

---

040418

---

--	--

\_\_\_\_\_ ( Fi bonacci ) \_\_\_\_\_

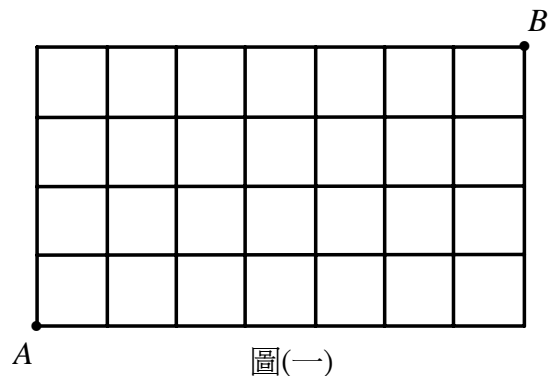
# 費老先生有群蛇，咿呀咿呀啲

## 摘要：

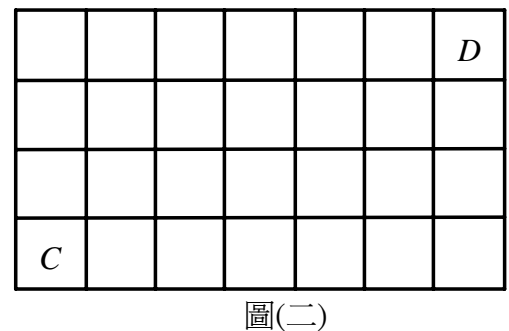
於高二下學期排列組合的課程中，一道有關棋盤形街道走捷徑的問題，改變遊戲規則來延伸探討不同狀況。首先，設計電腦程式來確定答案的正確性，以及用數學歸納法找尋其中的規則。最後，在解題的過程中發現其解竟然與費氏(Fibonacci)數列有著密切的關係，並且從中發現解題策略，進而試圖推測空間狀況是否仍有如此奇妙的結果。

## 壹、研究動機：

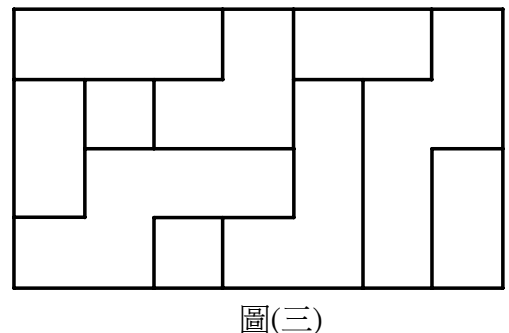
- 一、在高二下課程中，有一道走捷徑問題：  
如圖(一)，有隻蛇走一 $4 \times 7$ 棋盤形的街道，  
試求此蛇由A到B的捷徑走法。



- 二、若將此題的遊戲規則改成：  
此蛇不走街道，改走格子，  
即由左下角C格到右上角D格的捷徑走法。  
如圖(二)



- 三、再將上題的遊戲規則改成：  
蛇由C格出發，但蛇的走法只能往右 $\rightarrow$ ，  
往上 $\uparrow$ ，甚至可以停住。若此蛇已停住，  
將由另一條蛇來走，且不同蛇之間走過之  
格子不可重疊，換句話來說，此棋盤形格  
子將由“一群”蛇來覆蓋，例如右圖就是  
其中一種，試求有多少種能將此棋盤形格  
子覆蓋完畢的方法數。如圖(三)

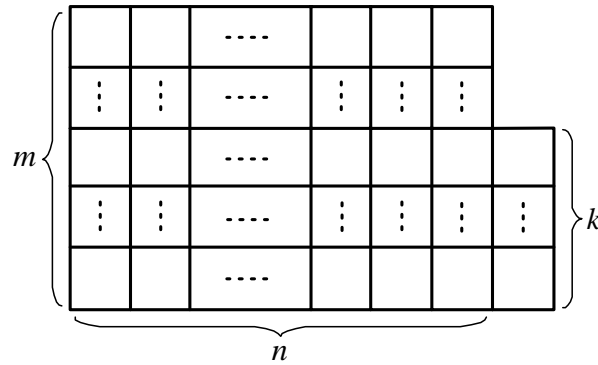


## 貳、研究目的：

- 一、從最簡單的 $1 \times 1$ 、 $1 \times 2$ 、 $1 \times 3$ 等棋盤形格子“蛇填充數”，進而推演至 $3 \times 1$ 、 $3 \times 2$ 、 $3 \times 3$ 等棋盤形格子“蛇填充數”，並利用數學歸納法得到 $1 \times n$ 、 $2 \times n$ 、 $3 \times n$ 等棋盤形格子“蛇填充數”的規律。
- 二、歸納出 $m \times n$ 、 $m \times n + k$ 等棋盤形格子“蛇填充數”的規律，並試著導出與費氏(Fibonacci)數列的關係。



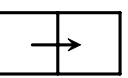


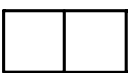
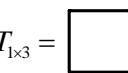
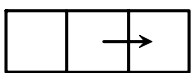
3.  $T_{m \times n+k}$  :  $T_{m \times n+k}$  表示將  $m \times n+k$  棋盤形格子覆蓋完畢之“蛇填充數”，而所謂  $m \times n+k$  棋盤形格子為：



(二)  $T_{1 \times n}$  :

1.  $T_{1 \times 1}$  :   $T_{1 \times 1} = 1$

2.  $T_{1 \times 2}$  :   $T_{1 \times 2} =$     $= 2T_{1 \times 1} = 2$

3.  $T_{1 \times 3}$  :   $T_{1 \times 3} =$     $= 2T_{1 \times 2} = 4$

4.  $T_{1 \times n}$  : 由 1. 2. 3. 推測  $T_{1 \times n} = 2^{n-1}$  ,  $n \geq 1$

證明：當  $n=1$  時，左式  $= T_{1 \times 1} = 1 = 2^{1-1} =$  右式， $\therefore n=1$  成立

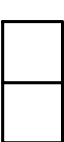


假設  $n=k$  時成立，即  $T_{1 \times k} = 2^{k-1}$

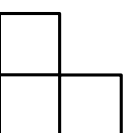
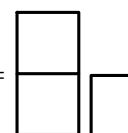
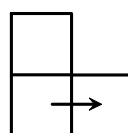
則  $n=k+1$  時，左式  $= T_{1 \times (k+1)}$

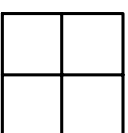
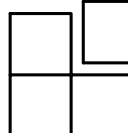
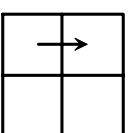
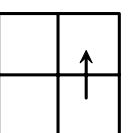
$$\begin{aligned}
 & \underbrace{\left[ \square \quad \dots \quad \square \quad \square \right]}_k T_{1 \times (k+1)} \\
 &= \underbrace{\left[ \square \quad \dots \quad \square \right]}_k \square + \underbrace{\left[ \square \quad \dots \quad \square \quad \rightarrow \right]}_k \\
 &= 2T_{1 \times k} = 2 \times 2^{k-1} = 2^k = \text{右式}
 \end{aligned}$$

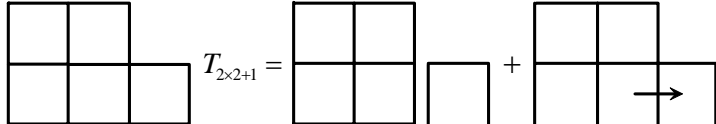
$\therefore$  由數學歸納法得知  $T_{1 \times n} = 2^{n-1}$  ,  $n \geq 1$  成立


(三)  $T_{2 \times n}$  :

1.  $T_{2 \times 1}$  :   $T_{2 \times 1} =$    $+$    $= 2T_{1 \times 1} = 2 \times 1 = 2$

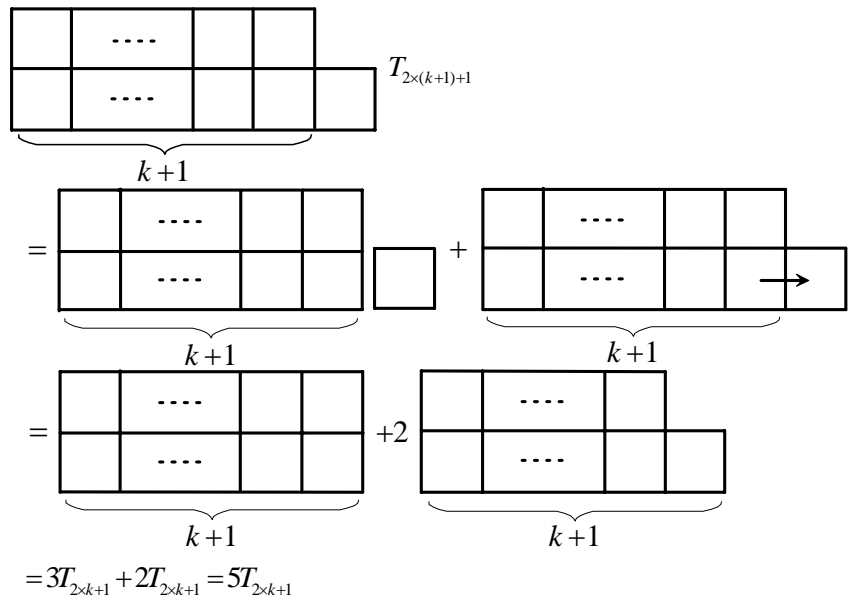
2.  $T_{2 \times 1+1}$  :   $T_{2 \times 1+1} =$     $= T_{2 \times 1} + T_{1 \times 1} = 2 + 1 = 3$

3.  $T_{2 \times 2}$  :   $T_{2 \times 2} =$    $+$    $+$    $= 3T_{2 \times 1+1} = 9$

4.  $T_{2 \times 2+1}$  :   $T_{2 \times 2+1} = T_{2 \times 2} + 2T_{2 \times 1} = 9 + 2 \times 3 = 15 = 5T_{2 \times 1+1}$

5.  $T_{2 \times 3}$  :   $T_{2 \times 3} = 3T_{2 \times 2+1} = 45 = 5T_{2 \times 2}$

6.  $T_{2 \times n+1}$  : 由 2.4. 推測  $T_{2 \times n+1} = 5T_{2 \times (n-1)+1}$  ,  $n \geq 2$   
 證明 : 當  $n = 2$  時 , 左式 =  $T_{2 \times 2+1} = 5T_{2 \times 1+1} =$  右式 ,  $\therefore n = 2$  成立  
 假設  $n = k$  時成立 , 即  $T_{2 \times k+1} = 5T_{2 \times (k-1)+1}$   
 則  $n = k + 1$  時 , 左式 =  $T_{2 \times (k+1)+1}$

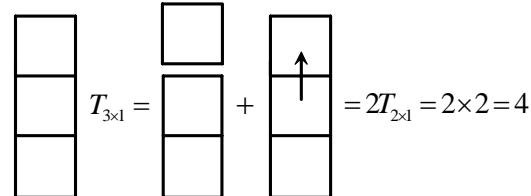


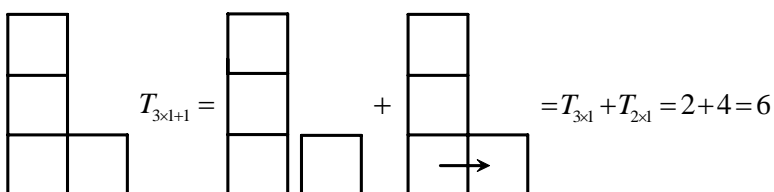
$$= 3T_{2 \times k+1} + 2T_{2 \times k+1} = 5T_{2 \times k+1}$$

$\therefore$  由數學歸納法得知  $T_{2 \times n+1} = 5T_{2 \times (n-1)+1}$  ,  $n \geq 2$  成立

7.  $T_{2 \times n}$  : 由 6. 推測  $T_{2 \times n} = 5T_{2 \times (n-1)}$  ,  $n \geq 2$   
 $\therefore T_{2 \times n} = 3T_{2 \times (n-1)+1} = 3 \times 5T_{2 \times (n-2)+1} = 5 \times 3T_{2 \times (n-2)+1} = 5T_{2 \times (n-1)}$  ,  $\forall n \geq 2$   
 $\therefore T_{2 \times n} = 5T_{2 \times (n-1)}$  ,  $\forall n \geq 2$

(四)  $T_{3 \times n}$  :

1.  $T_{3 \times 1}$  :   $T_{3 \times 1} = 2T_{2 \times 1} = 2 \times 2 = 4$

2.  $T_{3 \times 1+1}$  :   $T_{3 \times 1+1} = T_{3 \times 1} + T_{2 \times 1} = 2 + 4 = 6$

$$3. T_{3 \times 1+2} : \begin{array}{|c|c|} \hline \square \\ \hline \square & \square \\ \hline \square & \square \\ \hline \end{array} T_{3 \times 1+1} = \begin{array}{|c|c|} \hline \square \\ \hline \square & \square \\ \hline \square & \square \\ \hline \end{array} + \begin{array}{|c|c|} \hline \square \\ \hline \square & \square \\ \hline \square & \square \\ \hline \end{array} + \begin{array}{|c|c|} \hline \square \\ \hline \square & \square \\ \hline \square & \square \\ \hline \end{array}$$

$$= T_{3 \times 1+1} + T_{3 \times 1+1} + (T_{1 \times 2} + T_{1 \times 1}) = 6 + 6 + 3 = 15$$

$$4. T_{3 \times 2} :$$

$$\begin{array}{|c|c|} \hline \square & \square \\ \hline \square & \square \\ \hline \square & \square \\ \hline \end{array} T_{3 \times 2} = 3T_{3 \times 1} = 45$$

$$5. T_{3 \times 2+1} :$$

$$\begin{array}{|c|c|c|} \hline \square & \square & \\ \hline \square & \square & \\ \hline \square & \square & \square \\ \hline \end{array} T_{3 \times 2+1} = 3 \begin{array}{|c|c|c|} \hline \square & & \\ \hline \square & \square & \\ \hline \square & \square & \square \\ \hline \end{array}$$

$$= 3 \left[ \begin{array}{|c|c|c|} \hline \square & \square & \square \\ \hline \square & \square & \\ \hline \square & \square & \square \\ \hline \end{array} + \begin{array}{|c|c|c|} \hline \square & \square & \\ \hline \square & \square & \\ \hline \square & \square & \square \\ \hline \end{array} \right] = 3(T_{3 \times 1+2} + 9) = 72$$

$$\begin{array}{|c|c|c|} \hline \square & \square & \square \\ \hline \square & \square & \\ \hline \square & \square & \square \\ \hline \end{array} = \begin{array}{|c|c|} \hline \square & \square \\ \hline \square & \square \\ \hline \square & \square \\ \hline \end{array} + \begin{array}{|c|c|} \hline \square & \square \\ \hline \square & \square \\ \hline \square & \square \\ \hline \end{array} = 6 + 3 = 9$$

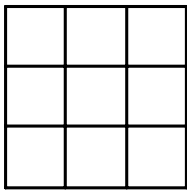
$$6. T_{3 \times 2+2} :$$

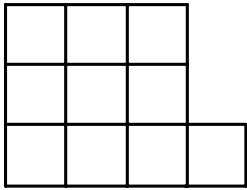
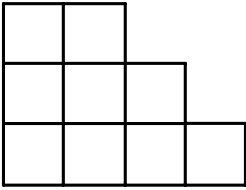
$$\begin{array}{|c|c|c|} \hline \square & \square & \\ \hline \square & \square & \square \\ \hline \square & \square & \square \\ \hline \end{array} T_{3 \times 2+2}$$

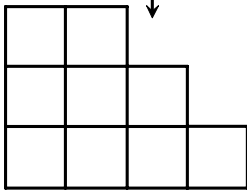
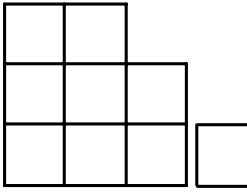
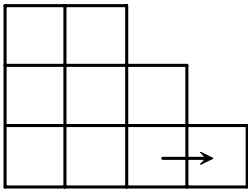
$$= \begin{array}{|c|c|c|} \hline \square & \square & \square \\ \hline \square & \square & \square \\ \hline \square & \square & \square \\ \hline \end{array} + \begin{array}{|c|c|c|} \hline \square & \square & \square \\ \hline \square & \square & \square \\ \hline \square & \square & \square \\ \hline \end{array} + \begin{array}{|c|c|c|} \hline \square & \square & \square \\ \hline \square & \square & \square \\ \hline \square & \square & \square \\ \hline \end{array}$$

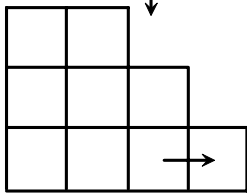
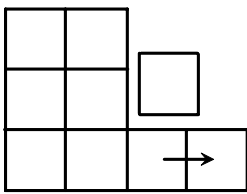
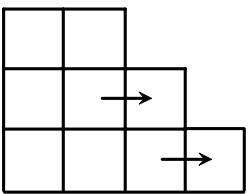
$$= \begin{array}{|c|c|c|} \hline \square & \square & \\ \hline \square & \square & \square \\ \hline \square & \square & \square \\ \hline \end{array} + \begin{array}{|c|c|c|} \hline \square & \square & \\ \hline \square & \square & \square \\ \hline \square & \square & \square \\ \hline \end{array} + 2 \begin{array}{|c|c|c|} \hline \square & \square & \\ \hline \square & \square & \square \\ \hline \square & \square & \square \\ \hline \end{array}$$

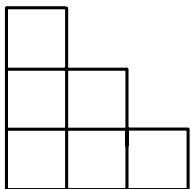
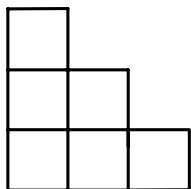
$$= T_{3 \times 2+1} + T_{3 \times 2+1} + 2 \times \frac{1}{3} T_{3 \times 2+1} = \frac{8}{3} T_{3 \times 2+1} = \frac{8}{3} \times 72 = 192$$

7.  $T_{3 \times 3}$  :   $T_{3 \times 3} = 3T_{3 \times 2+2} = 576$

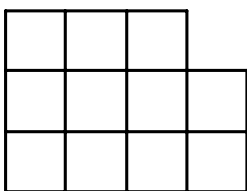
8.  $T_{3 \times 3+1}$  :   $T_{3 \times 3+1} = 3$    $= 3 \times 312 = 936$

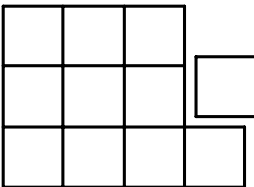
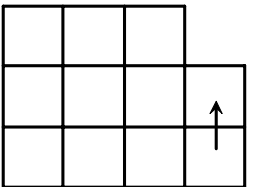
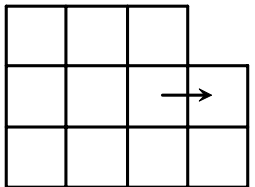
  $=$    $+$    $= 192 + 120 = 312$

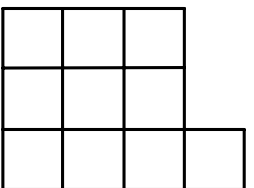
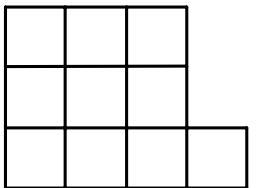
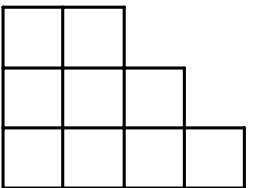
  $=$    $+$  

$= 3$    $+ 2$    $= 5 \times 24 = 120$

發現  $T_{3 \times 3+1} = 13T_{3 \times 2+1}$

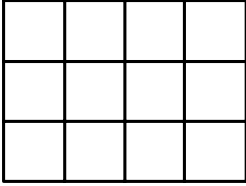
9.  $T_{3 \times 3+2}$  :   $T_{3 \times 3+2}$

$=$    $+$    $+$  

$=$    $+$    $+ 2$  

$= T_{3 \times 3+1} + T_{3 \times 3+1} + 2 \times \frac{1}{3} T_{3 \times 3+1} = \frac{8}{3} T_{3 \times 3+1} = \frac{8}{3} \times 936 = 2496$

發現  $T_{3 \times 3+2} = 13T_{3 \times 2+2}$

10.  $T_{3 \times 4}$  :   $T_{3 \times 4} = 3T_{3 \times 3+2} = 7488$

發現  $T_{3 \times 4} = 13T_{3 \times 3}$

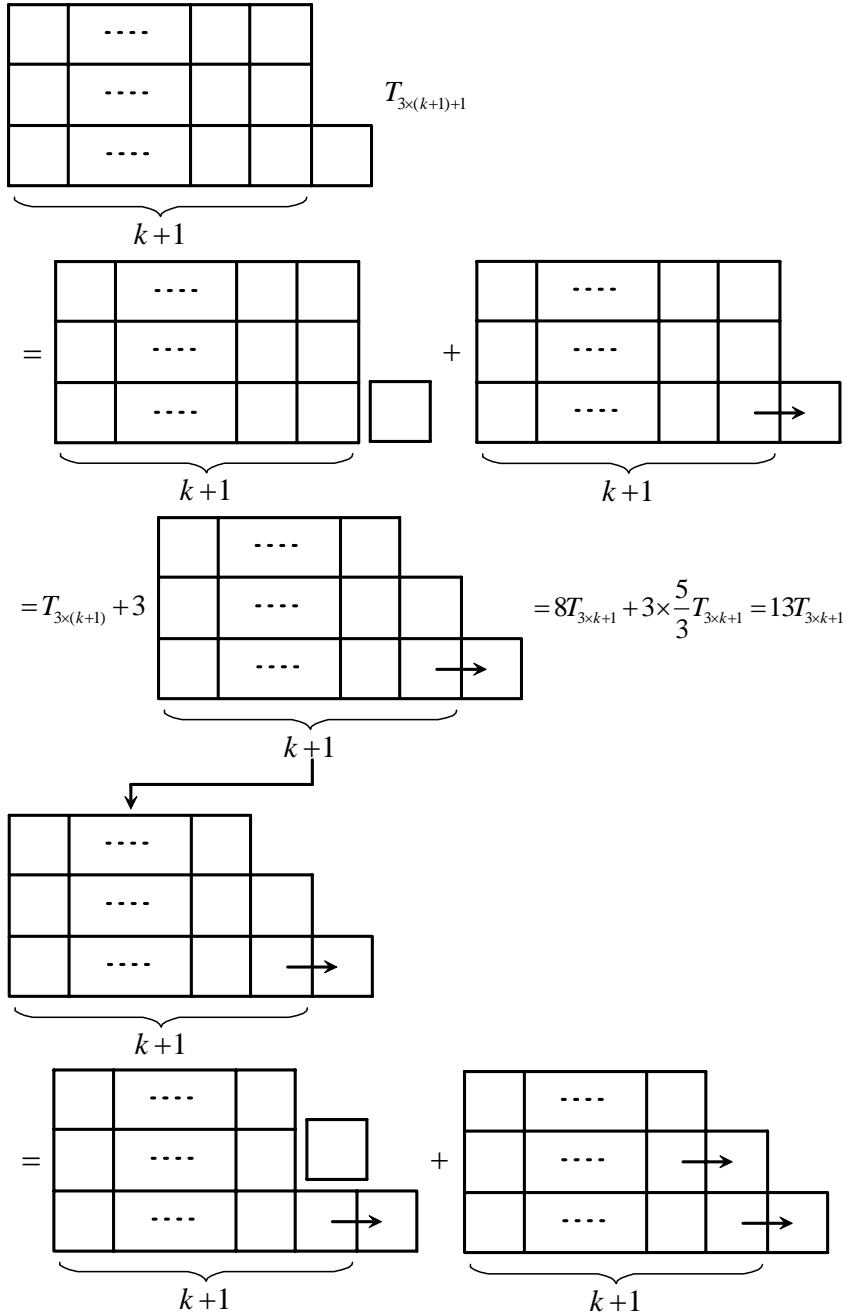
11.  $T_{3 \times n+1}$  :

由 5.8. 推測  $T_{3 \times n+1} = 13T_{3 \times (n-1)+1}$ ,  $n \geq 3$

證明：當  $n=3$  時，左式 =  $T_{3 \times 3+1} = 13T_{3 \times 2+1}$  = 右式， $\therefore n=3$  成立

假設  $n=k$  時成立，即  $T_{3 \times k+1} = 13T_{3 \times (k-1)+1}$

則  $n=k+1$  時，左式 =  $T_{3 \times (k+1)+1}$



$$\begin{aligned}
&= \underbrace{\begin{array}{|c|c|c|} \hline & \dots & \\ \hline & \dots & \\ \hline & \dots & \\ \hline \end{array}}_{k+1} + 2 \underbrace{\begin{array}{|c|c|c|} \hline & \dots & \\ \hline & \dots & \\ \hline & \dots & \\ \hline \end{array}}_{k+1} \\
&= (1+2 \times \frac{1}{3})T_{3 \times k+1} = \frac{5}{3}T_{3 \times k+1}
\end{aligned}$$

$\therefore$ 由數學歸納法得知 $T_{3 \times n+1} = 13T_{3 \times (n-1)+1}$ ， $n \geq 3$ 成立

12.  $T_{3 \times n+2}$  :

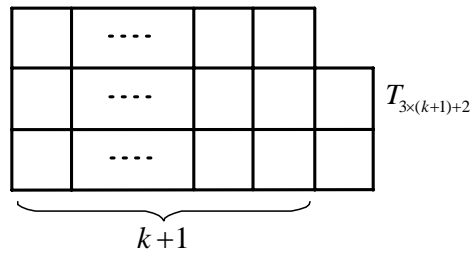
由 6.9.推測 $T_{3 \times n+2} = 13T_{3 \times (n-1)+2}$ ， $n \geq 3$

證明：當 $n = 3$ 時，左式 $= T_{3 \times 3+2} = 13T_{3 \times 2+2} =$ 右式， $\therefore n = 3$ 成立

假設 $n = k$ 時成立，即 $T_{3 \times k+2} = 13T_{3 \times (k-1)+2}$

則 $n = k + 1$ 時，

左式 $= T_{3 \times (k+1)+2}$



$$= \frac{8}{3}T_{3 \times (k+1)+1} = \frac{8}{3} \times 13T_{3 \times k+1} = 13 \times \frac{8}{3}T_{3 \times k+1} = 13T_{3 \times k+2}$$

$\therefore$ 由數學歸納法得知 $T_{3 \times n+2} = 13T_{3 \times (n-1)+2}$ ， $n \geq 3$ 成立

13.  $T_{3 \times n}$  :

$\therefore T_{3 \times (n+1)} = 3T_{3 \times n+2} = 3 \times 13T_{3 \times (n-1)+2} = 13 \times 3T_{3 \times (n-1)+2} = 13T_{3 \times n}$ ， $\forall n \geq 3$

$\therefore T_{3 \times (n+1)} = 13T_{3 \times n}$ ， $\forall n \geq 3$

(五)  $T_{4 \times n}$  :

1.  $T_{4 \times 1}$  :

$$T_{4 \times 1} = \begin{array}{|c|} \hline \\ \hline \\ \hline \\ \hline \end{array} = \begin{array}{|c|} \hline \\ \hline \\ \hline \\ \hline \end{array} + \begin{array}{|c|} \hline \uparrow \\ \hline \\ \hline \\ \hline \end{array} = 2T_{3 \times 1} = 2 \times 4 = 8$$

2.  $T_{4 \times 1+1}$  :

$$T_{4 \times 1+1} = \begin{array}{|c|c|} \hline \\ \hline \\ \hline \\ \hline \end{array} = \begin{array}{|c|c|} \hline \\ \hline \\ \hline \\ \hline \end{array} + \begin{array}{|c|c|} \hline \uparrow \\ \hline \\ \hline \\ \hline \end{array} = 2T_{3 \times 1+1} = 12$$

3.  $T_{4 \times 1+2}$  :

$$T_{4 \times 1+2} = T_{3 \times 1+2} + T_{3 \times 1+2} = 2T_{3 \times 1+2} = 30$$

4.  $T_{4 \times 1+3}$  :

$$T_{4 \times 1+3} = T_{3 \times 2} + 2T_{3 \times 1+2} = 45 + 2 \times 15 = 75$$

5.  $T_{4 \times 2}$  :

$$T_{4 \times 2} = 3T_{4 \times 1+3} = 3 \times 75 = 225$$

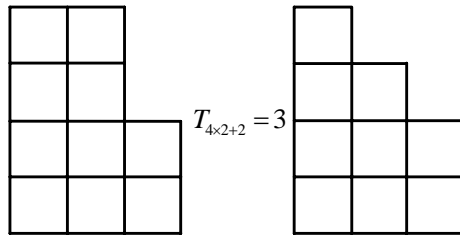
6.  $T_{4 \times 2+1}$  :

$$T_{4 \times 2+1} = 3 \left[ T_{4 \times 2+1} + T_{4 \times 2+1} + T_{4 \times 2+1} \right]$$

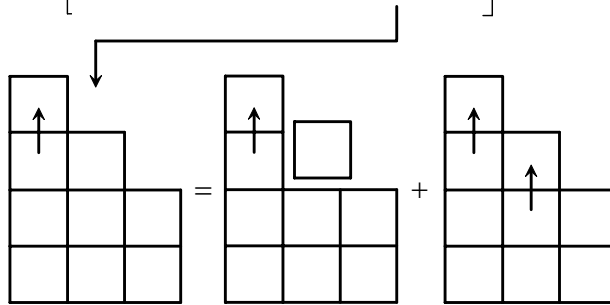
$$= 3 \left[ T_{4 \times 2+1} + 2T_{4 \times 2+1} \right]$$

$$= 3(72 + 2 \times 24) = 360$$

7.  $T_{4 \times 2+2}$  :

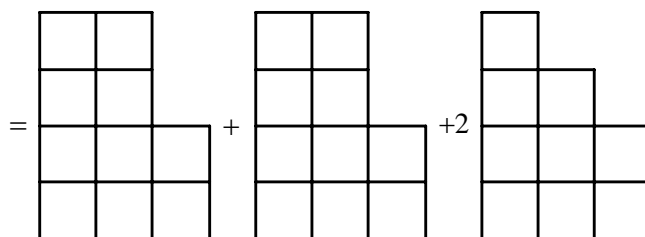
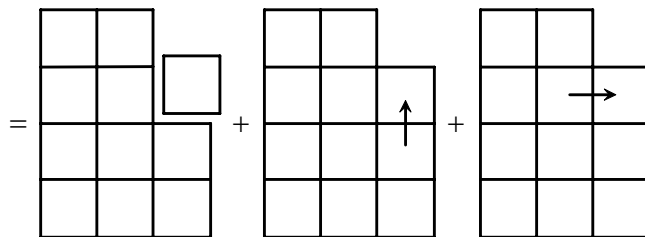
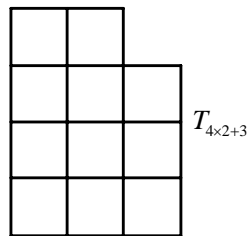


$$= 3 \left[ \begin{array}{c} \begin{array}{|c|c|c|} \hline \square & & \\ \hline \square & \square & \\ \hline \square & \square & \square \\ \hline \square & \square & \square \\ \hline \end{array} + \begin{array}{|c|c|c|} \hline \square & & \\ \hline \uparrow & \square & \\ \hline \square & \square & \square \\ \hline \square & \square & \square \\ \hline \end{array} \end{array} \right] = 3(192+120) = 936$$



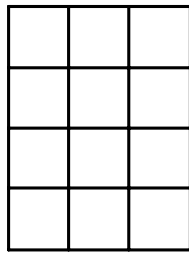
$$= 3 \begin{array}{|c|c|c|} \hline \square & & \\ \hline \square & \square & \\ \hline \square & \square & \square \\ \hline \square & \square & \square \\ \hline \end{array} + 2 \begin{array}{|c|c|c|} \hline \square & & \\ \hline \square & \square & \\ \hline \square & \square & \square \\ \hline \square & \square & \square \\ \hline \end{array} = 72 + 2 \times 24 = 120$$

8.  $T_{4 \times 2+3}$  :



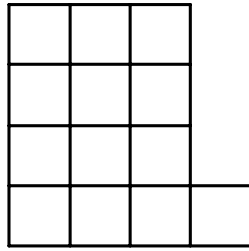
$$= T_{4 \times 2+2} + T_{4 \times 2+2} + 2 \times \frac{1}{3} T_{4 \times 2+2} = \frac{8}{3} T_{4 \times 2+2} = \frac{8}{3} \times 936 = 2496$$

9.  $T_{4 \times 3}$  :

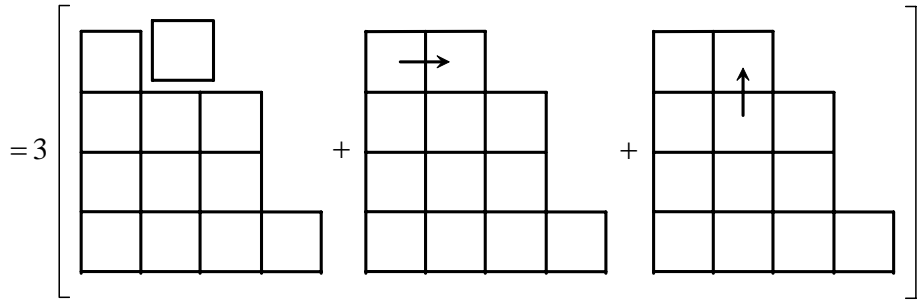
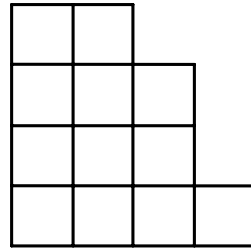


$$T_{4 \times 3} = 3T_{4 \times 2+3} = 3 \times 2496 = 7488$$

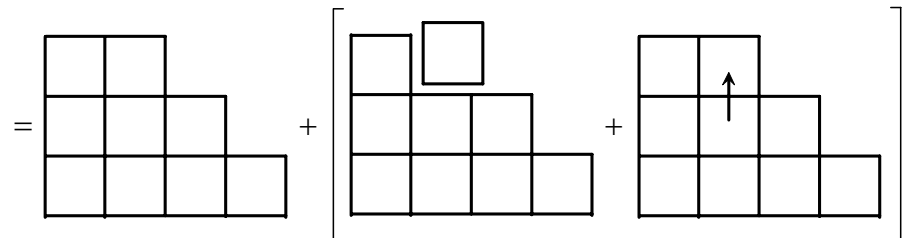
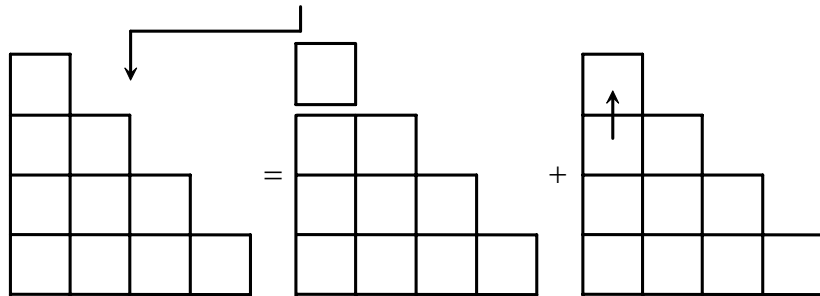
10.  $T_{4 \times 3+1}$  :



$$T_{4 \times 3+1} = 3$$

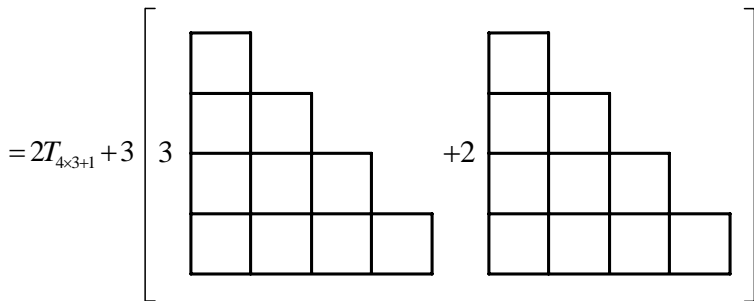
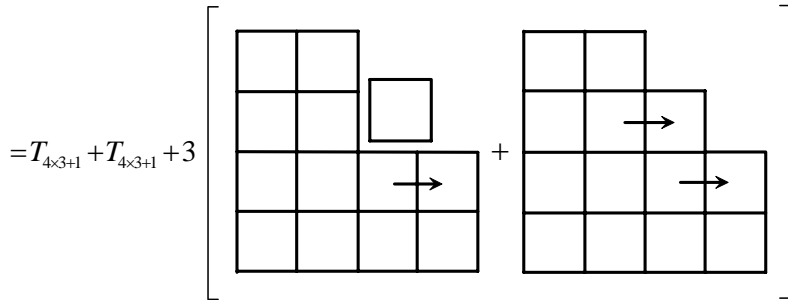
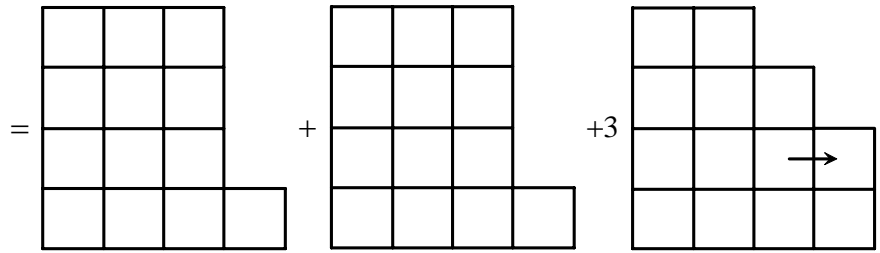
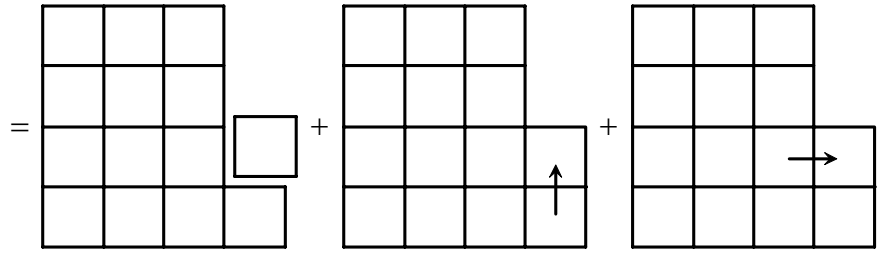
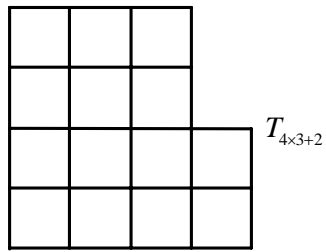


$$= 3(3+3+2) = 3 \times 8 \times 504 = 12096$$



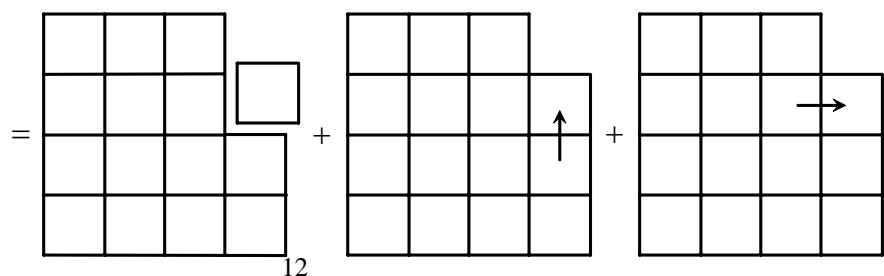
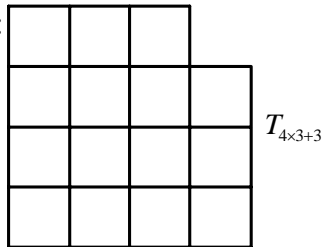
$$= \frac{1}{3} T_{4 \times 2+2} + \left[ 5 \begin{array}{ccc} \square & & \\ \square & \square & \\ \square & \square & \square \end{array} + 3 \begin{array}{ccc} \square & & \\ \square & \square & \\ \square & \square & \square \end{array} \right] = \frac{1}{3} \times 936 + 8 \times 24 = 504$$

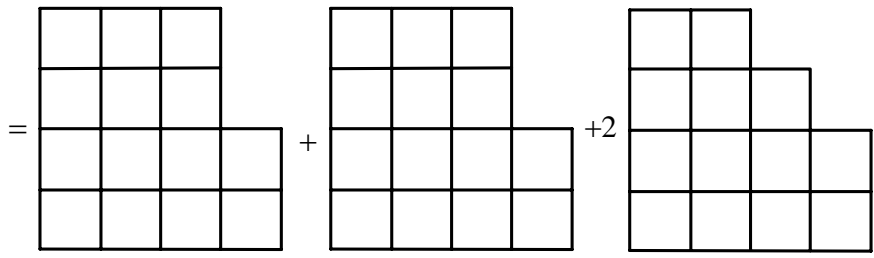
11.  $T_{4 \times 3+2}$  :



$$= 2 \times 12096 + 3 \times 5 \times 504 = 31752$$

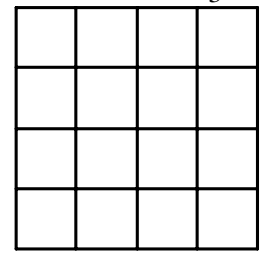
12.  $T_{4 \times 3+3}$  :





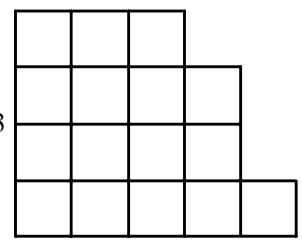
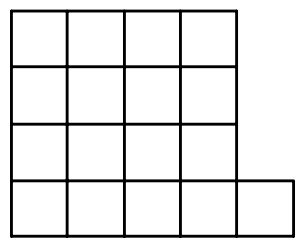
$$= T_{4 \times 3+2} + T_{4 \times 3+2} + 2 \times \frac{1}{3} T_{4 \times 3+2} = \frac{8}{3} T_{4 \times 3+2} = 84672$$

13.  $T_{4 \times 4}$  :

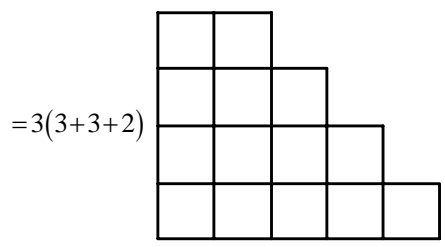
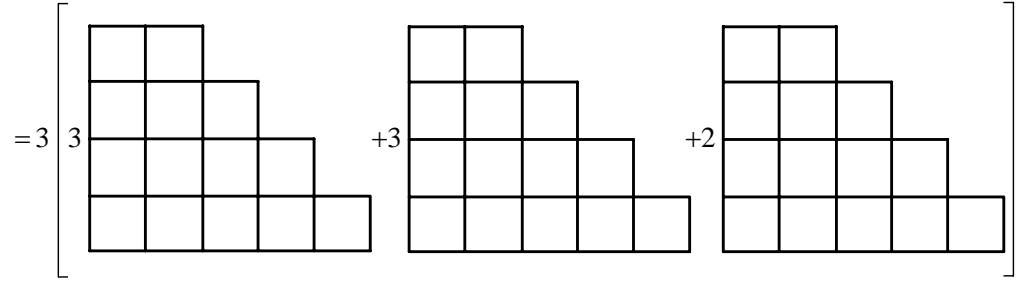
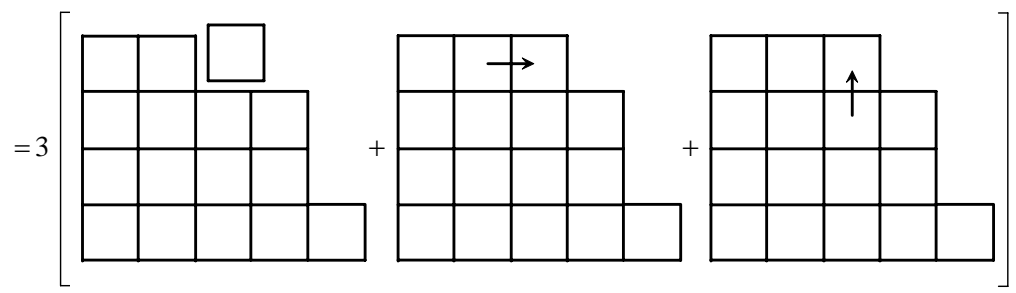


$$T_{4 \times 4} = 3T_{4 \times 3+3} = 3 \times 84672 = 254016$$

14.  $T_{4 \times 4+1}$  :

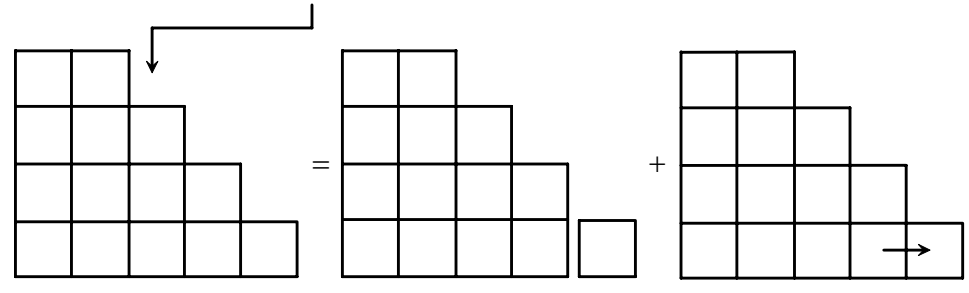


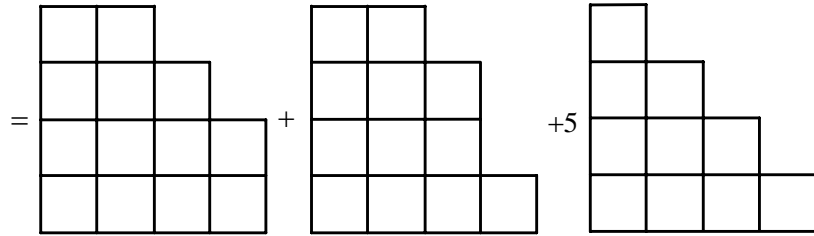
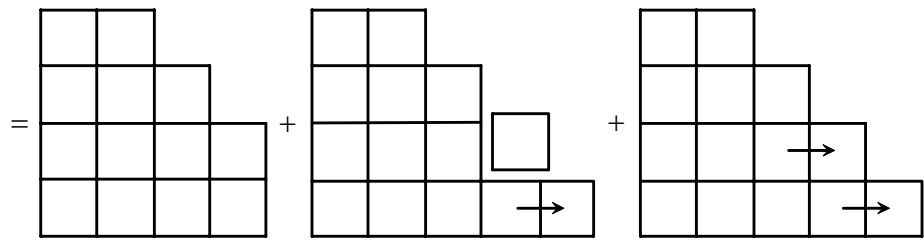
$$T_{4 \times 4+1} = 3$$



$$= 3(3+3+2)$$

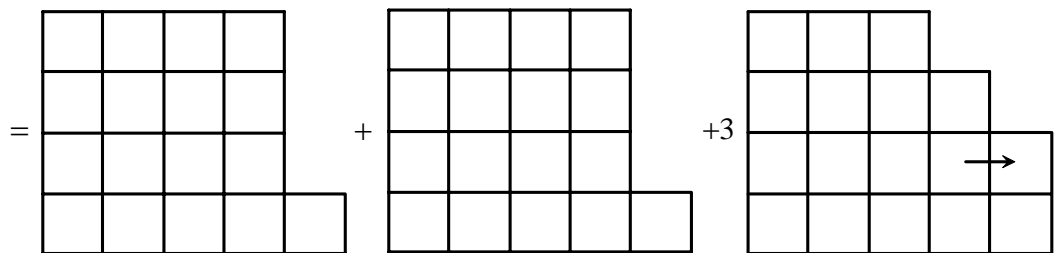
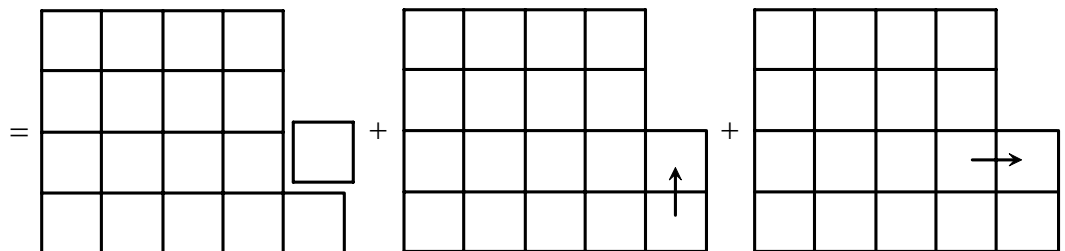
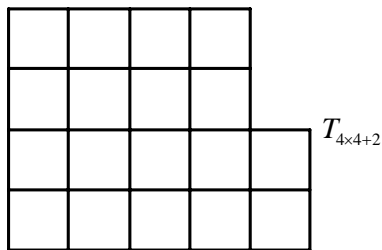
$$= 3 \times 8 \times 17136 = 411264$$





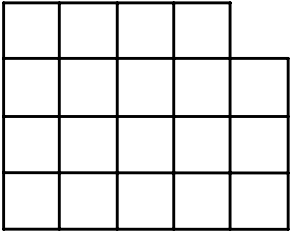
$$= \frac{1}{3}T_{4 \times 3+2} + \frac{1}{3}T_{4 \times 3+1} + 5 \times 504 = \frac{1}{3} \times 31752 + \frac{1}{3} \times 12096 + 2520 = 17136$$

15.  $T_{4 \times 4+2}$  :

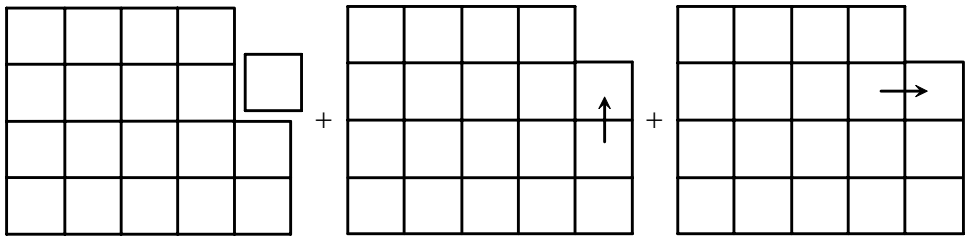


$$= T_{4 \times 4+1} + T_{4 \times 4+1} + 3 \times 5 \quad = 2 \times 411264 + 3 \times 5 \times 17136 = 1079568$$

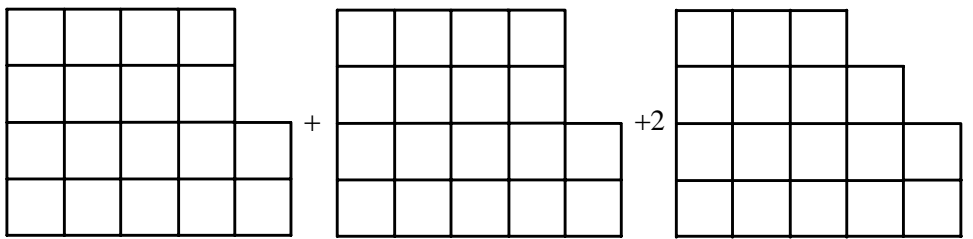
16.  $T_{4 \times 4+3}$  :



=

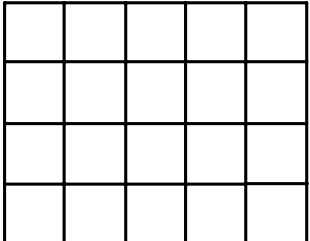


=



$$= T_{4 \times 4+2} + T_{4 \times 4+2} + 2 \times \frac{1}{3} T_{4 \times 4+2} = \frac{8}{3} T_{4 \times 4+2} = \frac{8}{3} \times 1079568 = 2878848$$

17.  $T_{4 \times 5}$  :



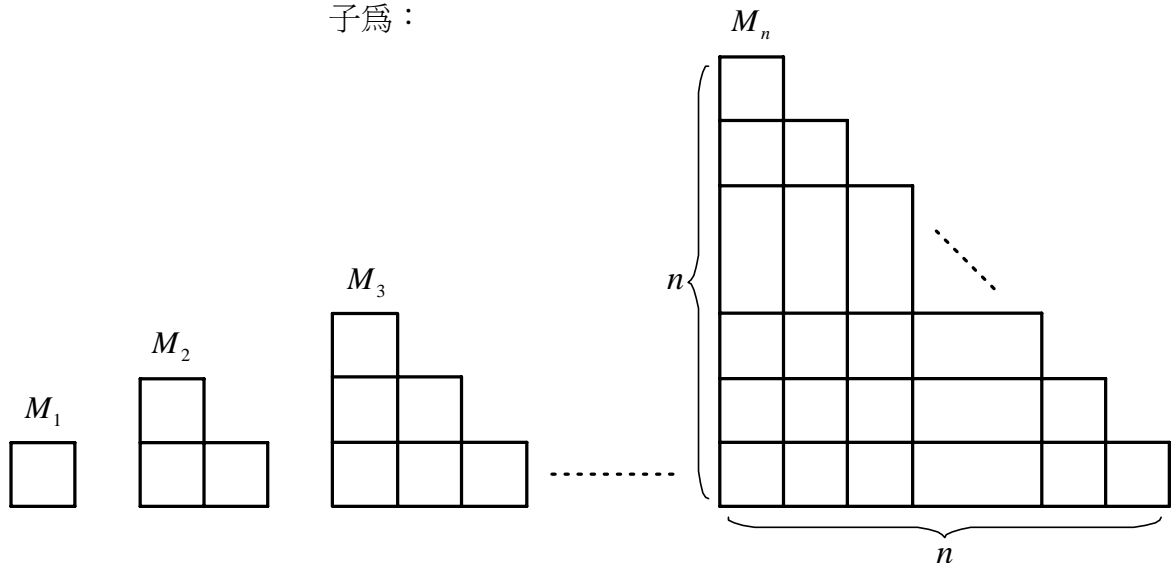
$T_{4 \times 5} = 3T_{4 \times 4+3} = 3 \times 2878848 = 8636544$

(六) 由(二)~(五)的演繹過程中，逐步發現其中的規律，以下就是研究過程：

1.  $M_n$  :

(1) 名詞解釋：

(a)  $M_n$ ：將  $n$  層階梯格子覆蓋完畢之“蛇填充數”，而所謂  $n$  層階梯格子為：



(b)  $a_{xy}$  : 由左至右數來第  $x$  行，由下至上第  $y$  列之格子

(c) 費氏 (Fibonacci) 數列  $\langle F_n \rangle$  :  $1, 1, 2, 3, 5, \dots$  , 它的遞迴關係為:  $F_1 = F_2 = 1, F_{n+2} = F_{n+1} + F_n, n \geq 1$

(2) 證明:  $M_n = F_2 \times F_4 \times F_6 \times \dots \times F_{2n}, n \geq 1$

證明:  $M_1$  :

$$\begin{array}{|c|} \hline \square \\ \hline \end{array} M_1 = 1 = F_2$$

$$M_2 : \begin{array}{|c|c|} \hline \square & \\ \hline \square & \square \\ \hline \end{array} = \begin{array}{|c|} \hline \square \\ \hline \end{array} + \begin{array}{|c|c|} \hline \square & \\ \hline a_{11} & a_{21} \\ \hline \end{array} + \begin{array}{|c|c|} \hline \square & \\ \hline a_{11} & a_{21} \\ \hline \end{array}$$

$M_2$  可由  $M_1$  加上  $a_{12}$  與  $a_{21}$  而得:

狀況甲: 當  $a_{11}$  與  $a_{21}$  不相連時,  $a_{12}$  可與  $a_{11}$  相連或不連

$\therefore$  有  $(1+1) = 2 = F_3$  種情形

狀況乙: 當  $a_{11}$  與  $a_{21}$  相連時,  $a_{12}$  只可與  $a_{11}$  不相連

$\therefore$  有  $1 = F_2$  種情形

又每一種情形之蛇填充數為  $M_1$

$$\therefore M_2 = (F_2 + F_3)M_1 = F_4 M_1 = F_2 \times F_4$$

$$M_3 : \begin{array}{|c|c|c|} \hline \square & & \\ \hline \square & \square & \\ \hline \square & \square & \square \\ \hline \end{array} = \begin{array}{|c|} \hline a_{13} \\ \hline \end{array} + \begin{array}{|c|c|} \hline a_{12} & a_{22} \\ \hline \square & a_{21} \\ \hline \end{array} + \begin{array}{|c|c|} \hline a_{13} & \\ \hline a_{12} & a_{22} \\ \hline \square & a_{21} \\ \hline \end{array} + \begin{array}{|c|c|c|} \hline \square & & \\ \hline \square & \square & \\ \hline \square & \square & \square \\ \hline \end{array}$$

$M_3$  可由  $M_2$  加上  $a_{13}$  與  $a_{22}$  與  $a_{31}$  而得:

狀況甲: 當  $a_{21}$  與  $a_{31}$  不相連時:

(a)  $a_{22}$  與  $a_{21}$  相連, 即  $a_{12}$  與  $a_{22}$  不相連即可視為與  $M_2$  之狀況甲相同

$\therefore$  有  $F_3$  種情形

(b)  $a_{22}$  與  $a_{21}$  不相連, 則可視為由  $a_{12}$  加上  $a_{13}$  與  $a_{22}$ , 亦即與由  $M_1$  推得  $M_2$  之狀況相同

$\therefore$  有  $F_2 + F_3$  兩種情形

狀況乙: 當  $a_{21}$  與  $a_{31}$  相連時, 即  $a_{22}$  與  $a_{21}$  不相連, 則可視為由  $a_{12}$  加上  $a_{13}$  與  $a_{22}$ , 亦即與  $M_1$  推得  $M_2$  之狀況相同

$\therefore$  有  $F_2 + F_3$  兩種情形

又每種情形之蛇填充數為  $M_2$

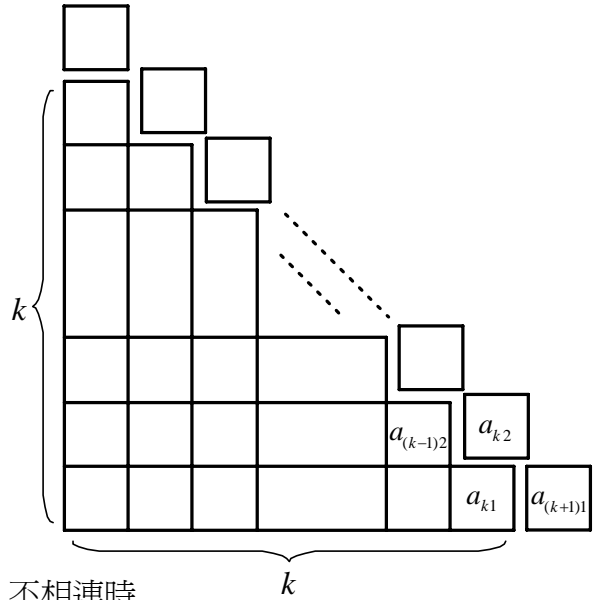
$$\begin{aligned} \therefore M_3 &= [F_3 + (F_2 + F_3)]M_2 + (F_2 + F_3)M_2 \\ &= (F_3 + F_4)M_2 + F_4 M_2 = (F_5 + F_4)M_2 = F_2 \times F_4 \times F_6 \end{aligned}$$

用數學歸納法證明其一般式 ( $M_n = F_2 \times F_4 \times F_6 \times \dots \times F_{2n}, n \geq 1$ )

當  $n=1$  時,  $M_1 = F_2$

假設  $\forall n \leq k$  成立, 即

$$M_k = F_{2k-1} M_{k-1} + F_{2k-2} M_{k-1} = F_{2k} M_{k-1} = F_2 \times F_4 \times F_6 \times \dots \times F_{2k}$$



則當  $n = k + 1$  時

左式 =  $M_{k+1}$

狀況甲：當  $a_{(k+1)1}$  與  $a_{k1}$  不相連時

(a)  $a_{k2}$  與  $a_{k1}$  相連，即  $a_{(k-1)2}$  與  $a_{k2}$  不相連，可視為與  $M_{k-1}$  之狀況甲相同

$\therefore$  有  $F_{2k-1}$  種情形

(b)  $a_{k2}$  與  $a_{k1}$  不相連，則與由  $M_{k-1}$  推得  $M_k$  之狀況相同

$\therefore$  有  $(F_{2k-1} + F_{2k-2})$  種情形

狀況乙：當  $a_{(k+1)1}$  與  $a_{k1}$  相連時，即  $a_{k2}$  與  $a_{k1}$  不相連，則與由  $M_{k-1}$  推得  $M_k$  之狀況相同

$\therefore$  有  $(F_{2k-1} + F_{2k-2})$  種情形

又每種情形之蛇填充數為  $M_k$

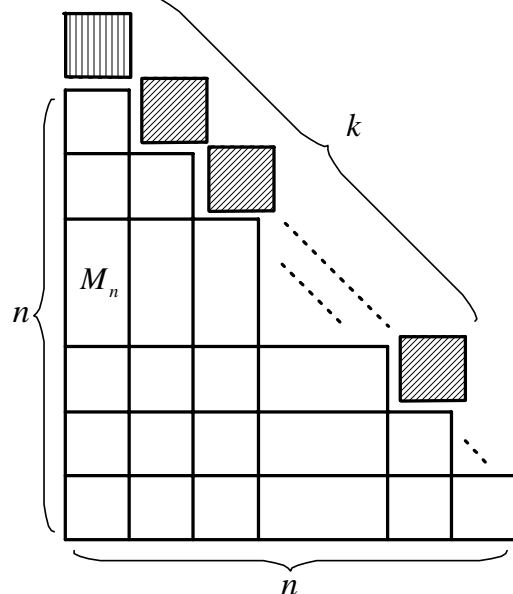
$$\begin{aligned} M_{k+1} &= [F_{2k-1} + (F_{2k-2} + F_{2k-1})]M_k + (F_{2k-2} + F_{2k-1})M_k \\ &= (F_{2k-1} + F_{2k})M_k + F_{2k}M_k = (F_{2k+1} + F_{2k})M_k \\ &= F_2 \times F_4 \times F_6 \times \cdots \times F_{2k} \times F_{2k+2} \end{aligned}$$

$\therefore$  由數學歸納法得證  $M_n = F_2 \times F_4 \times F_6 \times \cdots \times F_{2n}$

2.  $A_k$  :

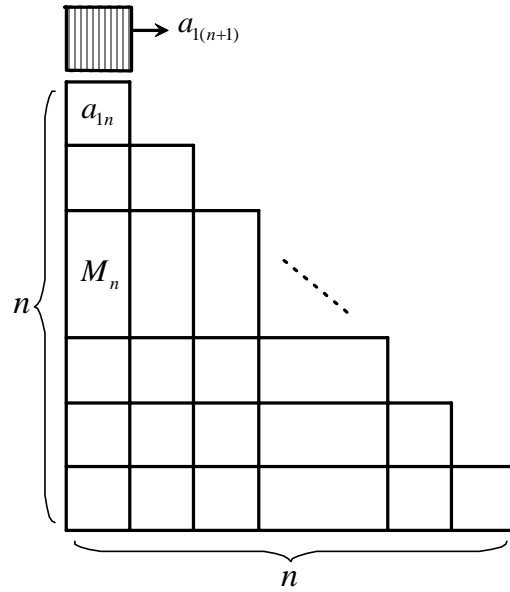
(1) 名詞解釋：

$A_k$  : 在  $n$  層階梯格子上從第一列上方往右下增添  $k$  個方格數(如圖)，且將此棋盤形格子覆蓋完畢之“蛇填充數”。



(2) 證明： $A_k = F_{2k+1}M_n$ ， $n \geq k \geq 1$

證明：當  $k = 1$  時，左式 =  $A_1$



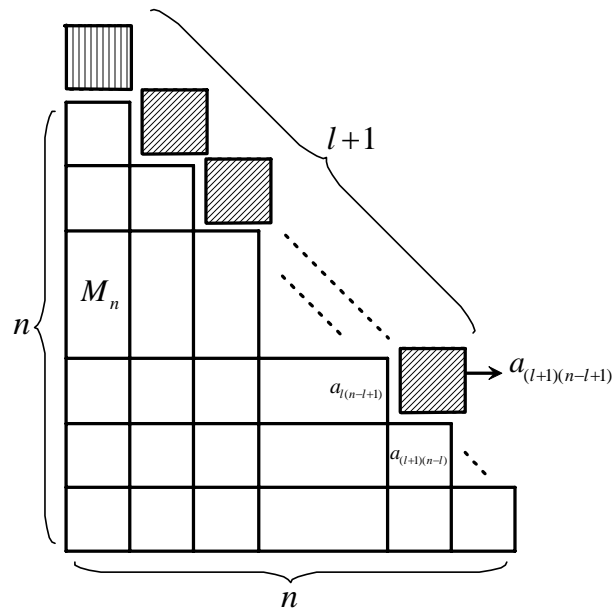
$$A_1 = 2M_n = F_3M_n$$

$\therefore k = 1$  時成立

假設  $k = l$  時成立，即  $A_l = F_{2l+1}M_n$

則當  $k = l + 1$  時

左式 =  $A_{l+1}$



狀況甲： $a_{(l+1)(n-l)}$  與  $a_{(l+1)(n-l+1)}$  相連時，情況與  $A_l$  同，有  $F_{2l+1}$  種情形

狀況乙：當  $a_{(l+1)(n-l+1)}$  完全不與其他相連時，則情況與  $A_l$  同，有  $F_{2l+1}$  種情形

狀況丙：當  $a_{(l+1)(n-l+1)}$  與  $a_{l(n-l+1)}$  相連時，則為  $A_l$  扣掉  $a_{l(n-l+1)}$  與  $a_{l(n-l+2)}$  相連，有  $(F_{2l+1} - F_{2l-1})$  種情形

又每種情形之蛇填充數為  $M_n$

$$\begin{aligned}
A_{l+1} &= (3F_{2l+1} - F_{2l-1})M_n = (F_{2l+3} - F_{2l+3} + 3F_{2l+1} - F_{2l-1})M_n \\
&= [F_{2l+3} - (F_{2l+2} + F_{2l+1}) + 3F_{2l+1} - F_{2l-1}]M_n \\
&= [F_{2l+3} - (F_{2l+1} + F_{2l}) + 2F_{2l+1} - F_{2l-1}]M_n \\
&= [F_{2l+3} - (F_{2l-1} + F_{2l}) + F_{2l+1}]M_n = F_{2l+3}M_n
\end{aligned}$$

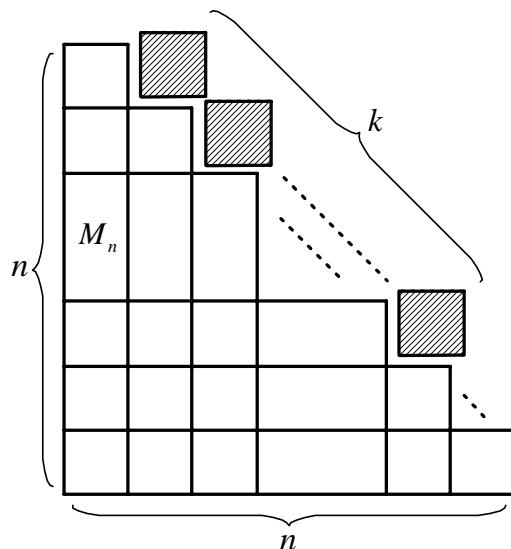
∴由數學歸納法得證  $A_k = F_{2k+1}M_n$  ,  $n \geq k \geq 1$

- (3) 令  $a_k$  表示在  $n$  層階梯格子斜向增加  $k$  個格子 ( $n \geq k \geq 1$ ) [其中若  $k=1$  , 表示此格子不可與左方的格子相連(或左方沒有格子); 若  $k \geq 2$  , 表示左上方第一個格子不可與左方的格子相連(或左方沒有格子), 其他  $k-1$  個格子與其他格子的關係則無限制。], 將其完全覆蓋完畢之蛇填充數與原來  $n$  層階梯格子的蛇填充數之倍數關係。由(2)可得  $a_k = F_{2k+1}$ 。

3.  $B_k$  :

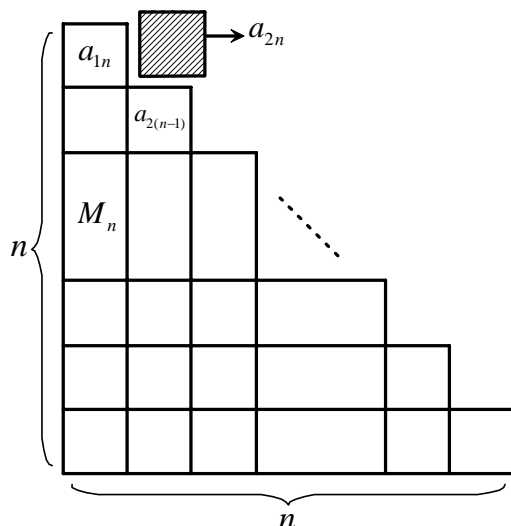
- (1) 名詞解釋 :

$B_k$  : 在  $n$  層階梯格子上從第二列上方往右下增添  $k$  個方格數(如圖), 且將此棋盤形格子覆蓋完畢之“蛇填充數”。



- (2) 證明 :  $B_k = F_{2k+2}M_n$  ,  $n > k \geq 1$

證明 : 當  $k=1$  時, 左式 =  $B_1$



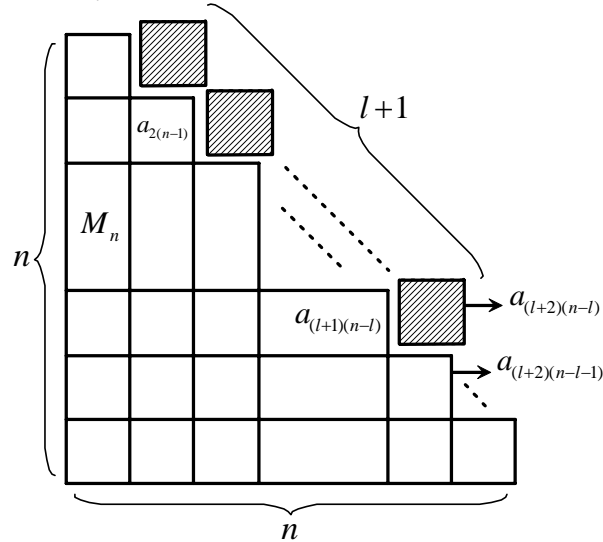
$$B_1 = 3M_n = F_4M_n$$

∴  $k = 1$  時成立

假設  $k = l$  時成立，即  $B_l = F_{2l+2}M_n$

則當  $k = l + 1$  時

左式 =  $B_{l+1}$



狀況甲：  $a_{(l+2)(n-l-1)}$  與  $a_{(l+2)(n-l)}$  相連時，情況與  $B_l$  同，有  $F_{2l+2}$  種情形

狀況乙：當  $a_{(l+2)(n-l)}$  完全不與其他相連時，情況與  $B_l$  同，有  $F_{2l+2}$  種情形

狀況丙：當  $a_{(l+2)(n-l)}$  與  $a_{(l+1)(n-l)}$  相連時，則為  $B_l$  扣掉  $a_{(l+1)(n-l)}$  與  $a_{(l+1)(n-l+1)}$  相連，有  $(F_{2l+2} - F_{2l})$  種情形

又每種情形之蛇填充數為  $M_n$

$$\therefore B_{l+1} = (3F_{2l+2} - F_{2l})M_n = F_{2l+4}M_n$$

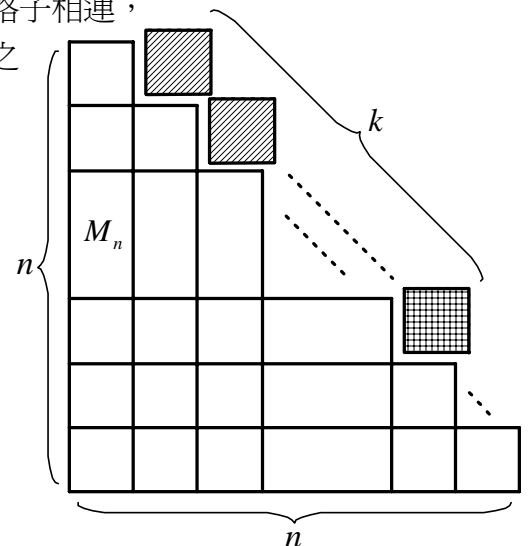
∴ 由數學歸納法得證  $B_k = F_{2k+2}M_n$ ， $n > k \geq 1$

- (3) 令  $b_k$  表示在  $n$  層階梯格子斜向增加  $k$  個格子 ( $n > k \geq 1$ ) [其中此  $k$  個格子與其他格子的關係並無限制]，將其完全覆蓋完畢之蛇填充數與原來  $n$  層階梯格子的蛇填充數之倍數關係。由(2)可得  $b_k = F_{2k+2}$ 。

#### 4. $C_k$ :

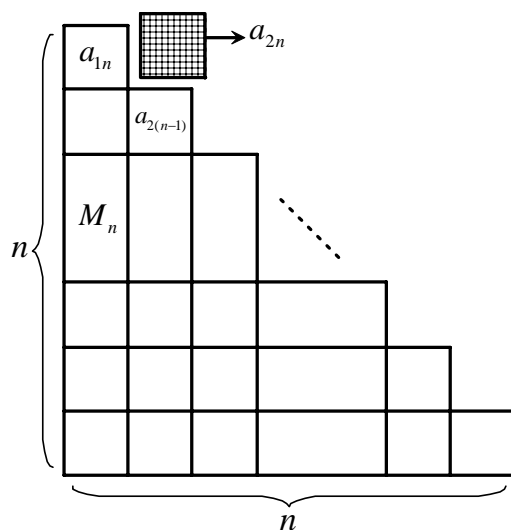
- (1) 名詞解釋：

$C_k$  : 在  $n$  層階梯格子上從第二列上方往右下增添  $k$  個方格數，但於右下最後增添的方格子不可與下方格子相連，如此將此棋盤形格子覆蓋完畢之“蛇填充數”稱之為  $C_k$ 。



(2)證明： $C_k = F_{2k+1}M_n$ ， $n \geq k \geq 1$

證明：當  $k=1$  時，左式 =  $C_1$



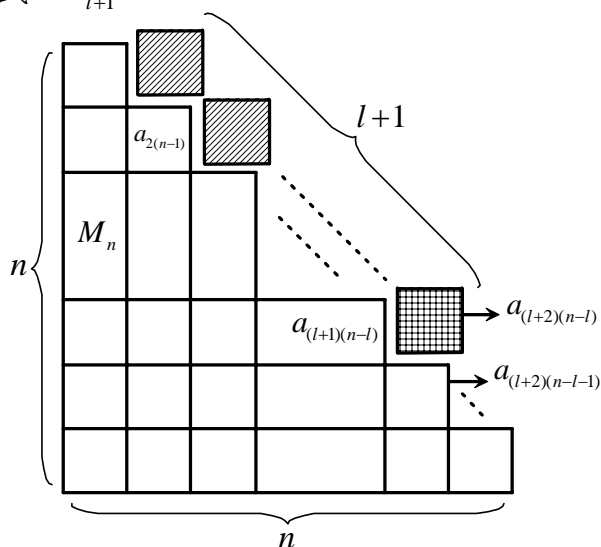
$$C_1 = 2M_n = F_3M_n$$

$\therefore k=1$  時成立

假設  $k=l$  時成立，即  $C_l = F_{2l+1}M_n$

則當  $k=l+1$  時

左式 =  $C_{l+1}$



狀況甲：當  $a_{(l+2)(n-l)}$  完全不與其他相連時，情況與  $B_l$  同，有  $F_{2l+2}$  種情形

狀況乙：當  $a_{(l+2)(n-l)}$  與  $a_{(l+1)(n-l)}$  相連時，情況與  $C_l$  同，有  $F_{2l+1}$  種情形  
又每種情形之蛇填充數為  $M_n$

$$\therefore C_{l+1} = (F_{2l+2} + F_{2l+1})M_n = F_{2l+3}M_n$$

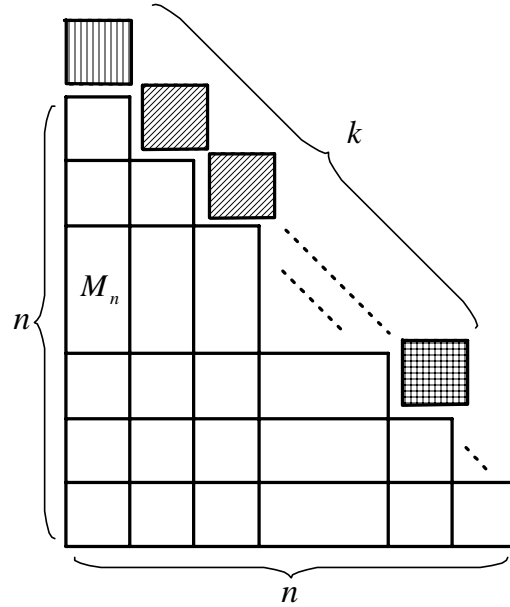
$\therefore$  由數學歸納法得證  $C_k = F_{2k+1}M_n$ ， $n \geq k \geq 1$

(3) 令  $c_k$  表示在  $n$  層階梯格子斜向增加  $k$  個格子 ( $n \geq k \geq 1$ ) [其中若  $k=1$ ，表示此格子不可與下方的格子相連(或下方沒有格子)；若  $k \geq 2$ ，表示右下方最後一個格子不可與下方的格子相連(或下方沒有格子)，其他  $k-1$  個格子與其他格子的關係則無限制。]，將其完全覆蓋完畢之蛇填充數與原來  $n$  層階梯格子的蛇填充數之倍數關係。由(2)可得  $c_k = F_{2k+1}$ 。

5.  $D_k$  :

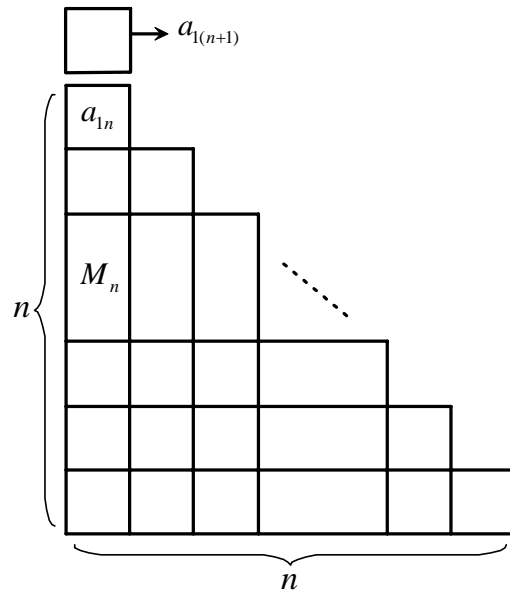
(1) 名詞解釋：

$D_k$  : 在  $n$  層階梯格子上從第一列上方往右下增添  $k$  個方格數(如圖)，但於右下最後增添的方格子不可與下方格子相連，如此將此棋盤形格子覆蓋完畢之“蛇填充數”稱之為  $D_k$ 。



(2) 證明： $D_k = F_{2k} M_n$  ,  $n+1 \geq k \geq 1$

證明：當  $k = 1$  時，左式 =  $D_1$



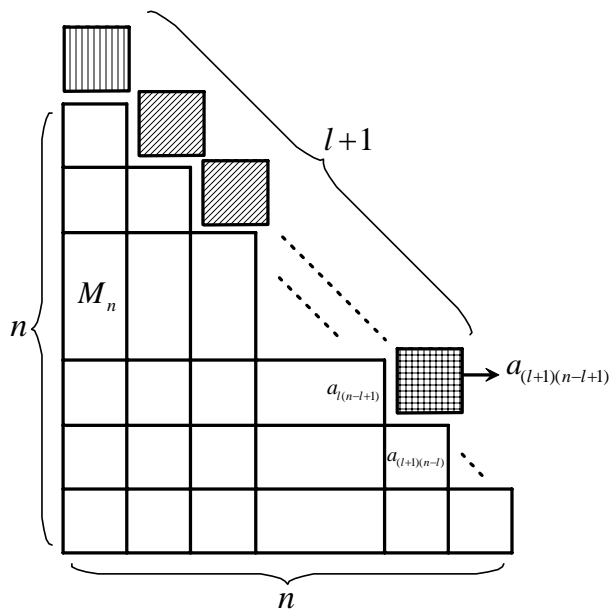
$$D_1 = M_n = F_2 M_n$$

$\therefore k = 1$  時成立

假設  $k = l$  時成立，即  $D_l = F_{2l} M_n$

則當  $k = l + 1$  時

左式 =  $D_{l+1}$



狀況甲：當  $a_{(l+1)(n-l+1)}$  完全不與其他相連時，則情況與  $A_l$  同，有  $F_{2l+1}$  種情形

狀況乙：當  $a_{(l+1)(n-l+1)}$  與  $a_{l(n-l+1)}$  相連時，則情況與  $D_l$  同，有  $F_{2l}$  種情形

又每種情形之蛇填充數為  $M_n$

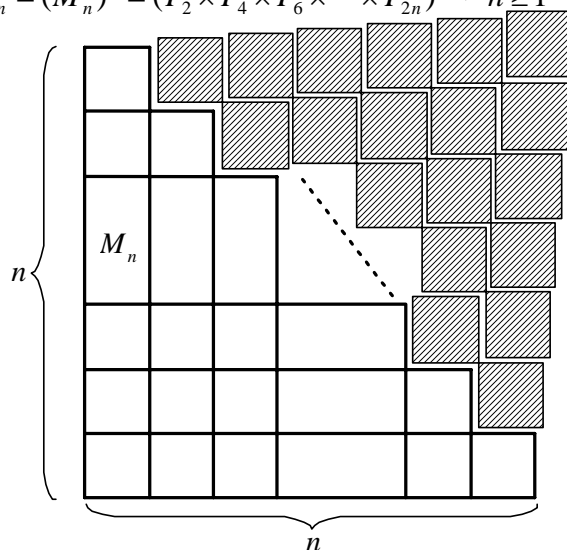
$$\therefore D_l = (F_{2l+1} + F_{2l})M_n = F_{2l+2}M_n$$

$$\therefore \text{由數學歸納法得證 } D_k = F_{2k}M_n, \quad n+1 \geq k \geq 1$$

- (3) 令  $d_k$  表示在  $n$  層階梯格子斜向增加  $k$  個格子 ( $n+1 \geq k \geq 1$ ) [其中若  $k=1$ ，表示此格子不可與左方的格子相連(或左方沒有格子)且不可與下方的格子相連(或下方沒有格子)；若  $k \geq 2$ ，表示左上方第一個格子不可與左方的格子相連(或左方沒有格子)，右下方最後一個格子不可與下方的格子相連(或下方沒有格子)，其他  $k-2$  個格子與其他格子的關係則無限制。]，將其完全覆蓋完畢之蛇填充數與原來  $n$  層階梯格子的蛇填充數之倍數關係。由(2)可得  $d_k = F_{2k}$ 。

6. 證明： $T_{n \times n} = (M_n)^2 = (F_2 \times F_4 \times F_6 \times \dots \times F_{2n})^2, \quad n \geq 1$

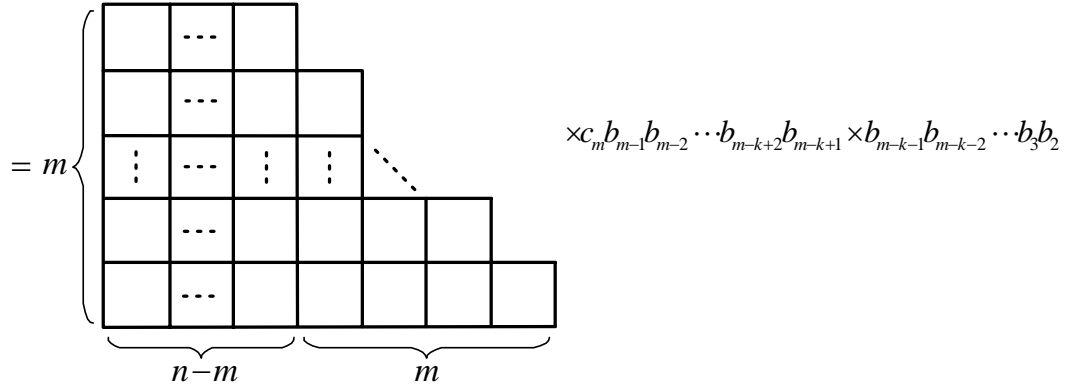
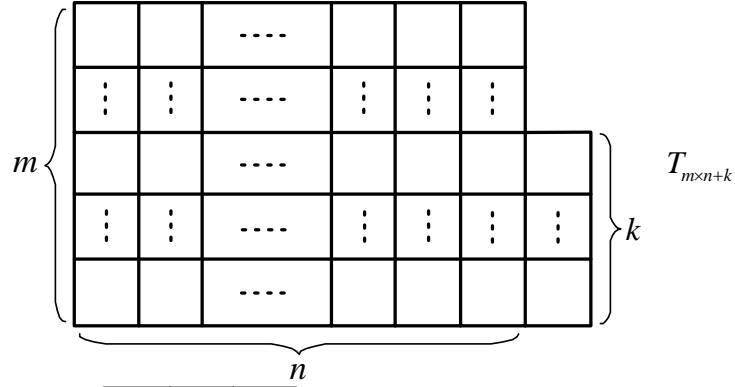
證明：



$$\begin{aligned}
T_{n \times n} &= M_n (b_{n-1} b_{n-2} b_{n-3} \times \cdots \times b_1) \\
&= M_n (F_{2n} F_{2n-2} F_{2n-4} \times \cdots \times F_4) \\
&= (M_n)^2 = (F_2 F_4 F_6 \times \cdots \times F_{2n})^2
\end{aligned}$$

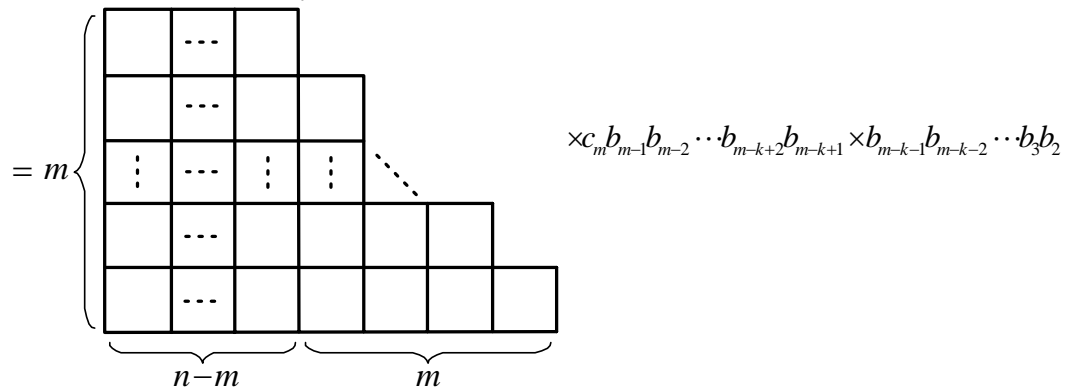
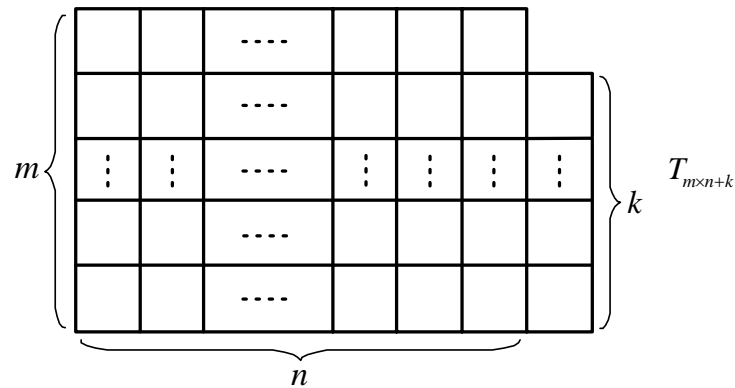
7. 證明： $T_{m \times n+k} = c_m T_{m \times (n-1)+k} = F_{2k+1} T_{m \times (n-1)+k}$ ， $n > m > k \geq 1$

證明：**(1)**  $T_{m \times n+k} = c_m T_{m \times (n-1)+k} = F_{2k+1} T_{m \times (n-1)+k}$ ， $n > m > k+1 \geq 2$



$$= c_m T_{m \times (n-1)+k}$$

**(2)**  $T_{m \times n+k} = c_m T_{m \times (n-1)+k} = F_{2k+1} T_{m \times (n-1)+k}$ ， $n > m = k+1 \geq 2$



$$= c_m T_{m \times (n-1)+k}$$

綜合(1)(2)可得  $T_{m \times n+k} = c_m T_{m \times (n-1)+k} = F_{2k+1} T_{m \times (n-1)+k}$  ,  $n > m > k \geq 1$

8. 證明： $T_{m \times (m+1)} = c_m T_{m \times m} = F_{2m+1} T_{m \times m}$  ,  $m \geq 1$

$$\begin{aligned} \text{證明：} T_{m \times (m+1)} &= M_m (c_m b_{m-1} b_{m-2} b_{m-3} \times \cdots \times b_1) \\ &= c_m \times M_m (b_{m-1} b_{m-2} b_{m-3} \times \cdots \times b_1) \\ &= c_m T_{m \times m} = F_{2m+1} T_{m \times m} \end{aligned}$$

9. 證明： $T_{m \times n} = (c_m)^{n-m} T_{m \times m} = (F_{2m+1})^{n-m} T_{m \times m}$  ,  $n > m \geq 1$

$$\begin{aligned} \text{證明：} \because T_{m \times (m+1)} &= c_m T_{m \times m} = F_{2m+1} T_{m \times m} \\ \therefore T_{m \times n} &= c_m T_{m \times (n-1)} = c_m \times c_m T_{m \times (n-2)} = c_m \times c_m \times c_m T_{m \times (n-3)} \\ &= \cdots = (c_m)^{n-m} T_{m \times m} = (F_{2m+1})^{n-m} T_{m \times m} \end{aligned}$$

10. 同理欲證明： $T_{(m+1) \times m} = c_m T_{m \times m} = F_{2m+1} T_{m \times m}$  ,  $m \geq 1$  , 或

$$T_{m \times n} = (c_n)^{m-n} T_{n \times n} = (F_{2n+1})^{m-n} T_{n \times n} , m > n \geq 1$$

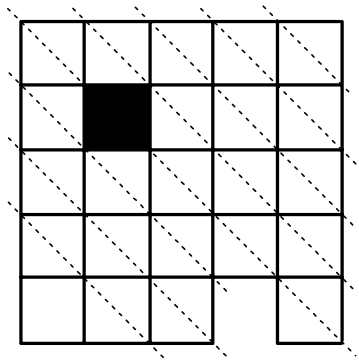
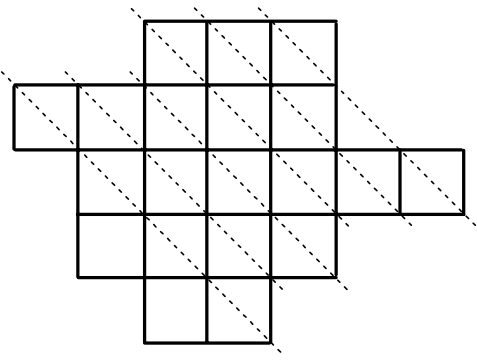
可以利用圖形與 8.9. 情況對稱直線  $y = x$  得證。

## 伍、研究結果：

- 一、一開始欲用遞迴關係以及利用 *Visual Basic* 軟體(如附件)來驗證答案，算到  $T_{5 \times 6} = 68387457600$  時，發覺欲以電腦程式來驗證答案結果，需花很長的時間 (備註：限於篇幅關係，將  $T_{5 \times 1} \sim T_{5 \times 6}$  做法省略)。我們改變一開始利用一個一個格子向上堆疊求取蛇填充數的方法，使用了一次斜向增加一至數個格子的方法 ( $a_k, b_k, c_k, d_k$ )，來求出蛇填充數。
- 二、棋盤格子形如： $T_{m \times n}, T_{m \times n+k}, M_n, A_k, B_k, C_k, D_k$  的蛇填充數均可表為費氏 (*Fibonacci*) 數列中某些項的乘積。

## 陸、討論：

- 一、若是棋盤不規則或有挖洞，亦可用  $a_k, b_k, c_k, d_k$  的方法找尋答案。例如：

	
<p>蛇填充數</p> $= 1 \times d_2 \times d_3 \times a_3 \times (a_1 c_2 d_1) \times (c_1 a_3) \times b_3 \times b_2 \times b_1$ $= F_4 F_6 F_7 F_3 F_5 F_2 F_3 F_7 F_8 F_6 F_4$ $= 40884480$	<p>蛇填充數</p> $= 1 \times (d_1 d_3) \times b_3 \times c_3 \times a_3 \times c_3 \times (b_1 c_1)$ $= F_2 F_6 F_8 F_7 F_7 F_7 F_4 F_3$ $= 2214576$



## 柒、結論：

棋盤格子形如： $T_{m \times n}$ ， $T_{m \times n+k}$ ， $M_n$ ， $A_k$ ， $B_k$ ， $C_k$ ， $D_k$ 的蛇填充數均可表為費氏 (*Fibonacci*) 數列中某幾項的乘積。

一、棋盤格子  $M_n = F_2 \times F_4 \times F_6 \times \cdots \times F_{2n}$ ， $n \geq 1$ 。

二、棋盤格子  $T_{n \times n} = (F_2 \times F_4 \times F_6 \times \cdots \times F_{2n})^2$ ， $n \geq 1$ 。

三、棋盤格子  $B_k = b_k M_n = F_{2k+2} M_n$ ， $n > k \geq 1$ 。

四、棋盤格子  $C_k = c_k M_n = F_{2k+1} M_n$ ， $n \geq k \geq 1$ 。

五、棋盤格子  $D_k = d_k M_n = F_{2k} M_n$ ， $n+1 \geq k \geq 1$ 。

六、棋盤格子  $T_{n \times n} = (M_n)^2 = (F_2 \times F_4 \times F_6 \times \cdots \times F_{2n})^2$ ， $n \geq 1$ 。

七、棋盤格子  $T_{m \times n+k} = c_m T_{m \times (n-1)+k} = F_{2m+1} T_{m \times (n-1)+k}$ ， $n > m > k \geq 1$ 。

八、棋盤格子  $T_{m \times (m+1)} = T_{(m+1) \times m} = c_m T_{m \times m} = F_{2m+1} T_{m \times m}$ ， $m \geq 1$ 。

九、棋盤格子  $T_{m \times n} = T_{n \times m} = (c_m)^{n-m} T_{m \times m} = (F_{2m+1})^{n-m} (F_2 \times F_4 \times F_6 \times \cdots \times F_{2m})^2$ ， $n > m \geq 1$ 。

## 捌、參考資料：

一、高級中學數學課本第一、四冊。

## 玖、附件 (Visual Basic 程式碼):

### 一、平面棋盤形格子 (Visual Basic 程式碼)

<pre>Dim box(100), dir(100), locus(1000), m, n, a As Integer Dim b, s, c, l, k(100) As Integer Dim sum As Long Private Sub Command1_Click() Command1.Enabled = False m = Val(Text1.Text) n = Val(Text2.Text) a = 0 '變數初始化 b = 0 c = 0 l = 0 sum = 0 For s = 0 To m * n - 1 box(s) = 0 dir(s) = 0 locus(s) = 0 k(s) = 0 Next s For s = 0 To 8 If Text4(s).Text &lt;&gt; "" And Val(Text4(s).Text) &gt; 0 And Val(Text4(s).Text) &lt;= m * n - 1 Then box(Val(Text4(s).Text)) = 2 Next s Call arrow Text3.Text = sum Command1.Enabled = True End Sub Sub arrow() step1: dir(a) = dir(a) + 1 Select Case dir(a) Case 1 '停留 If a &lt;= k(l) Then k(l) = a l = l + 1 box(a) = 1 locus(c) = a c = c + 1</pre>	<pre>step2: b = b + 1 If b = m * n Then sum = sum + 1 c = c - 1 b = locus(c) a = b GoTo step1 End If If box(b) &lt;&gt; 0 Then GoTo step2 a = b GoTo step1 Case 2 '向上走 If a &lt;= k(l) Then k(l) = a If a &gt;= m * n - n Then GoTo step1 If box(a + n) &lt;&gt; 0 Then GoTo step1 box(a) = 1 locus(c) = a c = c + 1 a = a + n GoTo step1 Case 3 '向右走 If a &lt;= k(l) Then k(l) = a If (a + 1) Mod n = 0 Then GoTo step1 If box(a + 1) &lt;&gt; 0 Then GoTo step1 box(a) = 1 locus(c) = a c = c + 1 a = a + 1 GoTo step1 Case 4 '無路可走，回到上一格 If a = 0 Then Exit Sub c = c - 1 l = l - 1 b = k(l) box(a) = 0 dir(a) = 0 a = locus(c) GoTo step1 End Select End Sub</pre>
---	--

A:當下執行動作的格子。

B:檢查格子是否空白，當循環到最後一格不空白，代表此方法結束。

C:用來當執行到第幾步驟，當此步驟的格子執行完動作則回到上一步驟。

K:代表格子連成的線的第一格。

Box:格子所成的陣列。

Dir:下一格的方向，有停留、往上、往右三種。

Locus:格子連線所成的軌跡。

## 二、空間狀況 (Visual Basic 程式碼)

<pre> Dim box(100), dir(100), locus(1000), m, n, a As Integer Dim b, s, c, l, k(100), space(30), z, i As Integer Dim sum As Currency Private Sub Command1_Click() Command1.Enabled = False m = Val(Text1.Text) n = Val(Text2.Text) z = Val(Text7.Text) a = 0 '變數初始化 b = 0 c = 0 l = 0 sum = 0 For s = 0 To m * n * z - 1 box(s) = 0 dir(s) = 0 locus(s) = 0 k(s) = 0 Next s Text6.Text = "" For s = 0 To i If space(s) &gt; 1 And space(s) &lt;= m * n * z - 1 Then     box(space(s)) = 2     Text6.Text = Text6.Text &amp; space(s) &amp; " " End If Next s Call arrow Text3.Text = sum Command1.Enabled = True Call Form_Load End Sub Sub arrow() step1: dir(a) = dir(a) + 1 Select Case dir(a) </pre>	<pre> Case 1 '停留     If a &lt;= k(l) Then k(l) = a     l = l + 1     box(a) = 1 locus(c) = a     c = c + 1 step2:     b = b + 1     If b = m * n * z Then         sum = sum + 1         c = c - 1         b = locus(c)         a = b         GoTo step1     End If     If box(b) &lt;&gt; 0 Then GoTo step2     a = b     GoTo step1 Case 2 '向前走     If a &lt;= k(l) Then k(l) = a     s = a \ (m * n) + 1     If a &gt;= m * n * s - n Then GoTo step1     If box(a + n) &lt;&gt; 0 Then GoTo step1     box(a) = 1     locus(c) = a     c = c + 1     a = a + n     GoTo step1 Case 3 '向右走     If a &lt;= k(l) Then k(l) = a     If (a + 1) Mod n = 0 Then GoTo step1     If box(a + 1) &lt;&gt; 0 Then GoTo step1     box(a) = 1     locus(c) = a     c = c + 1     a = a + 1 </pre>	<pre> GoTo step1 Case 4 '向上走     If a &lt;= k(l) Then k(l) = a     If a &gt;= m * n * (z - 1) Then GoTo step1     If box(a + m * n) &lt;&gt; 0 Then GoTo step1     box(a) = 1     locus(c) = a     c = c + 1     a = a + m * n     GoTo step1 Case 5 '無路可走，回到上一格     If a = 0 Then Exit Sub     c = c - 1     l = l - 1     b = k(l)     box(a) = 0     dir(a) = 0     a = locus(c)     GoTo step1 End Select End Sub Private Sub Command2_Click() If Val(Text4.Text) &lt;= 0 Then Exit Sub space(i) = Val(Text4.Text) i = i + 1 End Sub Private Sub Form_Load() i = 0 For s = 0 To 30 space(s) = 0 Next s End Sub </pre>
--	--	---

040418

1.

2.

Fi bonacci Numbers

3.