心得與反思

在這一年的校定必修中，我們在動態規劃題目的啟發下，希望以1*m方塊覆蓋矩形之方法數為題，找出其方法數的遞迴是並證明其正確性。

在一開始剛完成1*m方塊填入m*n空間中後，我們遇到了第一次的瓶頸：不知道後續較為複雜的狀態要如何分類與開始。而後我們嘗試了幾種方法，最後決定以開頭的排列方式分類討論，並分別找出其遞迴式再統合。這也讓我們後續的研究有了方向，使其更順暢。而第二次瓶頸則是發生在1*m方塊填入(m+1)*mn空間中的其中一種情形，在前面的研究中，我們都可以很直觀地找出其各字的遞迴式，但在這次，我們沒有辦法直接寫出。因此我們決定借助電腦的力量，跑出係數並觀察，而後發現了前綴和與巴斯卡三角形的性質，完成了證明（一）。但我們認為這樣的想法並不直觀，證明的過程也不夠漂亮，因此後來想出了證明（二）的排列想法的證明。第三次瓶頸則是發生在1*3方塊填入6*n空間中，我們發現因為它可以一行中同時擺兩個直的方塊，因此之前找出的遞迴是並不能套用在它身上。我們觀察它的排放方式後，覺得它和費氏數列有點類似。吸取了之前的經驗後，我們用電腦跑係數後發現它2的規律是$a_n = a_{n-2} + a_{n-3}$，因此開始嘗試證明出來。第四次的瓶頸是發生在1*2方塊填入2*5空間中，因為將開頭分類後，他們每一個分別的遞迴事都會有自已和別人，導致問題變得複雜。我們嘗試了幾種之前用過的方法與方向後，還是無法突破，而這也是我們未來想努力的方向。

在這一年的研究中，我感受到了觀察和證明的重要性，也在和同學和老師的討論中學習到了證明的想法與方法。在過程中和同學討論想法時，也會互相找出證明的漏洞或想出對方沒想出的方向，這讓研究進行得比較順利，同時也讓我從中學習到不一樣的想法。在這一年的研究中，讓我更感受到數學研究的有趣之處，雖然常常會被一個問題卡住而遲遲無法突破，但想出來時的快樂與成就感遠超於此。希望在未來也可以更深入的探討數學，學習更多相關知識，並完成現在尚未完成的部分。

數學校訂必修：1*m方塊覆蓋矩形之方法數

作者：劉蕃熙 董詠平

摘要

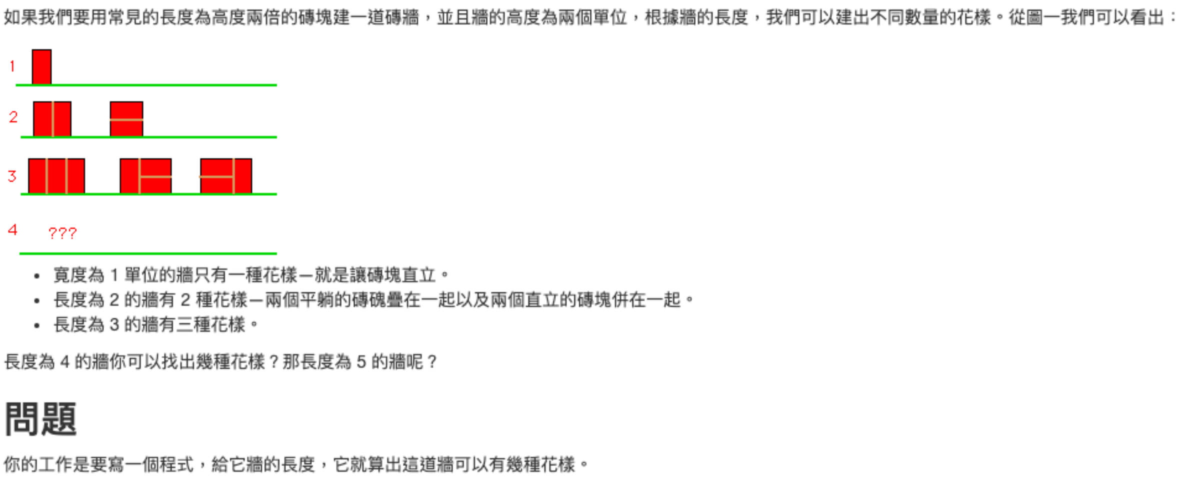
本次研究靈感來自於一個資訊題目：將1*2的方塊填出2*n的舉行之方法數。因此我們拓展問題，希望了解特定大小的方塊填入特定空間有多少種排法。

我們從1*2填入2*n格的格子著手，再將題目逐步複雜化，增加多出來的格數，找出其公式。我們將可填滿第一個直排的方法分類討論，找出每個情形下的遞迴式，並試圖找到某種規律去解釋此遞迴式。

壹、前言

一、研究動機

一次資訊課教到動態規劃時，在zerojudge寫到了這樣一題題目：


如果我們要用常見的長度為高度兩倍的磚塊建一道磚牆，並且牆的高度為兩個單位，根據牆的長度，我們可以建出不同數量的花樣。從圖一我們可以看出：

- 寬度為 1 單位的牆只有一種花樣—就是讓磚塊直立。
- 長度為 2 的牆有 2 種花樣—兩個平躺的磚塊疊在一起以及兩個直立的磚塊併在一起。
- 長度為 3 的牆有三種花樣。

長度為 4 的牆你可以找出幾種花樣？那長度為 5 的牆呢？

問題

你的工作是要寫一個程式，給它牆的長度，它就算出這道牆可以有幾種花樣。

這個題目可以發現它的遞迴式與費氏數列相同，而後我們就思考：如果將如果將題目延伸成1*3方塊填入3*n的空間的話遞迴式是什麼？1*m方塊填入(m+k)*n的空間的話遞迴式又是什麼呢？

二、研究目的

（一）、討論1*2方塊填入2*n空間中的方法數之遞迴式

（二）、討論1*m方塊填入m*n空間中的方法數之遞迴式

（三）、討論1*m方塊填入(m+k)*n空間中的方法數之遞迴式，並找出其通式

三、名詞解釋

$f(m,w,l)$:1*m的方塊填入w*l的方格裡的方法數

貳、研究設備及器材

紙、筆、電腦、Visual Studio Code

參、研究過程及方法

我們設放入m*n的空間時的方法數為f(n)，以下的遞迴式也會以此表示。我們的討論方法是將起始狀態分類討論，並分別找出以此方式為開頭的方法數後再統整。

一、1*2方塊填入2*n空間中

首先我們先證明1*2方塊放入2*n空間中的情形，我們發現在開頭僅有兩種擺放方式

(一)直立放：

在此情形下的排列方法數為$f(n-1)$

(二)橫著放：

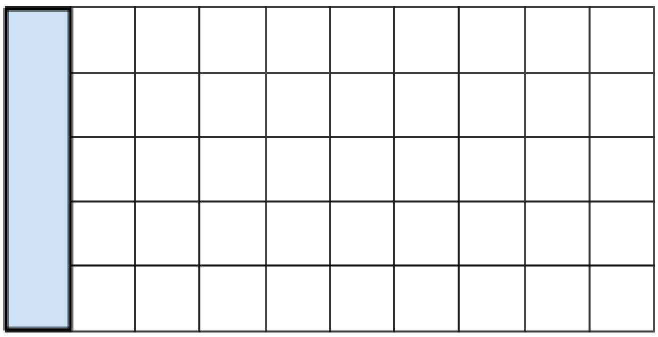
在此情形下的排列方法數為$f(n-2)$

由此可知遞迴式為$f(n)=f(n-1)+f(n-2)$, $f(0)=0$、$f(1)=1$

二、1*m方塊填入m*n空間中

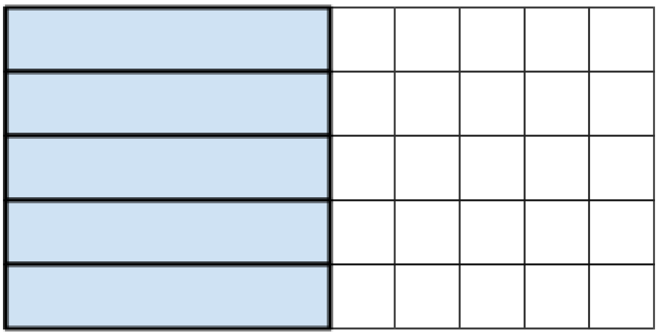
我們將開頭情形分成以下兩種情形

(一)直立放：



在此情形下的排列方法數為$f(n-1)$

(二)橫著放：



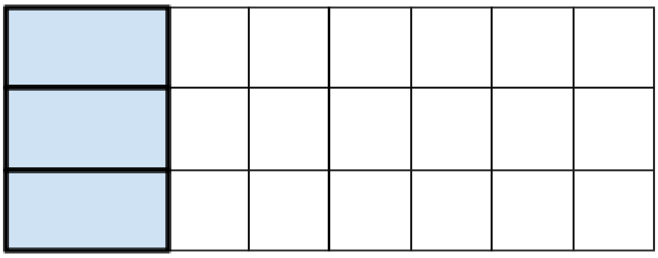
在此情形下的排列方法數為$f(n-m)$

由此可知遞迴式為$f(n)=f(n-1)+f(n-m),\ f(0\sim m-1)=1$

三、1*2方塊填入3*2n空間中

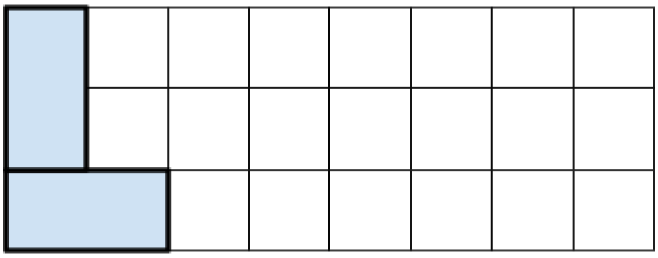
當問題延伸時，我們也利用相同的方法：將開頭的排法分類討論

（一）全部橫著放

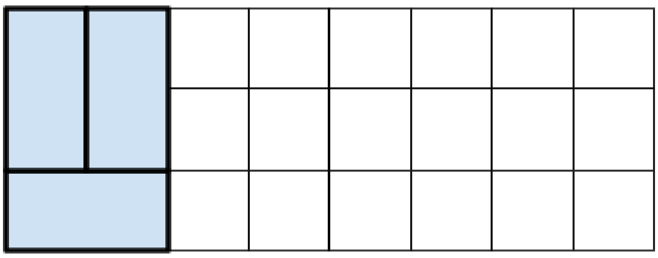


在此情形下，剩下2n-2行尚未排放，因此排列方法數為$f(2n-2)$

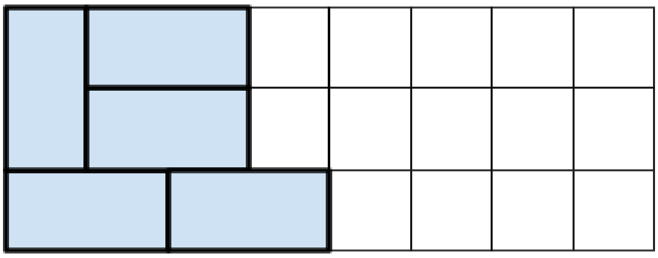
（二）上直下橫



在此情形下，下一個方塊有下列兩種排法



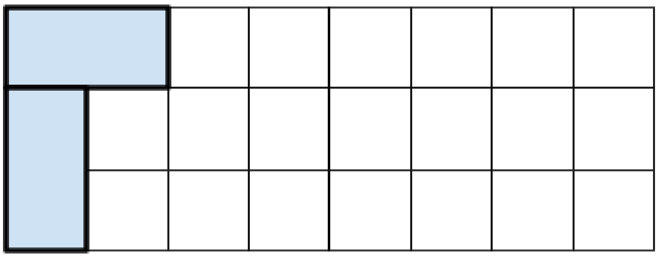
這一種排法因為它剛好填滿一直行且沒有多餘，可知剩下排列的方法數為$f(2n-2)$



這樣的情形下，我們發現他其實是第二種排法的n-2

因此我們可知第二種排法的方法數為$\sum_{i=1}^{i=n} f(2n-2i)$

（三）上橫下直



我們可知第三種排法其實與第二種相同，只是上下顛倒，因此排法數與第二種相同。

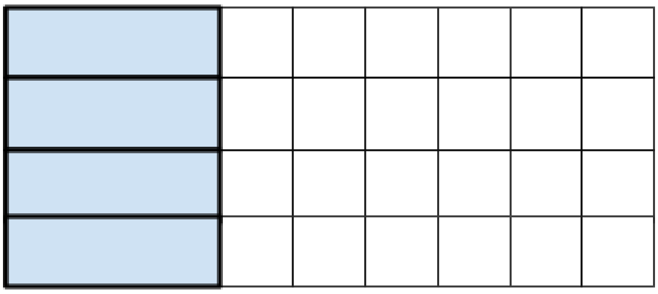
將以上三種情況的式子加總可得1*2方塊填入3*2n空間中的遞迴式為

$$f(n) = f(n-2) + 2 * \sum_{i=1}^{i=n} f(2n-2i),\ f(0) = 0、f(2) = 3$$

四、1*m方塊填入(m+1)*mn空間中

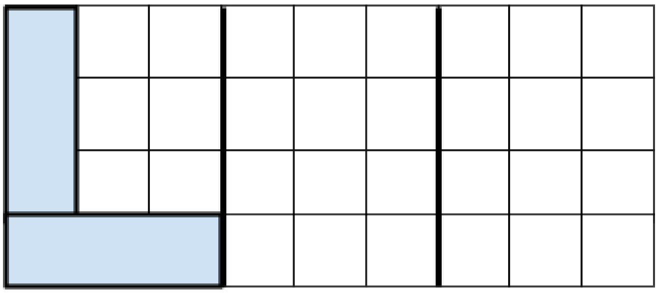
我們將1*2方塊填入3*2n空間的問題擴展成1*m方塊填入(m+1)*mn空間，並用相同方式分類開頭的排列情形討論。

（一）全部橫著放



在此情形下，剩下mn-m行尚未排放，因此排列方法數為$f(mn-m)$

（二）上直下橫



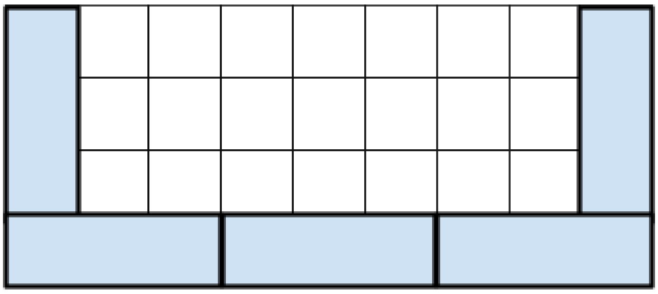
在此情形下，我們可以比照前面幾種遞迴式，寫出以下遞迴式
而我們有用兩種方式來證明：
證明一：
我們將其分成m排一組，設當每組剩下個數為k
當k=1時，$g(mn,1) = \sum_{i=2}^{i=n-1} g(mn - mi, 1) + 1$
當k=2時，因爲之後每都是k=1 + k=2，
$g(mn,2) = \sum_{i=2}^{i=n-1} (i - 1) * g(mn - mi, 2) + m$
……以此類推
將係數提出後得知每一項係數都為它前一項的前綴和
令每一項的係數為$t(m,i)$，則$t(m,i) = \sum_{k=2}^{k=m} t(k, i - 1)$
又$t(m - 1, i) = \sum_{k=2}^{k=m-1} t(k, i)$
因此可知$t(m,i) = t(m, i - 1) + t(m - 1, i)$
由此特性我們發現此係數表為巴斯可三角形，可以表示成
$\sum_{i=1}^{i=n} a_i * f(mn - im)$其中$a_i = C_{m-2}^{m+i-3}$

	$a_1 = C_0^{m-2}$	$a_2 = C_1^{m-1}$	$a_3 = C_2^{m}$	$a_4 = C_3^{m+1}$	$a_5 = C_4^{m+2}$	$a_6 = C_5^{m+3}$
$m = 2$	1	1	1	1	1	1
$m = 3$	1	2	3	4	5	6
$m = 4$	1	3	6	10	15	21
$m = 5$	1	4	10	20	35	56
$m = 6$	1	5	15	35	70	126

證明二：

$\sum_{i=1}^{i=n} a_i * f(mn - im)$

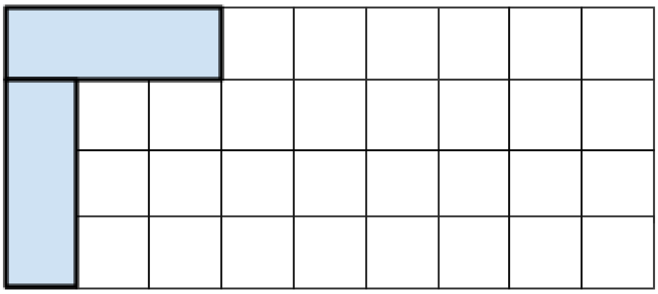
此處的a_i是指在前im行中，以上直下横的排法開頭，且恰好在第im行填滿的方法數。



由於需在最後一行恰好填平格子，所以頭尾必須擺直的方塊，且中間必需恰好有m-2個直的方塊、i-1組横的方塊。有了以上條件後，便可以將問題簡化成有m-2+i-1個空格需填入m-2個相同的東西以及另外i-1個相同的東西，則這種方法數可以用C_{m-2}^{m+i-3}來表示。

所以a_i=C_{m-2}^{m+i-3}。

（三）上横下直



我們可知第三種排法其實與第二種相同，只是上下顛倒，因此排法數與第二種相同。

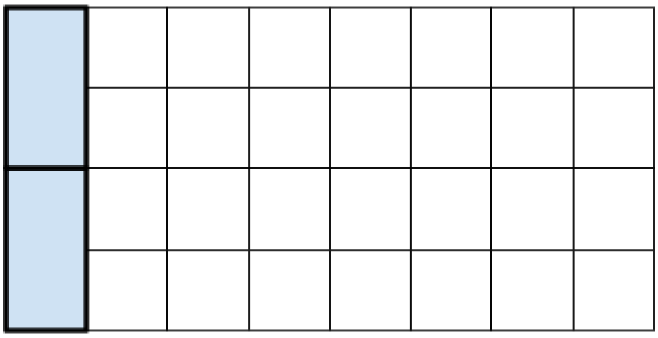
將以上三種情況的式子加總可得1*m方塊填入(m+1)*mn空間中的遞迴式為

$$f(mn) = f(mn - m) + 2 * \sum_{i=1}^{i=n} a_i * f(mn - mi), f(0) = 1、f(m) = 3$$

其中$a_i = C_{m-2}^{m+i-3}$。

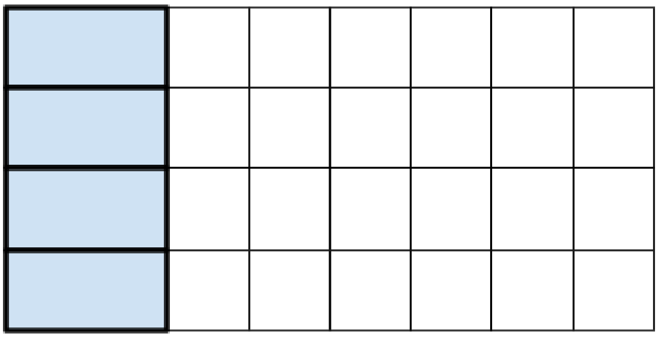
五、1*2方塊填入4*n空間中

（一）全部直著放



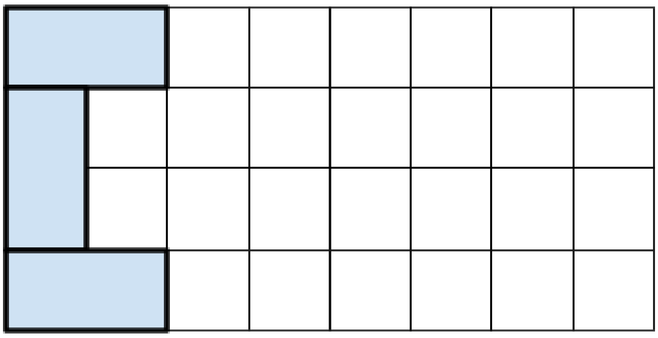
在此情形下，剩下n-1行尚未排放，因此排列方法數為$f(n-1)$

（二）全部橫著放

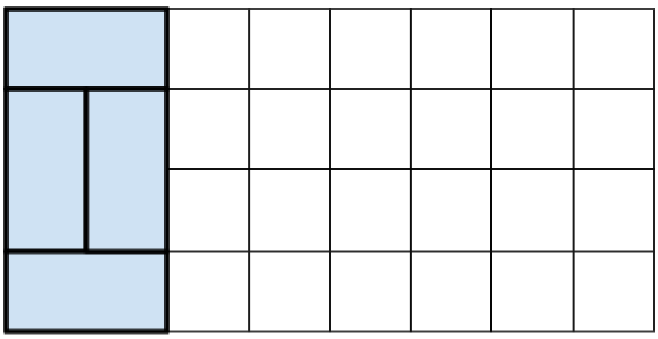


在此情形下，剩下n-2行尚未排放，因此排列方法數為$f(n-2)$

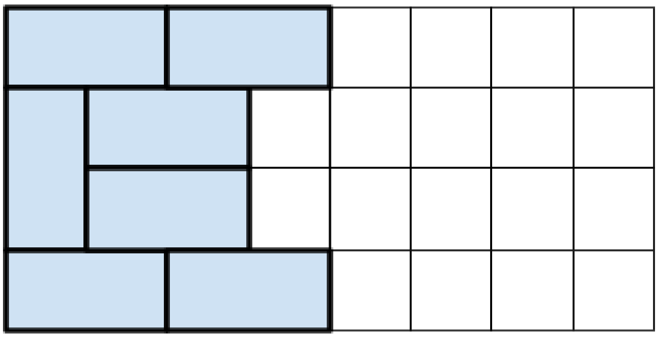
（三）上下直的放



在此情形下，下一個方塊有下列兩種排法



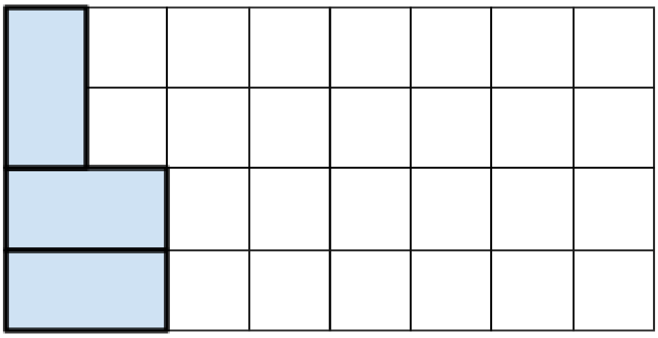
這一種排法因為它剛好填滿一直行且沒有多餘，可知剩下排列的方法數為$f(n-2)$



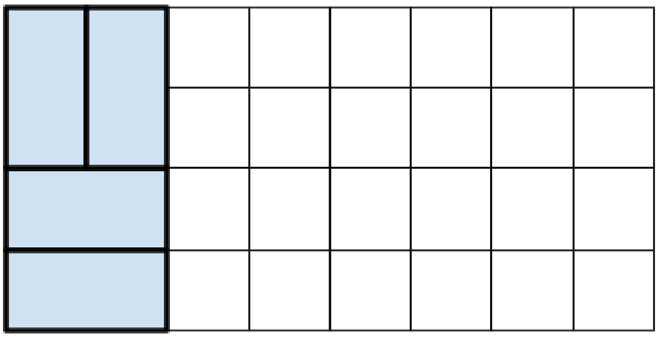
這樣的情形下，我們發現他其實是第三種排法的n-2

因此我們可知第三種排法的方法數為 $\sum_{i=1}^{i=\left[\frac{n}{2}\right]} f(n-2*i)$

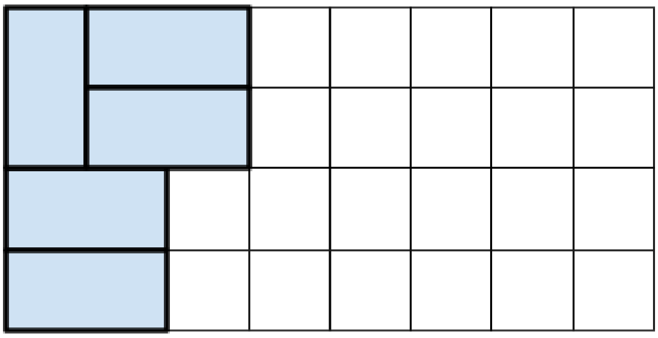
（四）上直下橫



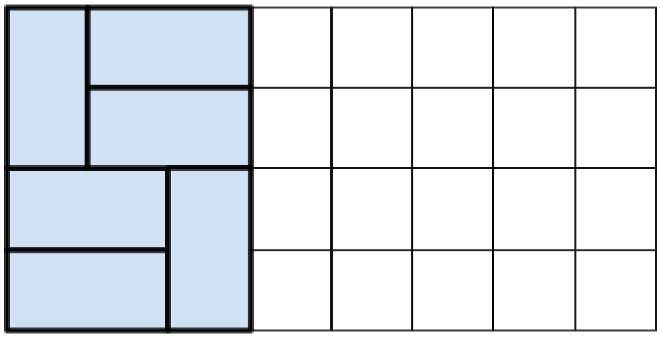
我們知道下一個方塊有兩種排法



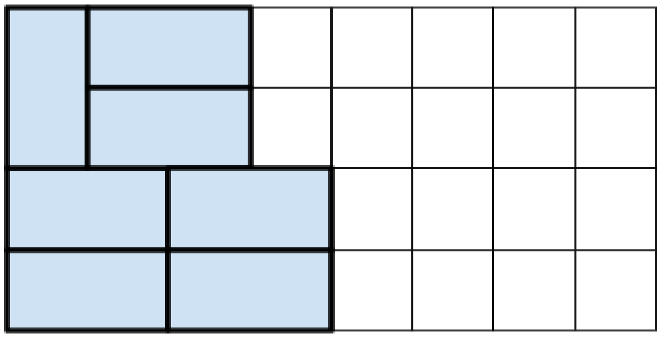
這一種排法因為它剛好填滿一直行且沒有多餘，可知剩下排列的方法數為$f(n-2)$



而若是用此方用此方法排列，則還有兩種排法



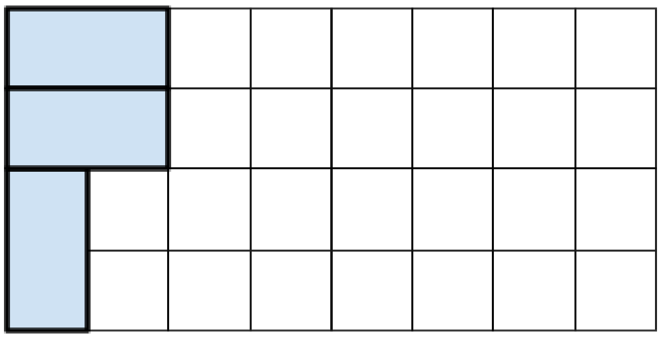
這一種排法也是剛好填滿一直行且沒有多餘，可知剩下排列的方法數為 $f(n-3)$



這樣的情形下，我們發現他其實是第四種排法的n-2，因此可以遞迴下去

因此我們可知第四種排法的方法數為 $\sum_{i=2}^{i=n-1} f(n-i)+2$

（五）下直上橫



我們可知第五種排法其實與第四種相同，只是上下顛倒，因此排法數與第四種相同。

將以上五種情況的式子加總可得1*2方塊填入4*n空間中的遞迴式為

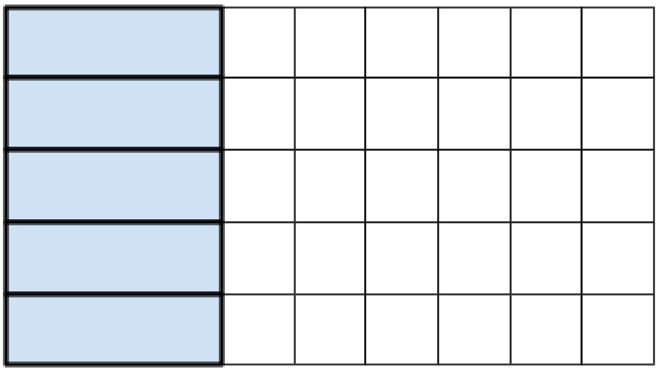
$$f(n)=f(n-1)+f(n-2)+\sum_{i=1}^{i=\left[\frac{n}{2}\right]} f(n-2*i)+2*\sum_{i=2}^{i=n-1} f(n-i)+2$$

其中$f(1)=1$、$f(2)=5$、$f(3)=11$

六、1*m方塊填入(m+2)*mn空間中

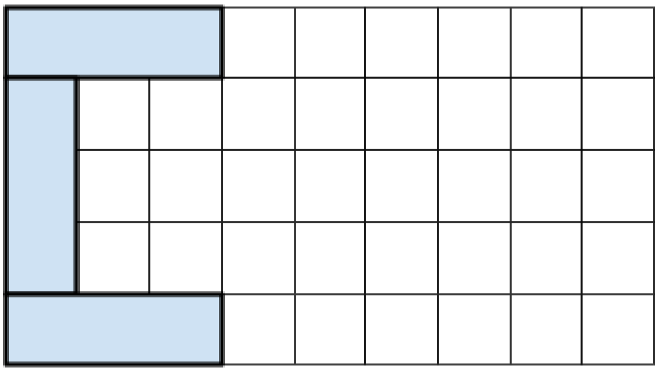
我們將1*2方塊填入4*n空間的問題擴展成1*m方塊填入(m+2)*mn空間，並用相同方式分類開頭的排列情形討論。

（一）全部橫著放



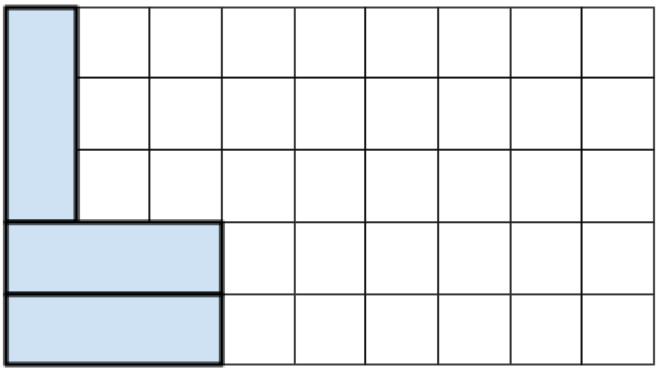
在此情形下，剩下mn-m行尚未排放，因此排列方法數為$f(mn - m)$

（二）上下直的放

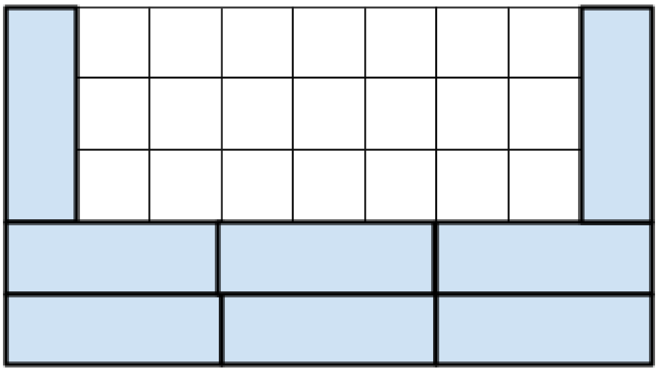


這種方法事實上和**四**(二)相同，只是上面多了一排，所以方法數也是$\sum_{i=1}^{i=n} a_i * f(mn - im)$，其中$a_i = C_{m-2}^{m+i-3}$

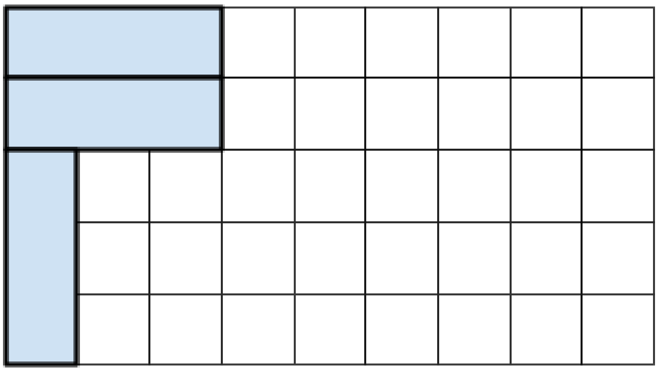
（三）上直下橫



這種方法事實上和**四**(二)相同，只是下面多了一排，所以方法數也是$\sum_{i=1}^{i=n} a_i * f(mn - im)$，其中$a_i = C_{m-2}^{m+i-3}$。



（四）上橫下直



我們可知第四種排法其實與第三種相同，只是上下顛倒，因此排法數與第三種相同。

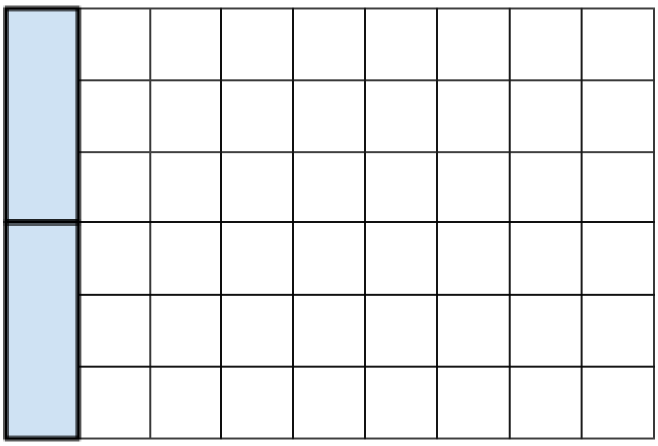
將以上四種情況的式子加總可得1*m方塊填入(m+2)*mn空間中的遞迴式為

$$f(mn) = f(mn - m) + 3\sum_{i=1}^{i=n} a_i * f(mn - mi)$$

$$f(0) = 1、f(m) = 4,\ a_i = C_{m-2}^{m+i-3}$$

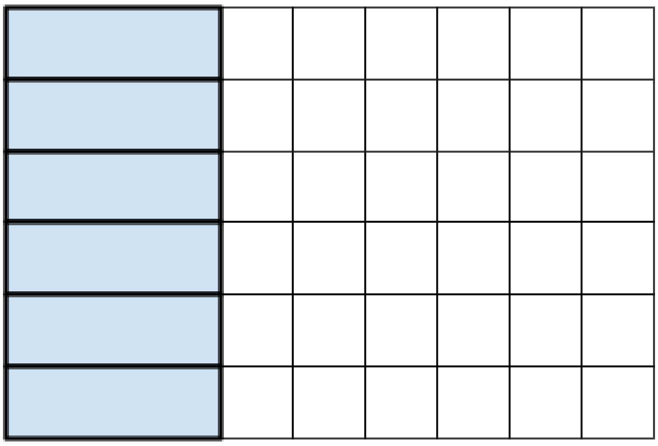
七、1*3方塊填入6*n空間中

（一）全部直著放



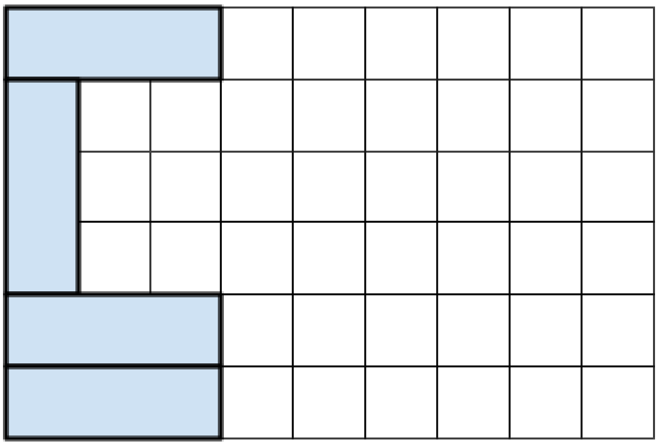
在此情形下，剩下n-1行尚未排放，因此排列方法數為$f(n - 1)$

（二）全部橫著放



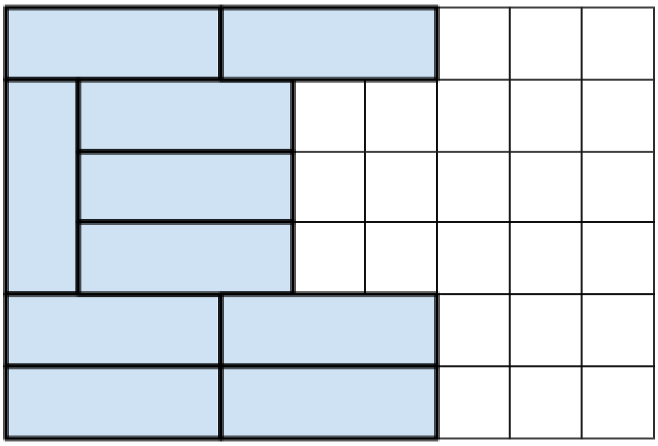
在此情形下，剩下n-3行尚未排放，因此排列方法數為$f(n-3)$

（三）上一直下二直

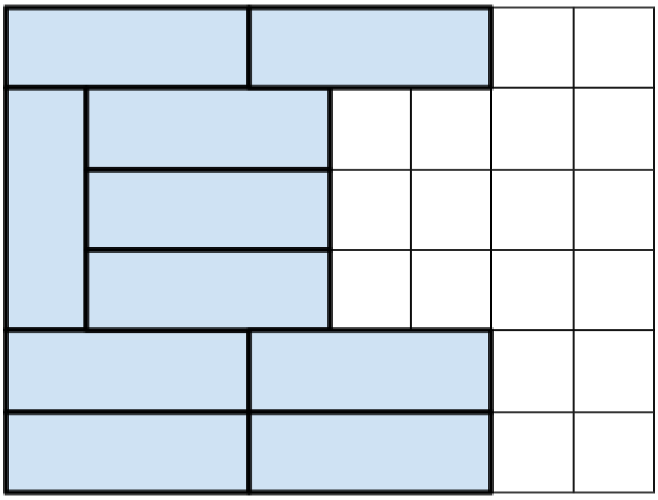


在此情形下，我們知道可以用將中間填滿再放了幾個值得方塊來分類

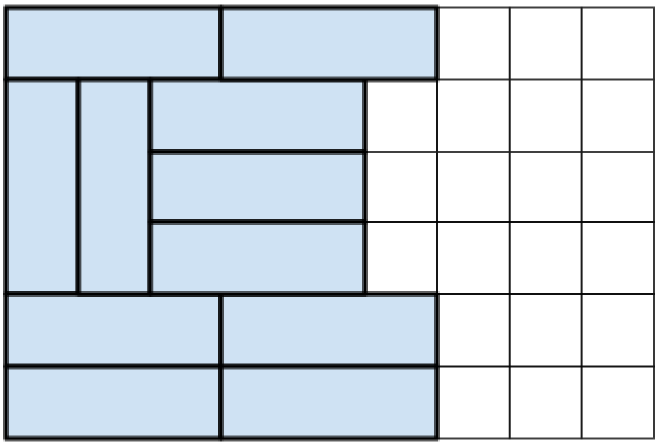
零個直的方塊：



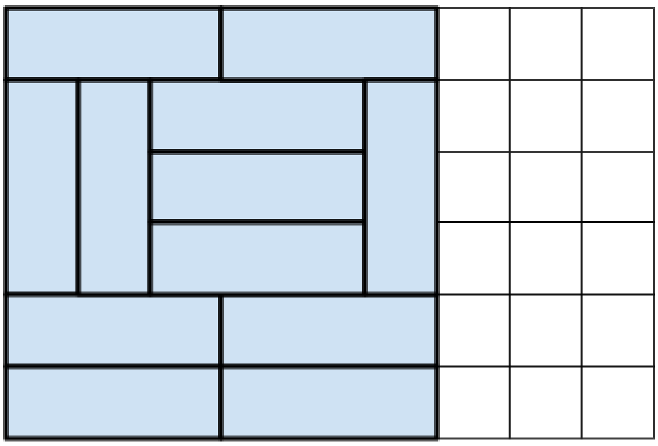
我們可知其其實等於排法三的n-3情形，但在剩下全空行數不足三時必須停止（如下圖），因此可以利用遞迴式求得時必須取$\frac{n}{3}$的下高斯



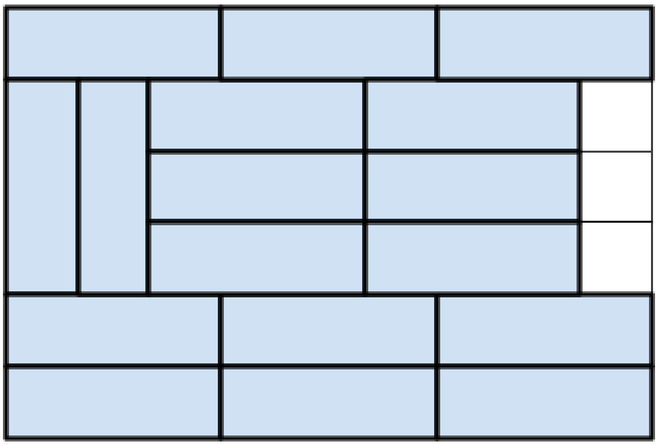
一個直的方塊：



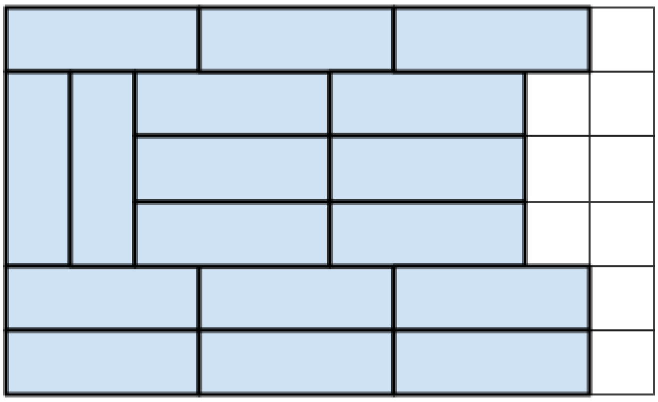
在此情形下，還有兩種排列方式



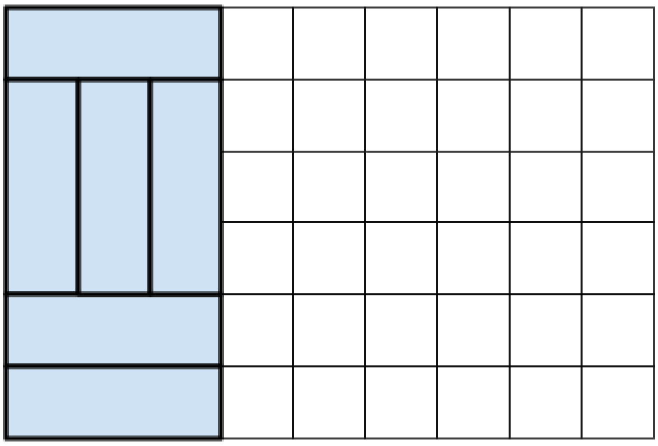
這一種排法剛好填滿一直行且沒有多餘，可知剩下排列的方法數為 $f(n - 3 * 2)$



而在此情形下時，則其就是排法三中的一個直n-3*2，但在剩下全空行數不足三時必須停止（如下圖），因此可遞迴求得時必須取$\frac{n}{3}$的下高斯



二個直的方塊：

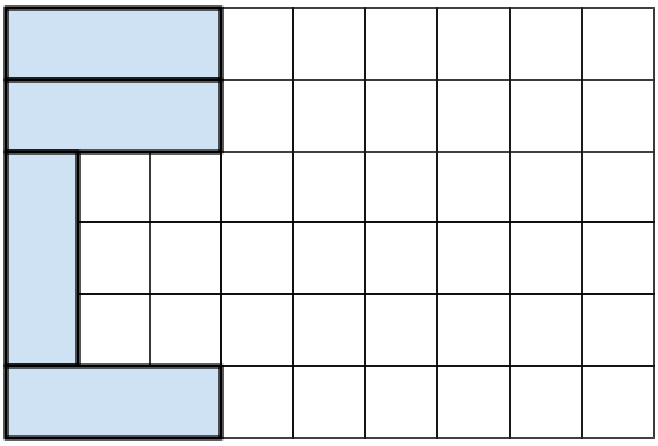


這一種排法剛好填滿一直行且沒有多餘，可知剩下排列的方法數為$f(n-3)$

由上述三種分類的排法相加，我們可得知剩下的排法數為

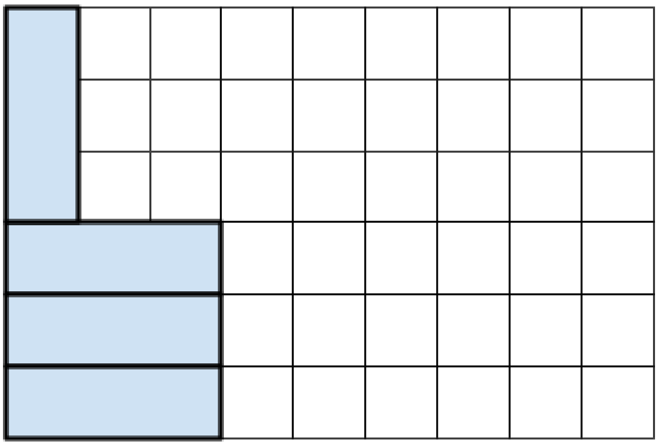
$$\sum_{i=1}^{i=\left[\frac{n}{3}\right]} i * f(n-3i)$$

（四）上二直下一直



我們可知第四種排法其實與第三種相同，只是上下顛倒，因此排法數與第三種相同。

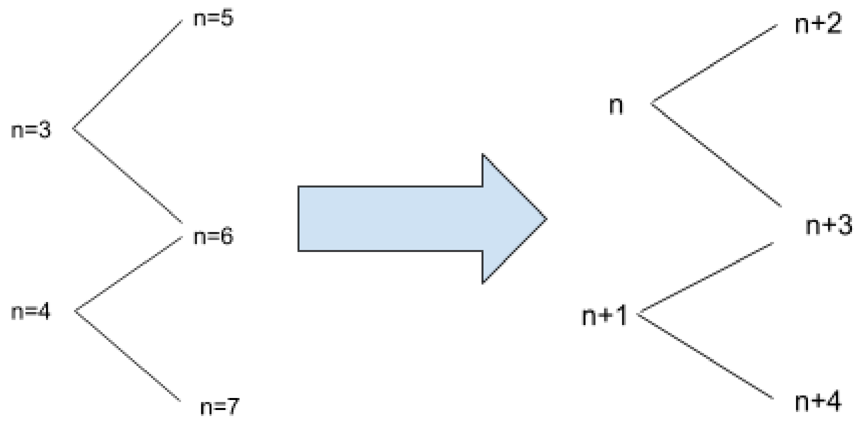
（五）上直下橫



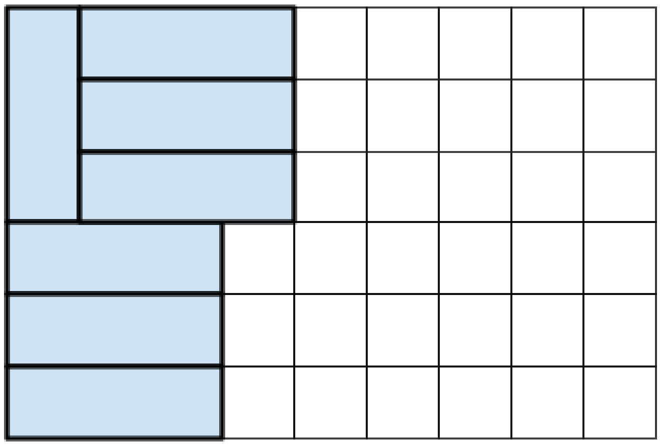
由於下半部分不像前面幾個情況只有一種擺法，所以必須要另外想出新的簡化方法來得到每一項遞迴的係數。我們將排法數利用程式暴力算出來之後觀察規律，發現了這樣的規律:$a_n = a_{n-2} + a_{n-3}$其中n代表恰好在第n行排滿的方法數，接著我們試圖尋找這個規律的證明方式。

證明:

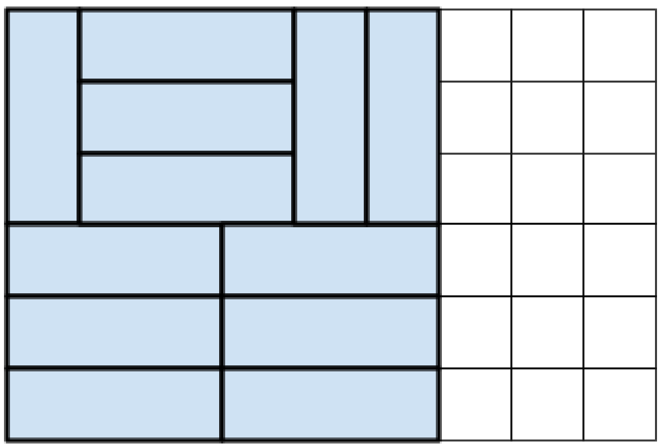
我們觀察到對所有多出一行的情況，有唯一的方法將行數增加2和3，且因為在將行數增加2的情況裡已涵蓋了多出兩行(直接填滿或是變成多一行)的情況，所以我們可以忽略不考慮，將以上關係以樹狀圖來表示則如下:



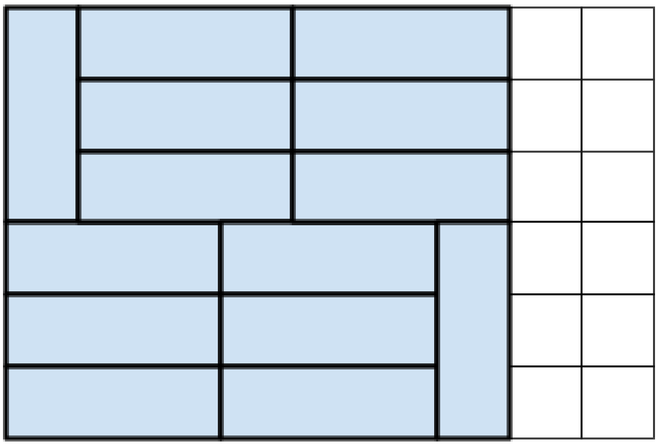
多一行的情況:



行數+2(多兩行的情況):

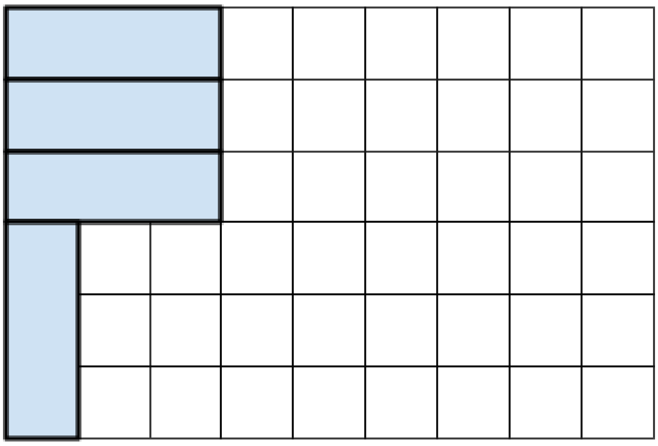


行數+3:



反推回去的話，就可以知道對此係數數列的第n項都是第n-2項與第n-3項的和。

（六）上橫下直



我們可知第六種排法其實與第五種相同，只是上下顛倒，因此排法數與第五種相同。

將以上六種情況的式子加總可得1*3方塊填入6*n空間中的遞迴式為

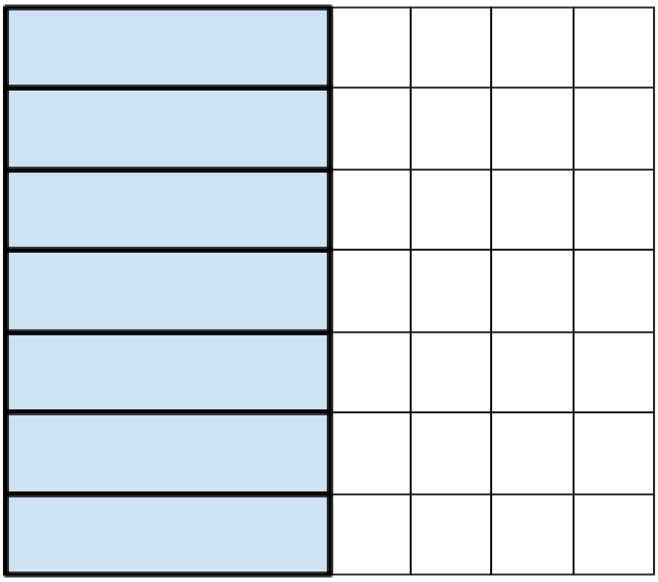
$$f(n) = f(n-1) + f(n-3) = 2 * \left(\sum_{i=3}^{i=n} a_i * f(n-i) + \sum_{i=1}^{i=\left[\frac{n}{3}\right]} i * f(n-3i) \right)$$

其中$f(0) = f(1) = f(2) = 1$、$a_n = a_{n-2} + a_{n-3}$、$a_3 = a_4 = a_5 = 1$

八、當$m > k$時，1*m方塊填入(m+k)*mn空間中

我們可知，當$m > k$時，因一直行無法放入兩個直放的方塊，因此我們可將其開頭情形分成以下兩類：

（一）：沒有直放方塊

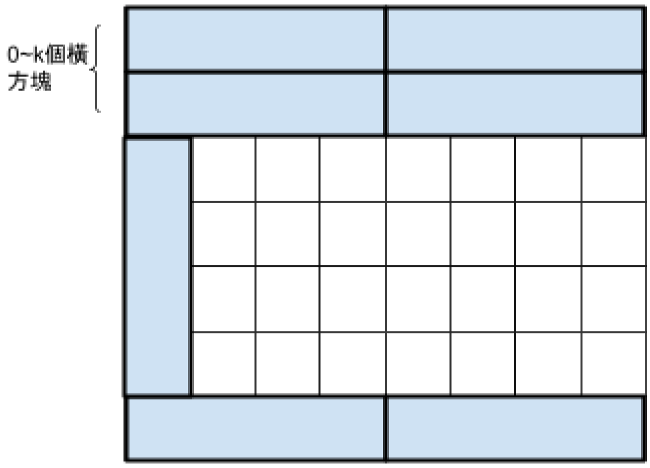


在此情形下，剩下mn-m行尚未排放，因此排列方法數為$f(mn-m)$

（二）：有一個直放方塊

我們可知在直放方塊上方可放置0~k橫向方塊（如下圖），又每一種情形的方法數都事實上都和**四**(二)相同，只是上面還有0~k個橫向方塊，所以方法數是

$(k+1)*\sum_{i=1}^{i=n} a_i * f(mn-im)$，其中$a_i = C_{m-2}^{m+i-3}$



將以上兩種情況的式子加總可得1*m方塊填入(m+k)*mn空間中的遞迴式為

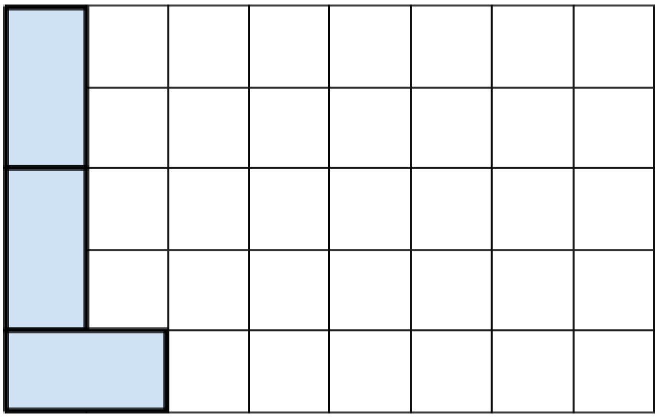
$$f(n) = f(mn-m) + (k+1)*\sum_{i=1}^{i=n} a_i * f(mn-im)$$

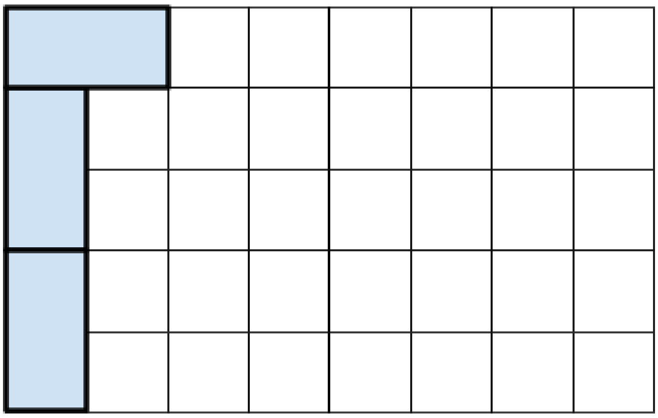
其中$a_i = C_{m-2}^{m+i-3}$, $f(1) = 1$、$f(m) = k+2$

八、1*2方塊填入2*5空間中

我們一樣依照開頭方式五大分類討論其各自的遞迴式，之後我們同時介紹其弟回式分類及其遞回式

（一）上一直下二橫/上一橫下二直



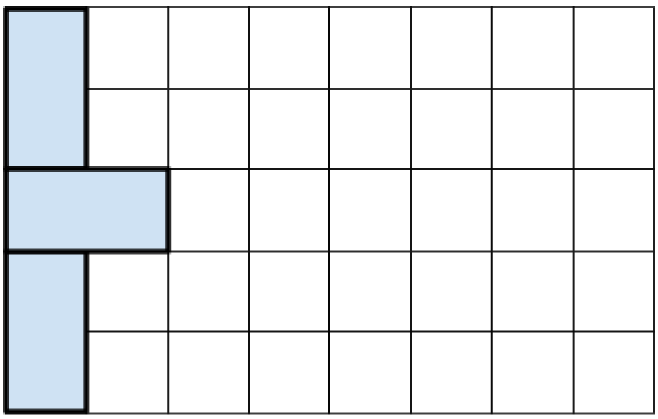


我們稱這兩種的遞迴式為$g(n,1)$，且

$$g(n,1) = f(2,5,n-2) + 3 * g(n-2,1)$$

$$g(n-2,2) + g(n-2,3) + g(N-2,4) + 1$$

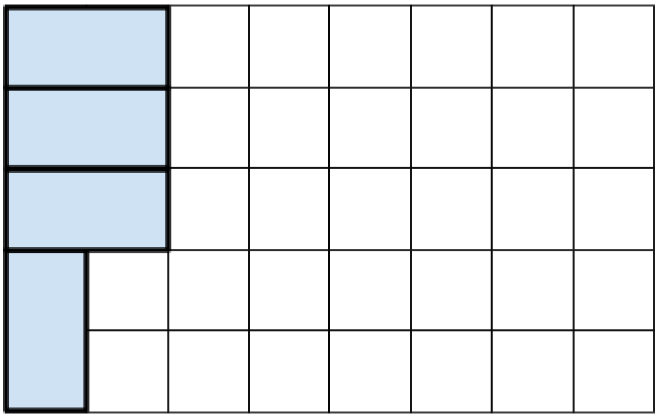
（二）上下直

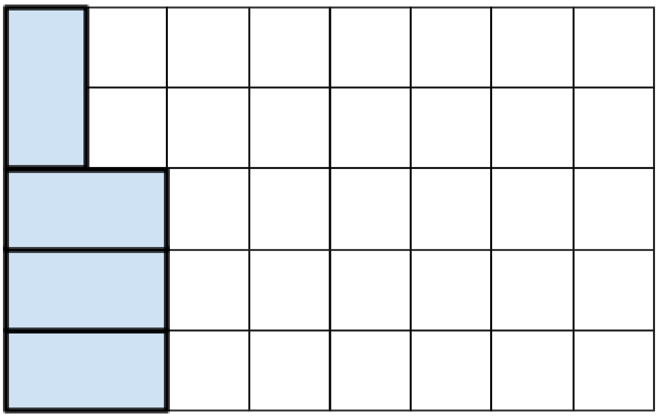


我們稱這種的遞迴式為$g(n,2)$，且

$$g(n,1) = f(2,5,n-2) + 2 * g(n-2,1) + 3 * g(n-2,2) + 2 * g(n-2,3)$$

（三）上橫下直/上直下橫

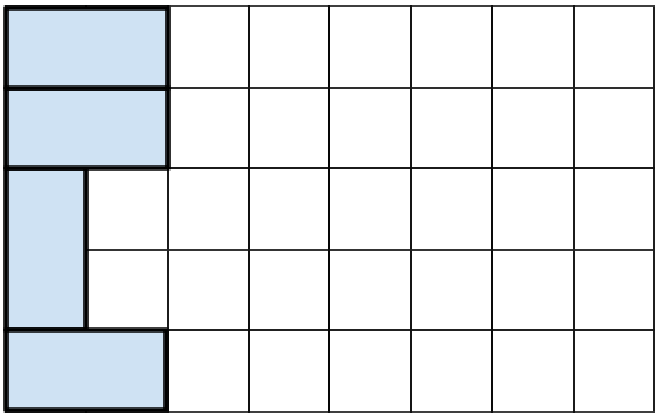


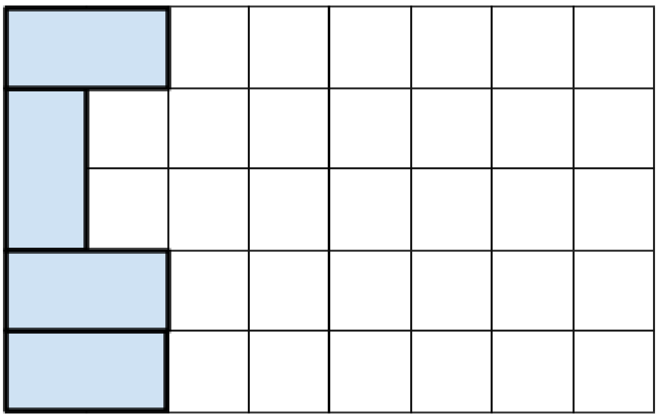


我們稱這兩種的遞迴式為$g(n,3)$，且

$$g(n,1) = f(2,5,n-2) + g(n-2,1) + g(n-2,2) + g(n-2,3)$$

（四）上二橫下一橫/上一橫下二橫

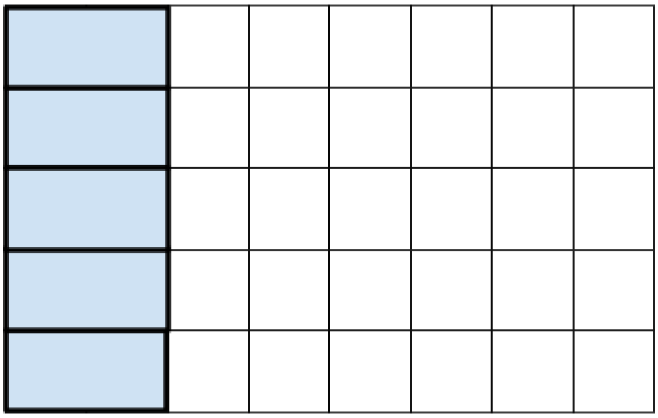




我們稱這兩種的遞迴式為$g(n,4)$，且

$$g(n,1) = f(2,5,n-2) + g(n-2,1) + g(n-2,4)$$

（五）全部橫著放



我們稱這種的遞迴式為$g(n,5)$，且$g(n,1) = f(2,5,n-2)$

我們嘗試突破個別的遞迴式，因此我們使用了c++及動態規劃來跑結果和係數（附錄一、二），希望可觀察到他們的規律，但最後很可惜並沒有發現，因此這一部份就做到個別開頭遞迴式的階段。

肆、研究結果

一、從1*2方塊填入2*n空間中的排列方法推展至1*m方塊填入m*n空間中的排列方法數為

$$f(n) = f(n-1) + f(n-m),\ f(0\sim m-1) = 1$$

二、從1*2方塊填入3*2n空間中的排列方法推展至1*m方塊填入(m+1)*mn空間中的排列方法數為

$$f(mn) = f(mn-m) + 2 * \sum_{i=1}^{i=n} a_i * f(mn-mi)$$

$$f(0) = 1、f(m) = 3,\ 其中a_i = C_{m-2}^{m+i-3}$$

三、從1*2方塊填入4*n空間中的排列方法推展至1*m方塊填入(m+2)*mn空間中的列方法數為

$$f(mn) = f(mn-m) + 3\sum_{i=1}^{i=n} a_i * f(mn-mi)$$

$$f(0) = 1、f(m) = 4,\ a_i = C_{m-2}^{m+i-3}$$

四、1*3方塊填入6*n空間中的排列方法數為

$$f(n) = f(n-1) + f(n-3) = 2 * \left(\sum_{i=3}^{i=n} a_i * f(n-i) + \sum_{i=1}^{i=\left[\frac{n}{3}\right]} i * f(n-3i)\right)$$

$f(0) = f(1) = f(2) = 1$、其中$a_n = a_{n-2} + a_{n-3}$、$a_3 = a_4 = a_5 = 1$

五、當$m > k$時，1*m方塊填入(m+k)*mn空間中的排列方法數為

$$f(n) = f(mn - m) + (k + 1) * \sum_{i=1}^{i=n} a_i * f(mn - im)$$

其中$f(1) = 1$、$f(m) = k + 2$, $a_i = C_{m-2}^{m+i-3}$

伍、討論與展望

我們從一開始最簡單的1*2填入2*n格的格子著手，推展到從1*m填入m*n格的格子，再將題目逐步複雜化，從多出一格再到多出兩格，先從簡單的數字版著手，再推展成公式化的解，原以為數字版的會比較好做，但是實際列出情況後才發現其實並沒有，因為在數字小的情況、比較不容易限制方塊的排法。但是在這些遞迴式裡我們只找出了一對一對應的關係，只能分別列出公式，還無法找到更直覺，更簡單化的公式，未來希望可以將1*2填入5*n整理出更一致的遞迴式、以及m*n填入x*y的格子的情況解完，更希望能找到某種規律去解釋這全部的遞迴式。

陸、結論

在此次研究中，我們透過分類首排完全覆蓋方塊的方法並找出遞迴式的方式進行研究，找出並證明1*2方塊填入2*n空間至1*3方塊填入6*n空間遞迴式，以及當$m > k$時，1*m方塊填入(m+k)*mn空間中的通式。

柒、參考資料

動機來源題目：https://zerojudge.tw/ShowProblem?problemid=d038

捌、附錄

（一）1*2方塊填入2*5空間中-結果

```
#include <bits/stdc++.h>
using namespace std;
typedef long long ll;
int dp[100][5]={0};
int main(){
    ios_base::sync_with_stdio(0);
    cin.tie(0);
    int n; cin >> n;
    for(int i=1; i<5; i++) dp[1][i]=1;
    for(int i=2; i<=n; i++){
        //for(int j=0; j<5; j++) dp[i][0]+=dp[i-1][j];
        dp[i][1]=3*dp[i-1][1]+2*dp[i-1][2]+dp[i-1][3]+dp[i-1][4];
        dp[i][2]=dp[i-1][1]+3*dp[i-1][2]+dp[i-1][3];
        dp[i][3]=dp[i-1][1]+2*dp[i-1][2]+dp[i-1][3];
        dp[i][4]=dp[i-1][1]+dp[i-1][4];
    }
    for(int j=1; j<=n; j++){
        ll a=1;
        for(int i=1; i<5; i++){
            a+=dp[j][i];
            cout << dp[j][i] << ' ';
        }
        cout<<a;
        /*int a=dp[j][0];
        for(int i=1; i<5; i++) a-=dp[j][i];
        cout<<a;*/
        cout << '\n';
    }
}
```

```
10
1 1 1 1 5
7 5 4 2 19
37 26 21 9 94
193 136 110 46 486
1007 711 575 239 2533
5257 3715 3004 1246 13223
27451 19406 15691 6503 69052
143359 101360 81954 33954 360628
748705 529393 428033 177313 1883445
3910247 2764917 2235524 926018 9836707
```

（二）1*2方塊填入2*5空間中-係數

```cpp
#include <bits/stdc++.h>
using namespace std;
typedef long long ll;
int dp[100][5]={0};
int main(){
    ios_base::sync_with_stdio(0);
    cin.tie(0);
    int n; cin >> n;
    for(int i=1; i<5; i++) dp[1][i]=1;
    for(int i=2; i<=n; i++){
        //for(int j=0; j<5; j++) dp[i][0]+=dp[i-1][j];
        dp[i][1]=3*dp[i-1][1]+2*dp[i-1][2]+dp[i-1][3]+dp[i-1][4];
        dp[i][2]=dp[i-1][1]+3*dp[i-1][2]+dp[i-1][3];
        dp[i][3]=dp[i-1][1]+2*dp[i-1][2]+dp[i-1][3];
        dp[i][4]=dp[i-1][1]+dp[i-1][4];
    }
    for(int j=1; j<=n; j++){
        ll a=1;
        for(int i=1; i<5; i++){
            a+=dp[j][i];
            cout << dp[j][i] << ' ';
        }
        cout<<a;
        /*int a=dp[j][0];
        for(int i=1; i<5; i++) a-=dp[j][i];
        cout<<a;*/
        cout << '\n';
    }
}
```

```
10
1 1 1 1 5
7 5 4 2 19
37 26 21 9 94
193 136 110 46 486
1007 711 575 239 2533
5257 3715 3004 1246 13223
27451 19406 15691 6503 69052
143359 101360 81954 33954 360628
748705 529393 428033 177313 1883445
3910247 2764917 2235524 926018 9836707
```