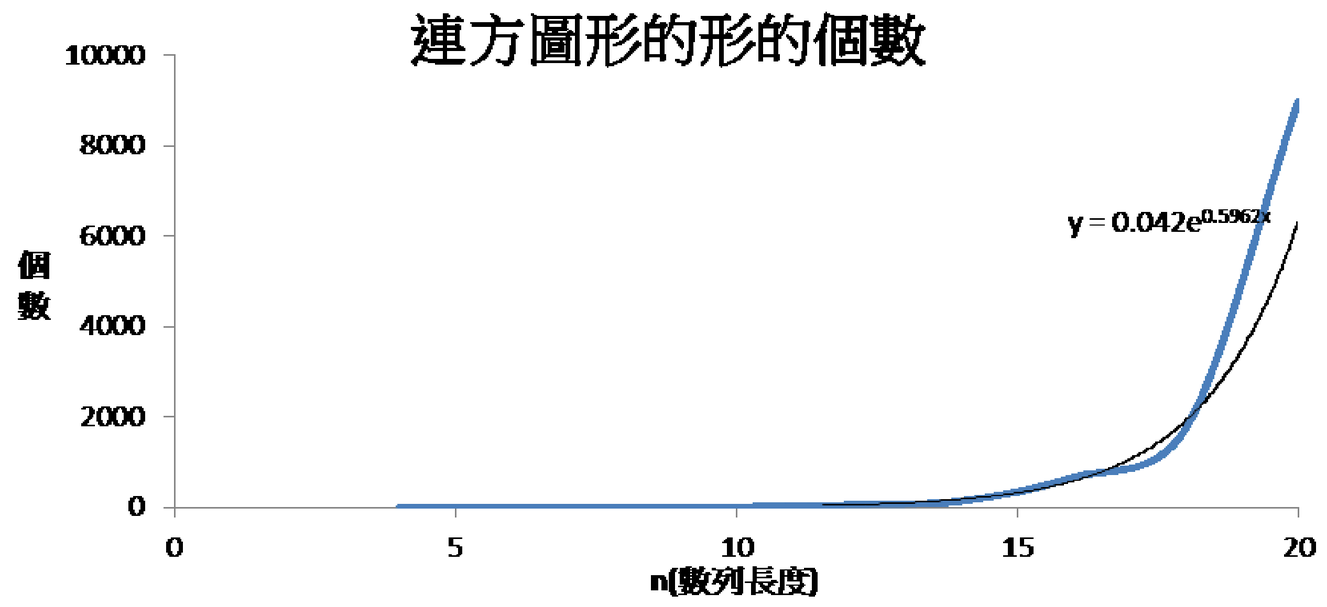
3.形的種類數：

- 形的種類數如右表及右圖
- 所有的形，都可表達爲一組二進位數

例如n=6的形 $\begin{bmatrix} + & + & - \\ + & + & - \end{bmatrix}$ 即是111100，十進位表示爲60

- 依照旋轉的規則，奇數位變號(二進位制)，每一位提前，最高位數變爲最後一位。例：$111100 \equiv 101100 \equiv 001100 \equiv 001101 \equiv 001111 \equiv 001011 \equiv 000011 \equiv 010011 \equiv 110011 \equiv 110010 \equiv 110000 \equiv 110100$
- 上述例子中的等價關係表示重複，搜尋$3 \times 2^{\frac{n}{2}-2}$到$2^{\frac{n}{2}}-1$中的十進位數，刪除重複項可以找出形的總數。

形的種類數					
n (數列長度)	4	6	8	10	12
種類數	1	1	4	12	52
n (數列長度)	14	16	18	20	…
種類數	142	668	1757	8996	…



肆、結論

一、針對特定形的多方塊種類數研究，可推得包含除數函數的個數公式。除數函數的範圍預測得以用於推估多方塊種類數的範圍。

二、目前相對接近n=6的多方塊種類函數的範圍

$$P(6,t) < \frac{\pi}{4}t^2 - \quad C_1 + \frac{2}{3}\left(\frac{t}{2}\right)^{\frac{3}{2}} + \frac{1}{2}\sqrt{\frac{t}{2}} + \frac{1}{24}\sqrt{\frac{2}{t}} - \frac{1}{24}\ \frac{\left[\sqrt{\frac{t}{2}}+\sqrt{\frac{t}{2}+1}\right]^{-5}}{\sqrt{\frac{t}{2}\left(\frac{t}{2}+1\right)}} + \frac{\left[\sqrt{\frac{t}{2}+1}+\sqrt{\frac{t}{2}+2}\right]^{-5}}{\sqrt{\left(\frac{t}{2}+1\right)\left(\frac{t}{2}+2\right)}} + \cdots$$

三、多方塊形的種類數的上界$M(n) \leq (3)^{\frac{n-6}{2}}\left(\frac{n-6}{2}\right)!$

四、多方塊的形的數量約隨著指數增加

五、以連方數列運算證明任意一種非矩形多方塊皆無法以相同三個，旋轉合併拼成一矩形

伍、參考資料

1.【摘要資料】Weisstein, Eric W. "Polyomino." From MathWorld--A Wolfram Web Resource. http://mathworld.wolfram.com/Polyomino.html
2.【學位論文】D. Hugh REDELMEIER . (1981). COUNTING POLYOMINOES: YET ANOTHER ATI' ACK. Discrete Mathematics ,36 ,191-203
3.【學位論文】D. Beauquier (1)M. Nivat (2).(1991-12). On translating one polyomino to tile the plane. L.I.T.P., Université Paris VI, 2 Place Jussieu, 75251, Paris Cédex 05, France
4.【學位論文】Xiangqian Zhou(1) Heping Zhang(2).(2016-6-19). A minimax result for perfect matchings of a polyomino graph. Discrete Applied MathematicsVolume 206, 19 June 2016, Pages 165 – 171
5.【期刊文章】JACOB TURNER.(2016-2-19). ON SUBPACKINGS OF POLYOMINO PACKINGS
6.【期刊文章】Svjetlan Feretic' . (1998). A new way of counting the column-convex polyominoes by perimeter
7.【期刊文章】Svjetlan Feretic' . (2000). An alternative method for q-counting directed column-convex polyominoes
8.【期刊文章】Mireille Bousquet-M´elou1 and Richard Brak2(2008-11-26). Exactly solved models of polyominoes and polygons
9.【期刊文章】Srinivas Ramanujan. (1915). On the sum of the square roots of the first n natural numbers
10.【網頁資料】http://mathworld.wolfram.com/Monomino.html /WolframMathWorld
11.【網頁資料】http://mathworld.wolfram.com/Domino.html /WolframMathWorld
12.【網頁資料】http://mathworld.wolfram.com/Triomino.html /WolframMathWorld
13.【網頁資料】http://mathworld.wolfram.com/Tetromino.html /WolframMathWorld
14.【網頁資料】http://mathworld.wolfram.com/Pentomino.html /WolframMathWorld
15.【網頁資料】http://mathworld.wolfram.com/Hexomino.html /WolframMathWorld
16.【網頁資料】http://mathworld.wolfram.com/Heptomino.html /WolframMathWorld
17.【中小學科展】陳佑昕(1)陳韋綸(2)李昕(3).(2016).PODO123.(polyomino do math)