中華民國第52屆中小學科學展覽會
作品說明書

高中組　數學科

佳作

040402

Avoid123

學校名稱：國立宜蘭高級中學

作者：	指導老師：
高二 梅治玄 高二 林雋庭 高二 蘇芃之	李光祥

關鍵詞：Avoid、矩陣、遞迴關係

摘要

在 n 個數字當中連續三個數字皆未出現連續遞增情形者我們將其稱作為 aviod123。

在 n 個數字當中連續三個數字皆未出現中小大排列情形者我們將其稱作為 aviod213。

我們針對這個問題用反面，分類的方式是來處理。將其依照前兩個數字的不同分類成下表

$$
\begin{array}{ccccc}
12xxx & 21xxx & \dots & \dots & n1xxx \\
13xxx & 23xxx & \dots & \dots & n2xxx \\
14xxx & 24xxx & \dots & \dots & n3xxx \\
\dots & \dots & \dots & \dots & \dots \\
1nxxx & 2nxxx & 3nxxx & \dots & n(n-1)xxx
\end{array}
$$

並將表中每一個情況所會產生的排法寫成矩陣形式配合遞迴關係式我們將會得到以下結果

1. aviod123 的結果：

$$
T_n=\begin{bmatrix}
A_{n-2}(0)+(n-2)(n-3)! & A_{n-1}(1)-A_{n-1}(0) & A_{n-1}(1)-A_{n-1}(0) & \dots & \dots & A_{n-1}(1)-A_{n-1}(0) \\
A_{n-2}(1)+(n-3)(n-3)! & A_{n-2}(1)+(n-3)(n-3)! & \ddots & \dots & \dots & A_{n-1}(2)-A_{n-1}(1) \\
\vdots & \dots & \ddots & \ddots & \dots & \vdots \\
\vdots & \dots & \dots & \ddots & \ddots & \vdots \\
A_{n-2}(n-2)+0(n-3)! & \dots & \dots & \dots & A_{n-2}(n-2)+0(n-3)! & A_{n-1}(n-1)-A_{n-1}(n-2)
\end{bmatrix}
$$

$$
T_2=\begin{bmatrix}0 & 0\end{bmatrix} \quad T_3=\begin{bmatrix}1 & 0 & 0 \\ 0 & 0 & 0\end{bmatrix}
$$

其中 $A_n(n)$為矩陣中所有元素的總合(被 aviod123 的數量)

我們將其用 sigma 表示並化簡，化簡如下：

$$
\begin{aligned}
A_n(n) = &\sum_{k=1}^{n-1}\sum_{l=k}^{n-1}\sum_{s=1}^{l-1}\sum_{t=s}^{n-2}(n-t-3)(n-5)! + \sum_{k=1}^{n-1}\sum_{l=k}^{n-1}\sum_{s=1}^{l-1}\sum_{t=s}^{n-2}\sum_{r=1}^{s-1}\left[\sum_{m=r}^{n-5}(n-m-5)(n-7)! + \sum_{m=r}^{n-5}A_{n-3}(m-1)+A_{n-3}(r-1)\right] \\
&+\sum_{k=1}^{n-1}\sum_{l=k}^{n-1}\sum_{s=1}^{l-1}\sum_{t=1}^{s-1}\left[\sum_{r=t}^{n-4}(n-t-4)(n-6)! + \sum_{r=t}^{n-4}A_{n-3}(r-1)+A_{n-4}(t-1)\right] + \sum_{k=1}^{n-1}\sum_{l=k}^{n-1}(n-l-1)(n-3)! \\
&+\sum_{k=1}^{n-1}\sum_{s=1}^{k}\sum_{t=s}^{n-5}(n-t-2)(n-4)! + \sum_{k=1}^{n-1}\sum_{s=1}^{k}\sum_{t=s}^{n-5}\sum_{r=t}^{s-1}\left[\sum_{m=r}^{n-4}(n-m-4)(n-6)! + \sum_{m=r}^{n-4}A_{n-3}(m-1)+A_{n-4}(r-1)\right] \\
&+\sum_{k=1}^{n-1}\sum_{s=1}^{k}\sum_{t=1}^{s-1}\left[\sum_{r=t}^{n-2}(n-r-3)(n-5)! + \sum_{r=t}^{n-2}A_{n-4}(r-1)+A_{n-2}(t-1)\right]
\end{aligned}
$$

2. avoid213 的結果：

Kn=

$$
\begin{bmatrix}
B(n-1,1,n-1) & (n-2)\times(n-3)!+B(n-1,1,0) & (n-3)(n-3)!+B(n-1,1,1) & \dots & 0(n-3)!+B(n-1,1,n-2) \\
B(n-1,2,n-2) & B(n-1,2,n-2) & (n-3)(n-3)!+B(n-1,2,1) & \dots & \vdots \\
\vdots & \vdots & \ddots & \ddots & \vdots \\
B(n-1,n-1,n-2) & B(n-1,n-1,n-2) & \dots & \dots & 0\times(n-3)!+B(n-1,n-1,n-2)
\end{bmatrix}
$$

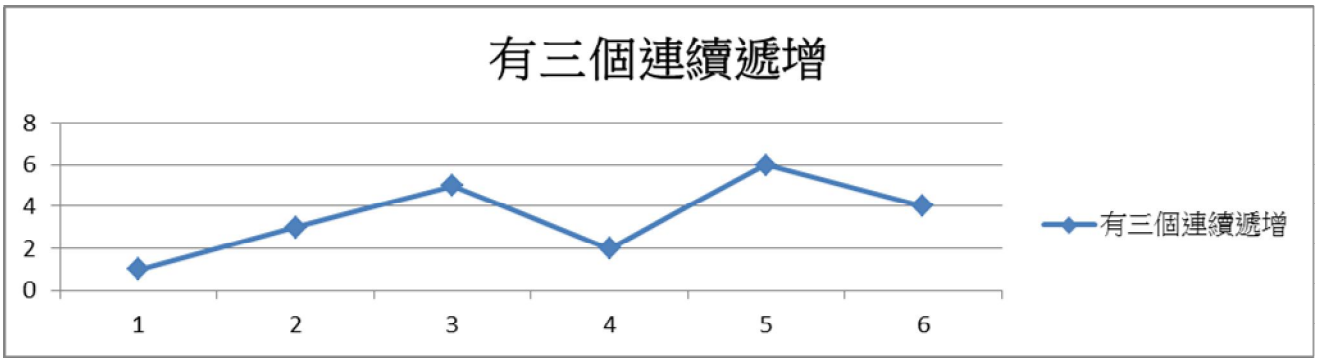
其中 B(j，k，u)=(第 j 個矩陣中，第 k 行，第 k 行的前 u 項)

一、　研究動機：

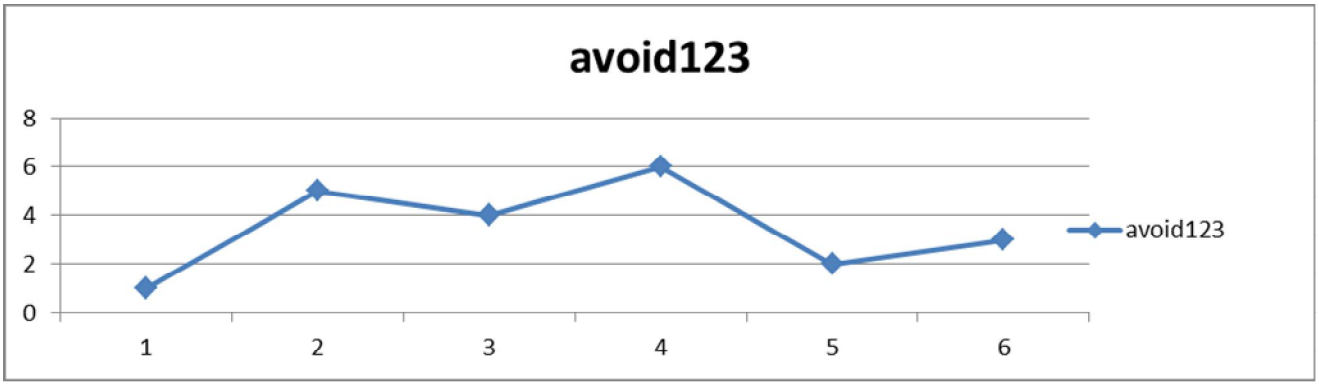
我們這個題目的來源是某一天老師在專題課時突然提出來的突擊問題，一開始我們只是把它當成專題來研究，然而隨著不斷的討論與失敗讓我們對這個題目越來越有興趣，並決定正式把這個當成我們的科展題目。後來又去上網查了一些資料(見附錄)，卻查到了很多外國文獻然而外國文獻和研究中都並未對這個問題有一個很好的處理方法(例如只有 n<=3 才能使用的公式)，抑或者是只有研究特定情況下的 avoid。例如：(avoid123 且 avoid132)或(只有一個 123 且 avoid312)等等同時有兩個條件的例子，感覺到此問題富有挑戰性，並期待可以找到一個更加適當的方法去全面地處理這個問題。

二、　研究問題的定義：

將 1~n 等 n 個正整數做直線排列，**其中任連續三個數 a_1、a_2、a_3必不符合 $a_1<a_2<a_3$的排列稱作 avoid123** 例如：135264 因為其中 1<3<5 固不符合 avoid123 而 154623 任三個數字無連續遞增，因此符合 avoid123。



上述圖中有連續三個遞增發生



上述圖中則沒有

三、　研究方法：

1.　研究過程：

一開始嘗試逐一列舉，並分出符合與不符合的數量。由於正面計算過於複雜，因此我們用反面作法計算有連續三個遞增的情形(也就是要被 aviod 掉的數量)但發現當**數量大時，此方法難以計算。**

我們第二個想到的方法就是用排容原理來處理，考慮每個由小到大的連續三個數字，將其填入 n 個位置，利用邏輯集合的方式計算不符合的個數，然而，嘗試一些數字以後，發現仍無法克服大數字時的龐大計算量。

我們嘗試的第三個方法是考慮上升下降的趨勢，由遞增遞減的方法去分組記數，計算到 6 個數字時勉強可以計算，數量大時亦無法處裡。第三個方法是考慮最大數字的位置，再填入剩下的數字。當最大數字填入的位置為前幾個位置時尚算好算，一旦排到第 5 個或是第 6 個時也是相當難以記數，因此此方法也是行不通的。

2.　分類處理：

最後我們考慮第一個數字，發現前兩項有關係，再進一步考慮第二個數字，並將其寫成表格形式。詳述如下：

以 n=5 為例，考慮前兩個數字分成 20 組討論

12xxx	21xxx	31xxx	41xxx	51xxx
13xxx	23xxx	32xxx	42xxx	52xxx
14xxx	24xxx	34xxx	43xxx	53xxx
15xxx	25xxx	35xxx	45xxx	54xxx

其中

12xxx：需要再排 345，而有連續三個遞增(也就是說沒有 avoid123 的情形)的排法有 6 種

13xxx：需要再排 245，而有連續三個遞增的排法有 5 種，

分別是 13245，13425，13452，13524，13542

14xxx：需要再排 235，而有連續三個遞增的排法有 3 種，

分別是 14235，14523，14532…

54xxx：需要再排 123，而有連續三個遞增的排法有 1 種，是 54123

將每一組的排法數量定義成一個 4x5 矩陣如下：

$$T_5=\begin{bmatrix} 6 & 3 & 3 & 3 & 3 \\ 5 & 5 & 2 & 2 & 2 \\ 3 & 3 & 3 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

設 $A_n(k)$為矩陣 T_n 前 k 行數字元素總和，$A_n(n)$則為 n 個數字時符合連續三個遞增。

定義第 1 行數字總和為 $A_5(1)=15$， 前 2 行數字總和為 $A_5(2)=27$…前 5 行數字總和為 $A_5(5)=50$

現在我們以 n 個數字為例，考慮前兩個數字分成(n-1)×n 組來討論：

$$\begin{array}{ccccc} 12\ xxx & 21\ xxx & \dots & \dots & n1\ xxx \\ 13\ xxx & 23\ xxx & \dots & \dots & n2\ xxx \\ 14\ xxx & 24\ xxx & \dots & \dots & n3\ xxx \\ \dots & \dots & \dots & \dots & \dots \\ 1nxxx & 2nxxx & 3nxxx & \dots & n(n-1)xxx \end{array}$$

如上圖所示，可將此矩陣中元素表示為 abxxx 並分為兩個三角形。

其中左下三角形都符合 a<b，相反的，右上三角形皆符合 a>b

3. 矩陣建立：

將這(n-1)×n 組符合連續三個遞增的排列數量寫成矩陣 T_n，其中我們規定 $A_n(0)=0$。

根據我們的研究得出結論為：

$$T_n=\begin{bmatrix} A_{n-2}(0)+(n-2)(n-3)! & A_{n-1}(1)-A_{n-1}(0) & A_{n-1}(1)-A_{n-1}(0) & \dots & \dots & A_{n-1}(1)-A_{n-1}(0) \\ A_{n-2}(1)+(n-3)(n-3)! & A_{n-2}(1)+(n-3)(n-3)! & \ddots & \dots & \dots & A_{n-1}(2)-A_{n-1}(1) \\ \vdots & \dots & \ddots & \ddots & \dots & \vdots \\ \vdots & \dots & \dots & \ddots & \ddots & \vdots \\ A_{n-2}(n-2)+0(n-3)! & \dots & \dots & \dots & A_{n-2}(n-2)+0(n-3)! & A_{n-1}(n-1)-A_{n-1}(n-2) \end{bmatrix}$$

T_n 中，左下的三角形的說明：

我們現在令每組前面三個數字分別是 a，b，c。對於左下三角形中必為 a<b。我們接下來要去考慮的就是第三個數字 c 的大小，若是 c 也大於 b 的話就代表著在前三個數字當中就出現了連續三個遞增的數字，剩下的 n-3 個數字也可以任意排列。也就是說若有 k 個 c 滿足 a<b<c 則其排法為 k×(n-3)!。

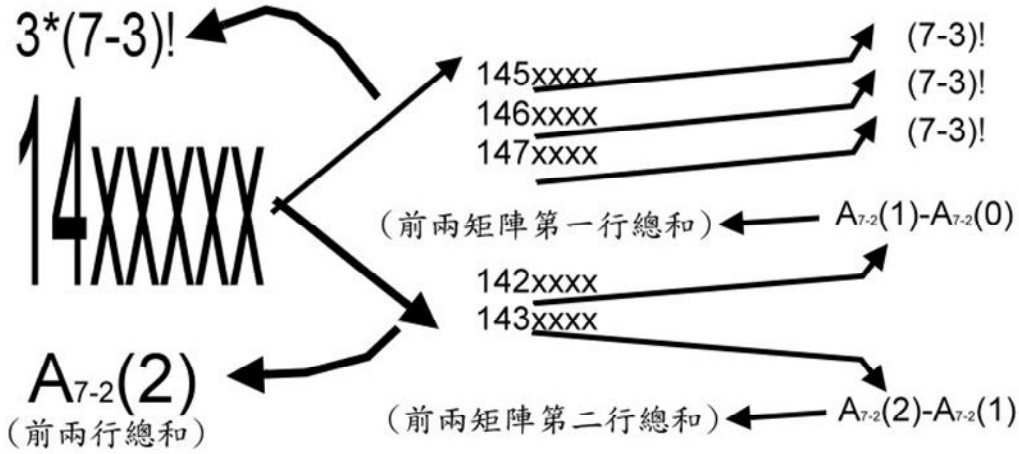
相對的，若是 c<b 的話，前三個數字必不存在連續三個遞增的情形，若要有連續三個遞增的情形發生必存在在於後面的 n-2 個數字當中(說明：也就是說從第三個數字重新開始排列)在剩下的 n-2 個數字當中重新作小到大的排列，就會出現和前兩項的相同的情形用這樣的方法就可以用遞迴的方式寫出上述的公式。)

T_n 中，右上的三角形的說明：

在右上三角形當中，必存在 a>b，因此在前面三個數字當中必定不可能會出現連續三個遞增的數字，這也就是說若存在連續遞增的三個數字必定存在在剩下的 n-1 個數字當中。所以我們只要考慮剩下的 n-1 個數字，將其由小到大做排列，即可使用前一項來寫出遞迴關係式。

舉例說明：

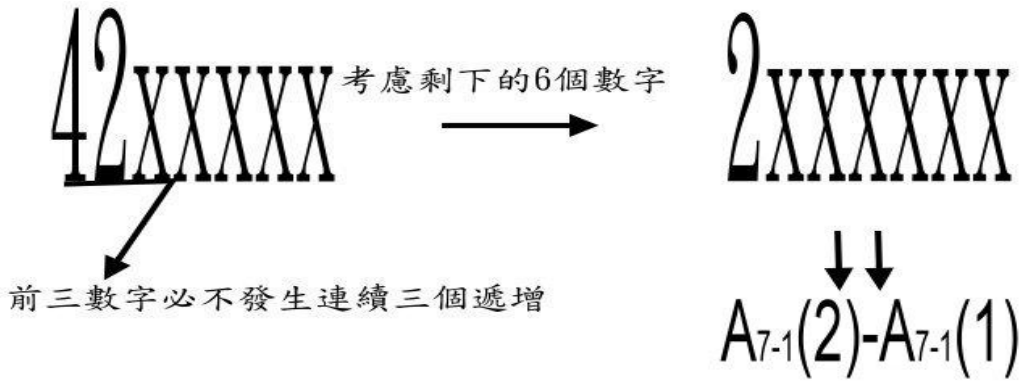
以 n=7 時，考慮左下三角形中 14xxxxx 的情況：



若第三個位置(c 的位置)排的數字是大於 4 的時候(5，6，7 共計 3 個)剩下四個數字及可以任意排列，也就是 4!。因此這部份數量就是 3×(7-3)!

若 c 的位置是小 4 的情況(2，3 共計兩個數字)則前三個數字中必不發生連續三個遞增情形，要發生連續三個遞增情形將出現在後 5 個數字中(包含第三個數字)。將剩下數字坐小到大的排列(2，3，5，6，7)在排列這些數字時可以將其看成 T_5 的情形。2 排在第三個位置時即代表著 T_5 中的第一行數值總和。3 排在第三個位置時即代表著 T_5 中的第二行數值總和。有此方法可以得出 14xxxxx 的情況中數量為 $A_5(2)+3\times4!$，其餘左下三角形中的所有數值皆可以使用此方法加以處理即可得到左下三角形中的所有數值。

以 n=7 時，考慮右上三角形中 42xxxxx 的情況：



因為前三個數字中必不會發生連續三個遞增的情形發生，因此若有連續三個遞增的情形發生必只發生在剩下六個數當中(包含第二個數字)接下來將剩下的六個數字由小到大做排列(1，2，3，5，6，7)，在排列剩下數字時，即可將其視為 T6 中的情形來計算。因為 2 在此六個數字當中是第二小的數字，因此就相當於第二行的數值總和 A6(2)- A6(1)，其餘右上三角形中的所有數值皆可以使用此方法加以處理即可得到右上三角形中的所有數值。

綜合以上兩部分的處理技巧即可得到

$$T_n=\begin{bmatrix} A_{n-2}(0)+(n-2)(n-3)! & A_{n-1}(1)-A_{n-1}(0) & A_{n-1}(1)-A_{n-1}(0) & \cdots & \cdots & A_{n-1}(1)-A_{n-1}(0) \\ A_{n-2}(1)+(n-3)(n-3)! & A_{n-2}(1)+(n-3)(n-3)! & \ddots & \cdots & \cdots & A_{n-1}(2)-A_{n-1}(1) \\ \vdots & \cdots & \ddots & \ddots & \cdots & \vdots \\ \vdots & \cdots & \cdots & \ddots & \ddots & \vdots \\ A_{n-2}(n-2)+0(n-3)! & \cdots & \cdots & \cdots & A_{n-2}(n-2)+0(n-3)! & A_{n-1}(n-1)-A_{n-1}(n-2) \end{bmatrix}$$

其中 $A_n(n)$為矩陣中所有元素的總合

四、　研究結果：

1. 矩陣：

$$T_n=\begin{bmatrix} A_{n-2}(0)+(n-2)(n-3)! & A_{n-1}(1)-A_{n-1}(0) & A_{n-1}(1)-A_{n-1}(0) & \cdots & \cdots & A_{n-1}(1)-A_{n-1}(0) \\ A_{n-2}(1)+(n-3)(n-3)! & A_{n-2}(1)+(n-3)(n-3)! & \ddots & \cdots & \cdots & A_{n-1}(2)-A_{n-1}(1) \\ \vdots & \cdots & \ddots & \ddots & \cdots & \vdots \\ \vdots & \cdots & \cdots & \ddots & \ddots & \vdots \\ A_{n-2}(n-2)+0(n-3)! & \cdots & \cdots & \cdots & A_{n-2}(n-2)+0(n-3)! & A_{n-1}(n-1)-A_{n-1}(n-2) \end{bmatrix}$$

4. 化簡：

我們將其用 sigma 表示並化簡，化簡如下：

$$A_n(k)=\sum_{s=1}^{k}\left[\sum_{t=s}^{n-1}(n-t-1)(n-3)!+\sum_{t=s}^{n-1}A_{n-2}(t-1)+A_{n-1}(s-1)\right]$$

$$A_n(n)=\sum_{k=1}^{n-1}\sum_{l=k}^{n-1}A_{n-2}(l-1)+\sum_{k=1}^{n-1}\sum_{l=k}^{n-1}(n-l-1)(n-3)!+\sum_{k=1}^{n-1}A_{n-1}(k)$$

$$A_n(n)=\sum_{k=1}^{n-1}\sum_{l=k}^{n-1}\sum_{s=1}^{l-1}\sum_{t=s}^{n-3}(n-t-3)(n-5)!+\sum_{k=1}^{n-1}\sum_{l=k}^{n-1}\sum_{s=1}^{l-1}\sum_{t=s}^{n-3}A_{n-4}(t-1)+\sum_{k=1}^{n-1}\sum_{l=k}^{n-1}\sum_{s=1}^{l-1}A_{n-3}(s-1)+\sum_{k=1}^{n-1}\sum_{l=k}^{n-1}(n-l-1)(n-3)!$$
$$+\sum_{k=1}^{n-1}\sum_{s=1}^{k}\sum_{t=s}^{n-2}(n-t-2)(n-4)!+\sum_{k=1}^{n-1}\sum_{s=1}^{k}\sum_{t=s}^{n-2}A_{n-3}(t-1)+\sum_{k=1}^{n-1}\sum_{s=1}^{k}A_{n-2}(s-1)$$

$$A_n(n)=\sum_{k=1}^{n-1}\sum_{l=k}^{n-1}\sum_{s=1}^{l-1}\sum_{t=s}^{n-3}(n-t-3)(n-5)!+\sum_{k=1}^{n-1}\sum_{l=k}^{n-1}\sum_{s=1}^{l-1}\sum_{t=s}^{n-3}\sum_{r=1}^{t-1}\left[\sum_{m=r}^{n-5}(n-m-5)(n-7)!+\sum_{m=r}^{n-5}A_{n-6}(m-1)+A_{n-5}(r-1)\right]$$
$$+\sum_{k=1}^{n-1}\sum_{l=k}^{n-1}\sum_{s=1}^{l-1}\sum_{t=1}^{s-1}\left[\sum_{r=t}^{n-4}(n-t-4)(n-6)!+\sum_{r=t}^{n-4}A_{n-5}(r-1)+A_{n-4}(t-1)\right]+\sum_{k=1}^{n-1}\sum_{l=k}^{n-1}(n-l-1)(n-3)!$$
$$+\sum_{k=1}^{n-1}\sum_{s=1}^{k}\sum_{t=s}^{n-2}(n-t-2)(n-4)!$$
$$+\sum_{k=1}^{n-1}\sum_{s=1}^{k}\sum_{t=s}^{n-2}\sum_{r=t}^{t-1}\left[\sum_{m=r}^{n-4}(n-m-4)(n-6)!+\sum_{m=r}^{n-4}A_{n-5}(m-1)+A_{n-4}(r-1)\right]$$
$$+\sum_{k=1}^{n-1}\sum_{s=1}^{k}\sum_{t=1}^{s-1}\left[\sum_{r=t}^{n-3}(n-r-3)(n-5)!+\sum_{r=t}^{n-3}A_{n-4}(r-1)+A_{n-3}(t-1)\right]$$

$$\sum_{k=1}^{n-1}k=C_2^n$$

$$\sum_{k=1}^{n-1}\sum_{l=1}^{k-1}l=C_2^n$$

$$\sum_{k=1}^{n-1}\sum_{l=k}^{n-1}(n-l-1)(n-3)!=(n-3)!\left[\sum_{k=1}^{n-1}\sum_{l=k}^{n-1}(n-1)-\sum_{k=1}^{n-1}\sum_{l=k}^{n-1}l\right]$$
$$=(n-3)!\left[\sum_{k=1}^{n-1}\sum_{l=k}^{n-1}(n-1)-\sum_{k=1}^{n-1}\sum_{l=k}^{k-1}(n-1)-\sum_{k=1}^{n-1}\sum_{l=k}^{n-1}l+\sum_{k=1}^{n-1}\sum_{l=k}^{k-1}l\right]$$
$$=(n-3)![(n-1)^2-(n-1)C_2^n+(n-1)^2-(n-1)C_2^n+C_2^n]=(n-3)!(n-1)\frac{n(n-2)}{6}=\frac{n!}{6}$$

同理，可知

$$\sum_{k=1}^{n-1}\sum_{s=1}^{k}\sum_{t=s}^{n-2}(n-t-2)(n-4)!=\frac{n!}{24}$$

$$\sum_{k=1}^{n-1}\sum_{l=k}^{n-1}\sum_{s=1}^{l-1}\sum_{t=s}^{n-3}(n-t-3)(n-5)!=\frac{n!}{120}$$

將 $A_n(n)$用相同的方法不斷帶下去，每帶一次將會讓底標中的數字變小(n → (n-1) or (n-2)) 在我們的第二個矩陣當中 $T_2=\begin{bmatrix}0 & 0\end{bmatrix}$，其中 $A_2(0)=A_2(1)=A_2(2)=0$

只要 $A_n(n)$在多次代換以後產生之 sigma 中出現 $A_2(k)$ (k 為 0，1，2)其值必為 0，另外為了計算方便我們訂 $A_1(1)=0$ $A_1(0)=0$(這樣就可以使 $A_3(k)$都符合矩陣的運算)也就是說我們要計算的部分，就會只剩下 sigma 中不含 $A_j(k)$的部分。由此可以化簡 sigma。

2. 定理：

定理一：T_n 最後一列皆為同一數字

proof：左下三角形和右上三角形中的每一列皆相等只需要證明第一行和最後一行最後一數字是相同的就可以了。

第一行最後一個數值為 $A_{n-2}(n-2)+0\times(n-3)!=A_{n-2}(n-2)$=前兩矩陣中的全部總和

最後一行的最後一個數值為 $A_{n-1}(n-1)-A_{n-1}(n-2)$=前一矩陣中最後一行的數值和

在前兩矩陣中，共有 n-2 個數字。在前一矩陣中，有 n-1 個數字，最後一行的數值總和代表的意義是第一個數字排 n-1。因為 n-1 在前一矩陣當中是最大的數字，也就是說只要考慮剩下的 n-2 個數字就可以了。因此 $A_{n-1}(n-1)-A_{n-1}(n-2)=A_{n-2}(n-2)$。

以上，可證得最後一列為同一數字。

定理二：T_n 末兩行完全一樣

proof：因為末兩行當中前 n-2 項根據右上三角形每列都相等的性質，兩行唯一可　能出現不一樣的情況必發生在最後的數值。又根據定理一可以得知

$A_{n-1}(n-1)-A_{n-1}(n-2)=A_{n-2}(n-2)$。因此最後數值也都相等，所以可以得知 T_n 末兩行完全一樣

定理三：T_n 第一行與最末行有遞減性

proof：在第一行中，考慮上面項與下面一項的情況，因為下面一項的所有排列情況，可以用上一項的部份情況來表示。也就代表著下面一項的數量必小於上面一項的數量。(舉例：12xxx 與 13xxx 中，只要將 2 與 3 掉換 13xxx 中所有數量都可以用 12xxx 的情況來表示，因此 13xxx 的數量必小於 12xxx 的數量。51xxx 和 52xxx 的情況類似，將 1、2 互換也可以得到上述結果)

定理四：每一列有遞減性

proof：根據定理四，得知最後一行有遞減性，因此可以得到每一列中會出現遞減情形

定理五：$A_n(n)=A_{n+1}(n+1)-A_{n+1}(n)=T_{n+2}$ 中之最小值

proof：綜合定理三和定理四可以自然得到此結論。

定理六：$(n+1)(n+2)A_n(n)<A_{n+2}(n+2)$

proof:在 T_{n+2} 矩陣中共有(n+1)(n+2)個數字，且根據定理四得知 $A_n(n)$為 T_{n+2} 中之最小值，也就是說 $A_n(n)$小於矩陣中的每一個數字，因此(n+1)(n+2)個 $A_n(n)$必定會小於矩陣中的所有數字和。

定理七： 當 n>=4 $A_n(2)=A_n(1)+\frac{An(3)}{3}$ 且 $A_n(3)$為 3 的倍數

proof：

$[A_{n-1}(1)-A_{n-1}(0)]-[A_{n-1}(2)-A_{n-1}(1)]$

$=[A_{n-3}(0)+(n-3)\times(n-4)!]-[A_{n-2}(1)-A_{n-2}(0)]$

$=[A_{n-2}(0)+(n-3)!]-[A_{n-2}(1)]$

$=[A_{n-2}(0)+(n-2)\times(n-3)!]-[A_{n-2}(1)+(n-3)\times(n-3)!]$

$\therefore[A_{n-2}(0)+(n-2)\times(n-3)!]-[A_{n-1}(1)-A_{n-1}(0)]$

$=[A_{n-2}(1)+(n-3)\times(n-3)!]-[A_{n-1}(2)-A_{n-1}(1)]$

$\therefore[A_n(2)-A_n(1)]-[A_n(1)-A_n(0)]=[A_n(3)-A_n(2)]-[A_n(2)-A_n(1)]$

$\therefore 3A_n(2)=3A_n(1)+A_n(3)$

$\therefore An(2)=A_n(1)+\frac{An(3)}{3}$

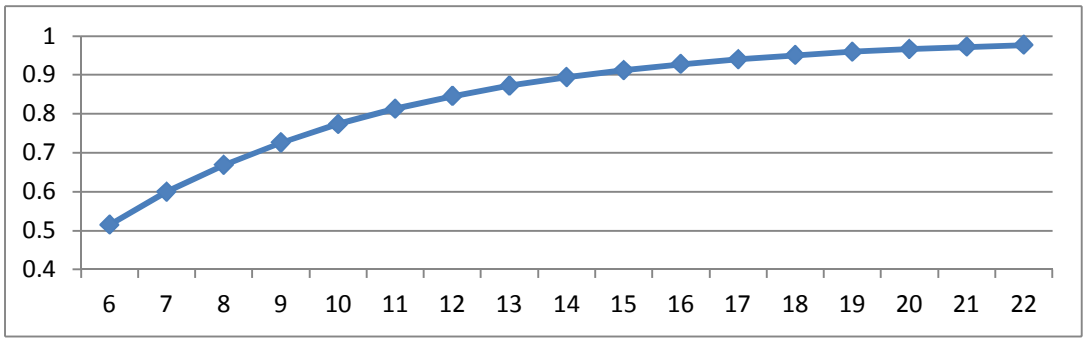
同時證明了 $A_n(3)$為 3 的倍數

若 n≦3 則(n-4)!在討論個數時無意義

∴n 必須≧4

3. 猜想：An(n)和 n!的比值將會愈來愈靠近 1

我們將 n 和 $A_n(n)/n!$作圖後可以得到下面的圖形。



根據以上圖形，可以發現，$A_n(n)$和 n!的比值將隨著 n 的增加而趨近。其中當 n>=6 時比值將大於 1/2。

由此我們觀察其遞增情形，發現比值將會愈來愈靠近 1

五、延伸問題：avoid213

1。 定義：**在 n 個數字當中不可在連續三個數字當中出現中小大的排列數量。**例如：<u>315</u>24 在這情況下即是有出現中小大的連續三個數字排列，就必須將這種情況排除，12345 則沒有任何的連續三個數字產生中小大的排列，也就不需要排除此種情況。

2。 矩陣建立

和上述的 avoid123 問題一樣用相同的邏輯去討論其與前一項的關係。由於在 avoid213 和 avoid123 一樣正面去處裡這個問題將會相當困難，於是我們一樣**只考慮將被刪除的（符合中小大）情形。**

沿用 avoid123 時做一樣的分類。考慮前兩個數字為何並依序將其數量寫成表格形式：

$$\begin{array}{ccccc} 12xxx & 21xxx & 31xxx & \cdots & n1xxx \\ 13xxx & 23xxx & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 1nxxx & 2nxxx & \cdots & \cdots & n(n-1)xxx \end{array}$$

其中 B(j，k，u)=(第 j 個矩陣中，第 k 行，第 k 行的前 u 項)

定義 B(j，k，0)=0

根據這樣的討論方法我們可以得出

Kn=

$$\begin{bmatrix} B(n-1,1,n-1) & (n-2)\times(n-3)!+B(n-1,1,0) & (n-3)(n-3)!+B(n-1,1,1) & \cdots & 0(n-3)!+B(n-1,1,n-2) \\ B(n-1,2,n-2) & B(n-1,2,n-2) & (n-3)(n-3)!+B(n-1,2,1) & \cdots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ B(n-1,n-1,n-2) & B(n-1,n-1,n-2) & \cdots & \cdots & 0\times(n-3)!+B(n-1,n-1,n-2) \end{bmatrix}$$

分成左下三角形和右上三角型來討論，並規定前三個數字分別為 a，b，c。

在左下的三角形當中我們可以看到 a<b 而在現在情況底下我們所要的排列是出現連續三個小中大的數字排列。再 a<b 的情況下永遠不可能在前面三個數字發生這種情形，因此可以將第一個數字刪除，並考慮 b 在剩下數字當中的大小，寫出遞回關係式。

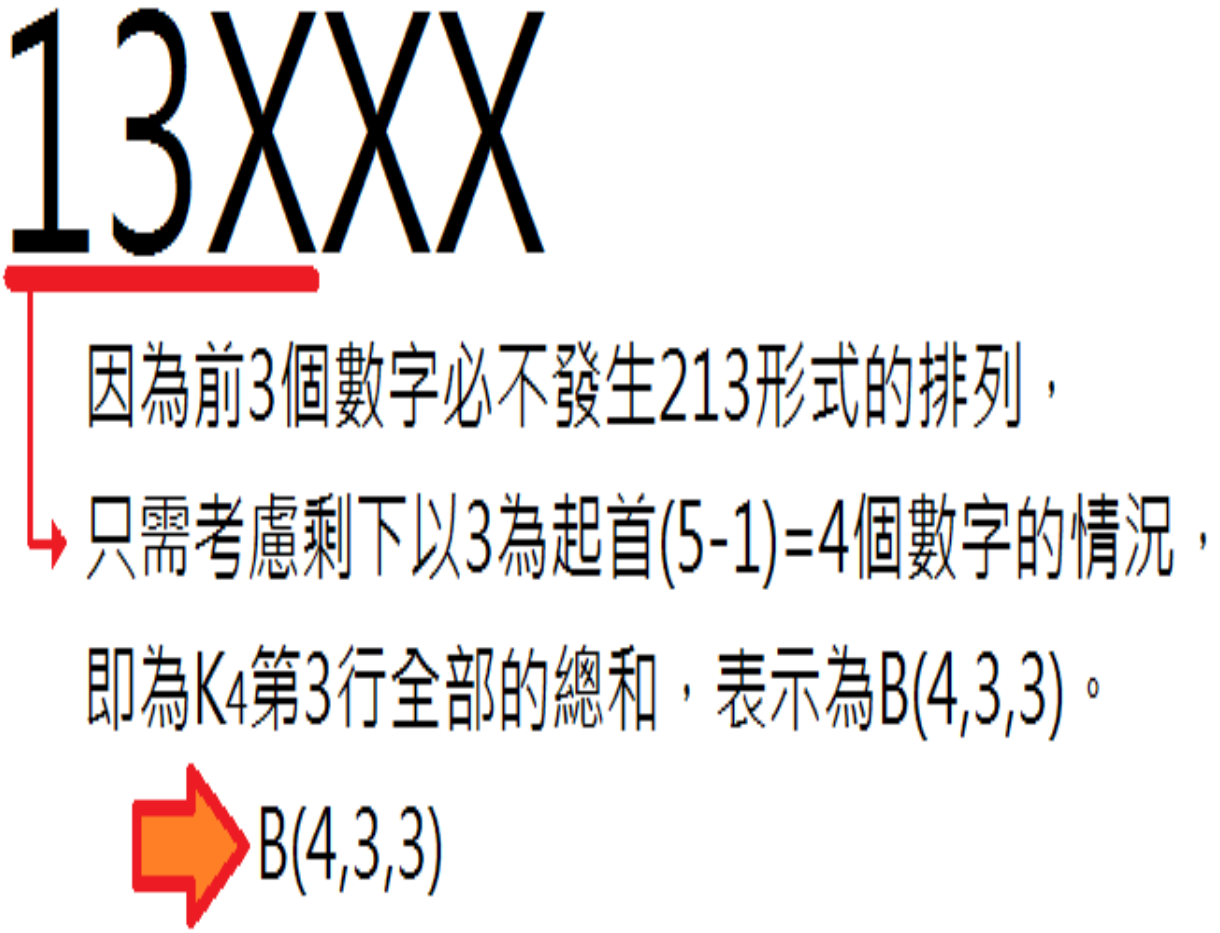
在右上三角形當中我們可以看到 a>b 在這種情況下。在此種情況底下只要若是 c>a>b 即是在前三個數字就發生了連續三個中小大的情形。剩下的(n-3)個數字就可以隨意排列其排法數量為(n-3)!

在 21xxx 的情況底下有(n-2)個 c(3，4，5，…，n)可以符合。在 31xxx 的情況底下有(n-3)個數字可以符合。以此類推就可以計算出每一個 abxxx(以 ab 為起首的 n 個數字排列)時 c>a>b 的情形的所有數量。

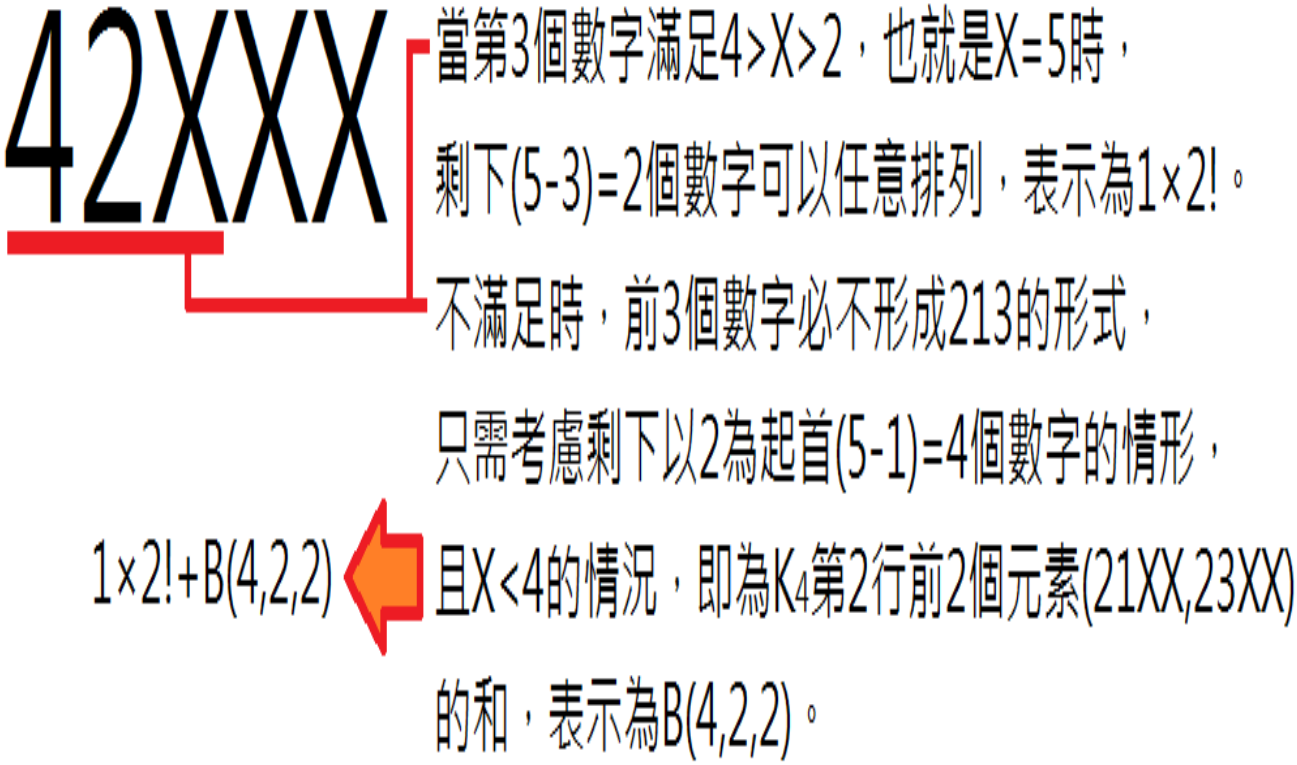
至於 c<a 或 c<b 的情況時，就不可能在前面三個數字就發生中小大的排列。也就是說可以忽略第一個數字 a。但並不能夠保證剩下的數字中並不會發生中小大的情形發生，也就是說必須進一步考慮剩下未被忽略的兩個數字 b，c。(這是和 avoid123 所不一樣的，avoid123 再將前一個或兩個數字刪除以後，就只需考慮一個數字即可)在 41xxx 的情況中，c=2，3 時可以忽略 4，考慮剩下 n-1 個數字就可以將其表示成 B(n-1，1，2)

舉例說明：

左下三角形：



右上三角形：



3. 比較：

另外我們發現那六種情況中(avoid213，avoid132，avoid231，avoid312， avoid321，avoid123)avoid321 的數量會和 avoid123 一樣多。 avoid132，avoid231，avoid312，avoid213 這四種的數量一樣多

假設有一組數字 abc…def…ghi 不符合 avoid123，即其中有連續三數 def 同時符合 d<e 且 e<f 那我們將這組數字左右相反做重新排列，即 ihg…fed…cba，其中 f>e 且 e>d，所以這組重新排出來的數列必不符合 avoid321，以上證明每組不符合 avoid123 必可以和一組 avoid321 做對應，相反的，每組不符合 avoid321 也必可以和一組 avoid123 做對應，故，不符合 avoid123 和不符合 avoid321 一樣多，**即 avoid123 的數量=avoid321 的數量**

同理 avoid132 的數量=avoid231 的數量 ， avoid213 的數量=avoid312 的數量

另外假設有一組數字 abc…def…ghi 不符合 avoid132，即其中有連續三數 def 符合 d<f<e，我們一樣對這個數列做重新排列，但這次是最小的數字和最大的數字換，第二小的和第二大的換…依此類推，則這組數字變成(n+1-a) (n+1-b) (n+1-c)…(n+1-d) (n+1-e) (n+1-f)…(n+1-g) (n+1-h) (n+1-i)

由於 d<f<e 所以-d>-f>-i 所以(n+1-d) >(n+1-f)> (n+1-e)，即重新排列後的數字不符合 avoid312

由此證明 avoid132 亦可與 avoid312 做一一對應，所以

avoid213 的數量=[avoid312 的數量= avoid132 的數量]=avoid231 的數量

六、結論：

1. 找到一個演算法來處裡 avoid123，對於所有一個已知的 n 可以計算出其數量。
2. 對於建立的矩陣發現一些有趣性質，並希望藉這些性質繼續推廣。
3. 對於 avoid213 也用相同類似的方法研究得出結果
4. 對於化簡 sigma 過程當中遇到很多困難，針對這個部分仍持續努力試圖找出一個更簡潔的通式

七、未來展望：

1. 藉由化簡 sigma 試圖尋找 $A_n(n)$更簡單的通式

八、參考文獻：

1. Permutations containing and avoiding certain patterns - Toufik Mansour - Department of Mathematics， Haifa University
2. Permutations Containing and Avoiding 123 and 132 Patterns - Aaron Robertson - Department of Mathematics， Temple University
3. RESTRICTED PERMUTATIONS FROM CATALAN TO FINE AND BACK - Aaron Robertson - Department of Mathematics， Colgate University
4. Pattern avoidance in compositions and multiset permutations - Carla D。 Savage & Herbert S。 Wilf - North Carolina State University & University of Pennsylvania
5. Restricted 132-Dumont permutations - Toufik Mansour - Department of Mathematics， Haifa University
6. Restricted Motzkin permutations， Motzkin paths， continued fractions， and Chebyshev polynomials - Sergi Elizalde & Toufik Mansour - Department of Mathematics， MIT， Cambridge & Department of Mathematics， Haifa University
7. PERMUTATIONS WITH RESTRICTED PATTERNS AND DYCK PATHS - C。 Krattenthaler - Institut f˙ur Mathematik der Universit¨at Wien， Strudlhofgasse 4， A-1090 Wien， Austria。

附錄一：

$T_2=\begin{bmatrix}0 & 0\end{bmatrix}$

$T_3=\begin{bmatrix}1 & 0 & 0\\ 0 & 0 & 0\end{bmatrix}$

$T_4=\begin{bmatrix}2 & 1 & 1 & 1\\ 1 & 1 & 0 & 0\\ 0 & 0 & 0 & 0\end{bmatrix}$

$T_5=\begin{bmatrix}6 & 3 & 3 & 3 & 3\\ 5 & 5 & 2 & 2 & 2\\ 3 & 3 & 3 & 1 & 1\\ 1 & 1 & 1 & 1 & 1\end{bmatrix}$

$T_6=\begin{bmatrix}24 & 15 & 15 & 15 & 15 & 15\\ 21 & 21 & 12 & 12 & 12 & 12\\ 17 & 17 & 17 & 9 & 9 & 9\\ 12 & 12 & 12 & 12 & 7 & 7\\ 7 & 7 & 7 & 7 & 7 & 7\end{bmatrix}$

$T_7=\begin{bmatrix}120 & 81 & 81 & 81 & 81 & 81 & 81\\ 111 & 111 & 72 & 72 & 72 & 72 & 72\\ 99 & 99 & 99 & 63 & 63 & 63 & 63\\ 84 & 84 & 84 & 84 & 55 & 55 & 55\\ 67 & 67 & 67 & 67 & 67 & 50 & 50\\ 50 & 50 & 50 & 50 & 50 & 50 & 50\end{bmatrix}$

$T_8=\begin{bmatrix}720 & 531 & 531 & 531 & 531 & 531 & 531 & 531\\ 681 & 681 & 492 & 492 & 492 & 492 & 492 & 492\\ 633 & 633 & 633 & 453 & 453 & 453 & 453 & 453\\ 576 & 576 & 576 & 576 & 417 & 417 & 417 & 417\\ 511 & 511 & 511 & 511 & 511 & 388 & 388 & 388\\ 441 & 441 & 441 & 441 & 441 & 441 & 371 & 371\\ 371 & 371 & 371 & 371 & 371 & 371 & 371 & 371\end{bmatrix}$

$$T_9=\begin{bmatrix} 5040 & 3933 & 3933 & 3933 & 3933 & 3933 & 3933 & 3933 & 3933 \\ 4851 & 4851 & 3744 & 3744 & 3744 & 3744 & 3744 & 3744 & 3744 \\ 4623 & 4623 & 4623 & 3555 & 3555 & 3555 & 3555 & 3555 & 3555 \\ 4356 & 4356 & 4356 & 4356 & 3375 & 3375 & 3375 & 3375 & 3375 \\ 4053 & 4053 & 4053 & 4053 & 4053 & 3216 & 3216 & 3216 & 3216 \\ 3721 & 3721 & 3721 & 3721 & 3721 & 3721 & 3093 & 3093 & 3093 \\ 3372 & 3372 & 3372 & 3372 & 3372 & 3372 & 3372 & 3023 & 3023 \\ 3023 & 3023 & 3023 & 3023 & 3023 & 3023 & 3023 & 3023 & 3023 \end{bmatrix}$$

$$T_{10}=\begin{bmatrix} 40320 & 33039 & 33039 & 33039 & 33039 & 33039 & 33039 & 33039 & 33039 & 33039 \\ 39213 & 39213 & 31932 & 31932 & 31932 & 31932 & 31932 & 31932 & 31932 & 31932 \\ 37917 & 37917 & 37917 & 30825 & 30825 & 30825 & 30825 & 30825 & 30825 & 30825 \\ 36432 & 36432 & 36432 & 36432 & 29757 & 29757 & 29757 & 29757 & 29757 & 29757 \\ 34767 & 34767 & 34767 & 34767 & 34767 & 28776 & 28776 & 28776 & 28776 & 28776 \\ 32943 & 32943 & 32943 & 32943 & 32943 & 32943 & 27939 & 27939 & 27939 & 27939 \\ 30996 & 30996 & 30996 & 30996 & 30996 & 30996 & 30996 & 27311 & 27311 & 27311 \\ 28979 & 28979 & 28979 & 28979 & 28979 & 28979 & 28979 & 28979 & 26962 & 26962 \\ 26962 & 26962 & 26962 & 26962 & 26962 & 26962 & 26962 & 26962 & 26962 & 26962 \end{bmatrix}$$

$$T_{13}=\begin{bmatrix} 39916800 & 35830917 & 35830917 & 35830917 & 35830917 & 35830917 & 35830917 & 35830917 & 35830917 & 35830917 & 35830917 & 35830917 & 35830917 \\ 39467979 & 39467979 & 35382096 & 35382096 & 35382096 & 35382096 & 35382096 & 35382096 & 35382096 & 35382096 & 35382096 & 35382096 & 35382096 \\ 38964807 & 38964807 & 38964807 & 34933275 & 34933275 & 34933275 & 34933275 & 34933275 & 34933275 & 34933275 & 34933275 & 34933275 & 34933275 \\ 38407284 & 38407284 & 38407284 & 38407284 & 34491735 & 34491735 & 34491735 & 34491735 & 34491735 & 34491735 & 34491735 & 34491735 & 34491735 \\ 37796517 & 37796517 & 37796517 & 37796517 & 37796517 & 34065864 & 34065864 & 34065864 & 34065864 & 34065864 & 34065864 & 34065864 & 34065864 \\ 37134909 & 37134909 & 37134909 & 37134909 & 37134909 & 37134909 & 33665157 & 33665157 & 33665157 & 33665157 & 33665157 & 33665157 & 33665157 \\ 36426348 & 36426348 & 36426348 & 36426348 & 36426348 & 36426348 & 36426348 & 33300177 & 33300177 & 33300177 & 33300177 & 33300177 & 33300177 \\ 35676387 & 35676387 & 35676387 & 35676387 & 35676387 & 35676387 & 35676387 & 35676387 & 32982468 & 32982468 & 32982468 & 32982468 & 32982468 \\ 34892403 & 34892403 & 34892403 & 34892403 & 34892403 & 34892403 & 34892403 & 34892403 & 34892403 & 32724411 & 32724411 & 32724411 & 32724411 \\ 34083720 & 34083720 & 34083720 & 34083720 & 34083720 & 34083720 & 34083720 & 34083720 & 34083720 & 34083720 & 32539015 & 32539015 & 32539015 \\ 33261679 & 33261679 & 33261679 & 33261679 & 33261679 & 33261679 & 33261679 & 33261679 & 33261679 & 33261679 & 33261679 & 32439638 & 32439638 \\ 32439638 & 32439638 & 32439638 & 32439638 & 32439638 & 32439638 & 32439638 & 32439638 & 32439638 & 32439638 & 32439638 & 32439638 & 32439638 \end{bmatrix}$$

$$T_{12}=\begin{bmatrix} 3628800 & 3179979 & 3179979 & 3179979 & 3179979 & 3179979 & 3179979 & 3179979 & 3179979 & 3179979 & 3179979 & 3179979 \\ 3574449 & 3574449 & 3125628 & 3125628 & 3125628 & 3125628 & 3125628 & 3125628 & 3125628 & 3125628 & 3125628 & 3125628 \\ 3512817 & 3512817 & 3512817 & 3071277 & 3071277 & 3071277 & 3071277 & 3071277 & 3071277 & 3071277 & 3071277 & 3071277 \\ 3443904 & 3443904 & 3443904 & 3443904 & 3018033 & 3018033 & 3018033 & 3018033 & 3018033 & 3018033 & 3018033 & 3018033 \\ 3367899 & 3367899 & 3367899 & 3367899 & 3367899 & 2967192 & 2967192 & 2967192 & 2967192 & 2967192 & 2967192 & 2967192 \\ 3285219 & 3285219 & 3285219 & 3285219 & 3285219 & 3285219 & 2920239 & 2920239 & 2920239 & 2920239 & 2920239 & 2920239 \\ 3196548 & 3196548 & 3196548 & 3196548 & 3196548 & 3196548 & 3196548 & 2878839 & 2878839 & 2878839 & 2878839 & 2878839 \\ 3102873 & 3102873 & 3102873 & 3102873 & 3102873 & 3102873 & 3102873 & 3102873 & 2844816 & 2844816 & 2844816 & 2844816 \\ 3005513 & 3005513 & 3005513 & 3005513 & 3005513 & 3005513 & 3005513 & 3005513 & 3005513 & 2820117 & 2820117 & 2820117 \\ 2906136 & 2906136 & 2906136 & 2906136 & 2906136 & 2906136 & 2906136 & 2906136 & 2906136 & 2906136 & 2806759 & 2806759 \\ 2806759 & 2806759 & 2806759 & 2806759 & 2806759 & 2806759 & 2806759 & 2806759 & 2806759 & 2806759 & 2806759 & 2806759 \end{bmatrix}$$

附錄二：

$K_2=\begin{bmatrix}0 & 0\end{bmatrix}$

$K_3=\begin{bmatrix}0 & 1 & 0\\ 0 & 0 & 0\end{bmatrix}$

$K_4=\begin{bmatrix}0 & 2 & 1 & 0\\ 1 & 1 & 2 & 1\\ 0 & 0 & 0 & 0\end{bmatrix}$

$K_5=\begin{bmatrix}1 & 6 & 4 & 3 & 1\\ 3 & 3 & 6 & 5 & 3\\ 3 & 3 & 3 & 5 & 3\\ 1 & 1 & 1 & 1 & 1\end{bmatrix}$

$K_6=\begin{bmatrix}8 & 24 & 19 & 16 & 13 & 8\\ 13 & 13 & 24 & 21 & 18 & 13\\ 14 & 14 & 14 & 22 & 19 & 14\\ 14 & 14 & 14 & 14 & 19 & 14\\ 8 & 8 & 8 & 8 & 8 & 8\end{bmatrix}$

附錄三：

```
/*Tn(資料庫)1。txt==Tn*/
/*Tn(資料庫)2。txt==An(k)*/
/*Tn(資料庫)1。txt==Tn*/
/*Tn(資料庫)2。txt==An(k)*/
#include <stdio。h>
#include <stdlib。h>
#include <string。h>
#define MAX 100
#define STA 6  /*start*/
#define ANL 20       /*lenth of an*/
struct deta{
        int nu[MAX];
        int len;
}an[3][ANL+1];

int plus(int， int*， int， int*， int*);
int sub(int， int*， int， int*， int*);
int mul(int， int*， int， int*， int*);
int lev(int， int*);
int tackle(char*， int*);/*字串轉數字*/
void clean(int*， int);  /*清空 int 陣列*/
int main()
{
int i， j， k， l， n， end， ln， l1;
```

```
int nn[MAX]， n1[MAX]， count[5];
struct deta tn1[ANL]， tn2[ANL];
char tn[100]， num[MAX*4];
FILE *ind1o， *ind2i， *ind2o;
while(printf("Tn(資料庫) 增 n 個>>") && scanf("%d"， &n)!=EOF){
 /*初始化*/
clean(nn， MAX-1);
clean(n1， MAX-1);
clean(count， 4);
/*input*/
ind2i = fopen("Tn(資料庫)2。txt"， "r");
printf("input\n");
i = STA;
while(!feof(ind2i)){
fflush(stdin);
fgets(tn， 100， ind2i);
for(j=1;j<=i;j++){
fscanf(ind2i， "%s"， num);
an[i%2][j]。len = tackle(num， an[i%2][j]。nu);
/**/
printf("%d"， an[i%2][j]。nu[an[i%2][j]。len]);
for(k=an[i%2][j]。len-1;k>=0;k--)
printf("%04d"， an[i%2][j]。nu[k]);
printf(" ");
}
printf("\n");
scanf(ind2i， "\n");
i++;
}
end = i-1;
if(n==1)
printf("n=%d\n"， end+1);
else
printf("n=%d~%d\n"， end+1， end+n);
fclose(ind2i);
/**/
for(i=1;i<=n;i++){
/*tn1[j]=A(end+i-2， j-1)+(end+i-j-1)*(end+i-3)!*/
ln = lev(end+i-3， nn);
for(j=1;j<=end+i-1;j++){
printf("A(%d， %d)=%d"， end+i-2， j-1， an[(end+i-2)%2][j-1]。nu[an[(end+i-2)%2][j-1]。len]);
for(k=an[(end+i-2)%2][j-1]。len-1;k>=0;k--)
printf("%04d"， an[(end+i-2)%2][j-1]。nu[k]);
printf("\n");
printf("A(%d， %d)=%d"， end+i-1， j， an[(end+i-1)%2][j]。nu[an[(end+i-1)%2][j]。len]);
for(k=an[(end+i-1)%2][j]。len-1;k>=0;k--)
```

```
printf("%04d"， an[(end+i-1)%2][j]。nu[k]);
printf(" A(%d， %d)=%d"， end+i-1， j-1， an[(end+i-1)%2][j-1]。nu[an[(end+i-1)%2][j-1]。len]);
for(k=an[(end+i-1)%2][j-1]。len-1;k>=0;k--)
printf("%04d"， an[(end+i-1)%2][j-1]。nu[k]);
printf("\n");
count[0] = end+i-j-1;
l1 = mul(0， count， ln， nn， n1); /*n1=(end+i-j-1)*(end+i-3)!*/
tn1[j]。len = plus(an[(end+i-2)%2][j-1]。len， an[(end+i-2)%2][j-1]。nu， l1， n1， tn1[j]。nu);
/*tn2[j]=A(end+i-1， j)-A(end+i-1， j-1)*/
tn2[j]。len = sub(an[(end+i-1)%2][j]。len， an[(end+i-1)%2][j]。nu， an[(end+i-1)%2][j-1]。len，
an[(end+i-1)%2][j-1]。nu， tn2[j]。nu);
}
/*清空*/
for(j=1;j<=ANL;j++)
clean(an[(end+i)%2][j]。nu， MAX-1);
clean(n1， MAX-1);
/*An*/
for(j=1;j<=end+i;j++){
/*tn1*/
for(k=j;k<end+i;k++){
printf("+ tn1[%d] %d"， k， tn1[k]。nu[tn1[k]。len]);
for(l=tn1[k]。len-1;l>=0;l--)
printf("%04d"， tn1[k]。nu[l]);
/**/
l1 = plus(an[(end+i)%2][j]。len， an[(end+i)%2][j]。nu， tn1[k]。len， tn1[k]。nu， n1);
count[0] = 0;
an[(end+i)%2][j]。len = plus(l1， n1， 0， count， an[(end+i)%2][j]。nu);
/**/
printf(" = %d"， an[(end+i)%2][j]。nu[an[(end+i)%2][j]。len]);
for(l=an[(end+i)%2][j]。len-1;l>=0;l--)
printf("%04d"， an[(end+i)%2][j]。nu[l]);
printf("\n");
}
printf("An = %d"， an[(end+i)%2][j]。nu[an[(end+i)%2][j]。len]);
for(k=an[(end+i)%2][j]。len-1;k>=0;k--)
printf("%04d"， an[(end+i)%2][j]。nu[k]);
printf("\n");
/*tn2*/
for(k=1;k<j;k++){
printf("+ tn2[%d] %d"， k， tn2[k]。nu[tn2[k]。len]);
for(l=tn2[k]。len-1;l>=0;l--)
printf("%04d"， tn2[k]。nu[l]);
/**/
l1 = plus(an[(end+i)%2][j]。len， an[(end+i)%2][j]。nu， tn2[k]。len， tn2[k]。nu， n1);
count[0] = 0;
an[(end+i)%2][j]。len = plus(l1， n1， 0， count， an[(end+i)%2][j]。nu);
```

```
/**/
printf(" = %d"， an[(end+i)%2][j]。nu[an[(end+i)%2][j]。len]);
for(l=an[(end+i)%2][j]。len-1;l>=0;l--)
printf("%04d"， an[(end+i)%2][j]。nu[l]);
printf("\n");
}
printf("An = %4d"， an[(end+i)%2][j]。nu[an[(end+i)%2][j]。len]);
for(k=an[(end+i)%2][j]。len-1;k>=0;k--)
printf("%04d"， an[(end+i)%2][j]。nu[k]);
printf("\n");
/**/
count[0] = 0;
l1 = plus(an[(end+i)%2][j]。len， an[(end+i)%2][j]。nu， 0， count， n1); /*an = n1*/
an[(end+i)%2][j]。len = plus(l1， n1， an[(end+i)%2][j-1]。len， an[(end+i)%2][j-1]。nu，
an[(end+i)%2][j]。nu);
/**/
printf("A(n， %d) = %4d"， j， an[(end+i)%2][j]。nu[an[(end+i)%2][j]。len]);
for(k=an[(end+i)%2][j]。len-1;k>=0;k--)
printf("%04d"， an[(end+i)%2][j]。nu[k]);
printf("\n");
}
/*output Tn*/
ind1o = fopen("Tn(資料庫)1。txt"， "a"); /*Tn*/
fprintf(ind1o， "T%d>>\n"， end+i);
printf("T%d>>\n"， end+i);
for(j=1;j<=end+i-1;j++){
fprintf(ind1o， "%4d"， tn1[j]。nu[tn1[j]。len]);
printf("%4d"， tn1[j]。nu[tn1[j]。len]);
for(k=tn1[j]。len-1;k>=0;k--){
printf("%04d"， tn1[j]。nu[k]);
fprintf(ind1o， "%04d"， tn1[j]。nu[k]);
}
fprintf(ind1o， " ");
printf(" ");
fprintf(ind1o， "%4d"， tn2[j]。nu[tn2[j]。len]);
printf("%4d"， tn2[j]。nu[tn2[j]。len]);
for(k=tn2[j]。len-1;k>=0;k--){
fprintf(ind1o， "%04d"， tn2[j]。nu[k]);
printf("%04d"， tn2[j]。nu[k]);
}
 fprintf(ind1o， "\n");
 printf("\n");
}
/*output An*/
ind2o = fopen("Tn(資料庫)2。txt"， "a"); /*An*/
fprintf(ind2o， "A%d>>\n"， end+i);
```

```
printf("A%d>>\n"， end+i);
for(j=1;j<=end+i;j++){
fprintf(ind2o， "%4d"， an[(end+i)%2][j]。nu[an[(end+i)%2][j]。len]);
printf("%4d"， an[(end+i)%2][j]。nu[an[(end+i)%2][j]。len]);
for(k=an[(end+i)%2][j]。len-1;k>=0;k--){
fprintf(ind2o， "%04d"， an[(end+i)%2][j]。nu[k]);
printf("%04d"， an[(end+i)%2][j]。nu[k]);
}
fprintf(ind2o， " ");
printf(" ");
}
fprintf(ind2o， "\n");
printf("\n");
}
 /*關檔*/
fclose(ind1o);
fclose(ind2o);
}
return 0;
}
int plus(int la，int *a，int lb，int *b，int *ans)
{
int i， j， len=la>lb?la：lb;

clean(ans， MAX-1);
for(i=0;i<=len;i++){
*(ans+i) += *(a+i)+*(b+i);
//printf("a[%d] = %04d      b[%d] = %04d\nans[%d] = %d\n"， i， *(a+i)， i， *(b+i)， i， *(ans+i));
if(*(ans+i)>=10000){
*(ans+i+1) += *(ans+i)/10000;
*(ans+i) %= 10000;
if(i+1>len)
len = i+1;
}
}
while(*(ans+len)==0 && len>0) len--;
return len;
}
int sub(int la，int *a，int lb，int *b，int *ans)
{
int i， j， len=la>lb?la：lb;
int sub1[MAX]={0}， sub2[MAX]={0};

clean(ans， MAX-1);
/*copy*/
for(i=0;i<=la;i++)
sub1[i] = *(a+i);
```

```
for(i=0;i<=lb;i++)
sub2[i] = *(b+i);
/**/
printf("a=%d"， sub1[la]);
for(i=la-1;i>=0;i--)
printf("%04d"， sub1[i]);
printf(" b=%d"， sub2[lb]);
for(i=lb-1;i>=0;i--)
printf("%04d"， sub2[i]);
printf("\n");
/**/
for(i=0;i<=len;i++){
if(sub1[i]<sub2[i]){
j = 1;
while(sub1[i+j]==0)
j++;
while(j>0){
sub1[i+j] -= 1;
sub1[i+j-1] += 10000;
j--;
}
}
printf("i=%d    %d-%d=%d\n"， i， sub1[i]， sub2[i]， ans[i]);
*(ans+i) = sub1[i]-sub2[i];
while(sub1[len]==0 && len>0)       len--;
}
while(*(ans+len)==0 && len>0) len--;
/**/
printf("len=%d ans=%d"， len， *(ans+len));
for(i=len-1;i>=0;i--)
printf("%04d"， *(ans+i));
printf("\n");
return len;
}
int mul(int la，int *a，int lb，int *b，int *ans)
{
int i， j， len=la+lb;

for(i=0;i<MAX;i++)  *(ans+i) = 0;
for(i=0;i<=la;i++){
for(j=0;j<=lb;j++){
*(ans+i+j) += *(a+i)**(b+j);
if(*(ans+i+j)>=10000){
*(ans+i+j+1) += *(ans+i+j)/10000;
*(ans+i+j) %=10000;
if(i+j+1>len)
len = i+j+1;
}
```

```
}
}
while(*(ans+len)==0 && len>0) len--;
return len;
}
int lev(int n，int *ans)
{
int count[5]={0}， tem[MAX]={0}， len=0， i， j;

for(i=0;i<MAX;i++){
*(ans+i) = 0;
tem[i] = 0;
}
*(ans+0) = 1;
tem[0] = 1;
while(n>0){
count[0] = n;
if(n%2==1)
len = mul(len， tem， 0， count， ans);
else
len = mul(len， ans， 0， count， tem);
n--;
}
return len;
}

int tackle(char *str，int *ans)
{
int i， j， ls=strlen(str)-1;
char s[5];

s[4] = '\0';
for(i=0;i<ls/4;i++){
for(j=0;j<4;j++)
s[3-j] = *(str+(ls-4*i-j));
*(ans+i) = atoi(s);
}
s[ls%4+1] = '\0';
for(j=ls%4;j>=0;j--) s[j] = *(str+j);
*(ans+i) = atoi(s);
return i;
}
void clean(int *num， int len)
{
	int i;

	for(i=0;i<=len;i++)
		*(num+i) = 0;
```

}

【評語】040402

數學科展與純數學論文不同之處在於前者注重日常生活中的應用。本作品有待加強這方面的探索。

缺點：海報上沒有出現參考資料。

書面報告當中附有英文參考資料，然而其中參有中文標點符號。參考資料收集得雖然豐富，然而以其書寫方式令人懷疑作者對於其內容是否有所瞭解。