2010 臺灣國際科學展覽會

優勝作品專輯

編 號：010020-09

作品名稱

由蟲子問題衍生一路領先與*Motzkin* 路徑之對應及推廣

得獎獎項

數學科大會獎一等獎

美國正選代表:美國第61屆國際科技展覽會

學校名稱：高雄市立高雄高級中學

作者姓名：侯宗誠、許德瑋

指導老師：鍾玉才

關鍵詞：遞迴關係、一路領先、*Motzkin* 數列

作者簡介

我是侯宗誠，就讀於高雄中學三年級。

我的興趣是數學及打棒球。平時喜歡思考一些數學難題，常常一想就想好幾個小時，但我很喜歡破解問題的那種成就感。我也常會去書店看一些有關科學的書，我對時間及空間的本質有著濃厚的興趣。而我目前也屬於學校棒球隊的一員，很喜歡那種跟大家在場上拚鬥的感覺。

今年是我們第一次參加台灣國際科展，很高興可以入圍。希望藉由這次機會，可以認識一些志同道合的好朋友。

我是許德瑋，就讀於高雄中學三年級。

小時候爸媽買了許許多多的科普書籍，培養了我廣泛閱讀的習慣。平時閒暇時習慣聽聽音樂放鬆，或者讓自己沉醉在古典詩詞的美感中。

做科展的過程中，往往會在很久的一段時間後，發現原來研究的對象及目標間存在著如此簡單的道理！這也是最令我感到欣喜及驕傲的地方！希望藉由這次參加國際科展的機會，參觀更多優秀的作品，進而拓展自己的視野！

摘要

在數學課堂中，老師拋出一道甄試的口試題目，那是一道有關蟲類繁殖過程中，探討子代存在位置及其規律性的題目。此問題引起我們繼續討論的興趣，並試著應用至「一路領先」問題。我們試著改變其形狀來構造「一路領先」的路徑，再擴張其維度來解決任一人數「一路領先」的問題！

由於發現*Motzkin* 數列和三人「一路領先」給定得票數的情況一一對應，我們找到一種對應方法，將*Motzkin* 路徑和「一路領先」得票過程做一對一的對應！以*Motzkin* 路徑和三人「一路領先」為基礎，我們構造了「立體*Motzkin* 」，發現其路徑走法數竟和五人「一路領先」得票過程總方法數完全相同！若限制向量**(1,0,0)**只能出現在*xy* 平面上，則和四人「一路領先」得票過程一一對應！當我們在網路上搜尋資料時，發現有一種*lattice path* 的規則和四人「一路領先」的方法數完全一樣！我們一樣找到一種對應規則，讓此走法和四人「一路領先」得票過程一一對應！

架構出「立體*Motzkin* 」後，我們試著架構「n維*Motzkin*」，發現給定有規律的$(2n-1)$ 個n維向量，就可以構造出n人的「一路領先」！此方法對解決*lattice path*和投票問題等有顯著的幫助！

Abstract

Once, our math teacher raised to us a question regarding the reproduction of the worms. The question arouses our interest to discuss about the question further. Therefore, we start with probing into what pattern the reproduction of the worms is, and how many worms exist after a specific period of time. Then, we apply the results to the "Lead All The Way" problem. We first limit the original problem to a finite range and make it more viable to count the possibilities of the ballot path! Furthermore, we not only intend to solve the original problem, but also expand its dimension into 3-dimension, or even n-dimension! By expanding the dimension, we can increase as many candidates as necessary! We strive to solve the ballot path problem in the n-dimensional setting!

Seeing that Motzkin series correspond to that of "Lead All the Way" with only three candidates and the fixed total of ballots, we aim to draw a bijection between the Motzkin path and the ballot path! In terms of the bijection we have drawn, we construct five 3-dimensional vectors to build "solid Motzkin", whose possibilities of the path are equal to that of "Lead All the Way" with 5 candidates. However, when the vector (1,0,0) exists on the x-y dimension, the true is the same of "Lead All the Way" with 4 candidates. When we are searching for the related information on the Internet, we find that there's a lattice path pattern whose possibilities of paths are the same as four-candidate "Lead All The Way". Thus, we also draw a bijestion between these two, and we discover that solving this problem is not only helpful in Math but Physics! Above all, we considerthat the n-dimensional Motzkin does exist, and it is related to (2n-1)-candidate "Lead All The Way"!!

We consider our approach to solving the problems may be useful to many math problems,especially to the lattice path and ballot path problems!!

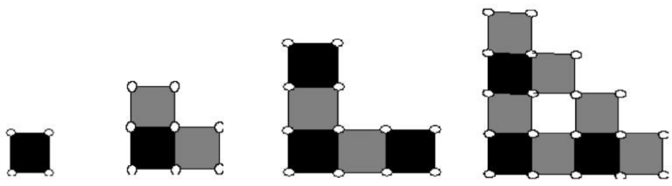
壹、研究動機：

一、甄試口試題目

在一道甄試的題目中，有一蟲類隨著時間增長的問題：

有一種昆蟲，佔有同一個方格為家，永遠不搬家。同一代的昆蟲在同一時間於其所在處的上方、右方各生下一個昆蟲(可能無性生殖)，生了之後就沒有生育能力，但也死不了。假設兩隻昆蟲同時佔有相同方格，那麼這兩隻蟲會彼此吞噬對方而亡。

假設第一代僅有一隻昆蟲，令$G(n)$ 為第n 代昆蟲的總數，$T(n)$為第一代至第n 代昆蟲的總數。已知$G(1)=1$ ，$G(2)=2$ ，$G(3)=2$ ；$T(1)=1$ ，$T(2)=3$ ，$T(3)=5$如下圖所示。請問(1). (2) ? n G = (2). (2) = ? n T 其中$n=0$ ， 1 ， 2 ， 3 ，・・・。(3). $T(1027)=?$

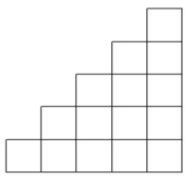


第一代　第二代　第三代　第四代

觀察每一子代的生長情況，探索蟲子生長的規律性。

二、改變昆蟲在同一時間於其所在處的上方、右方與前方各生下一個昆蟲(可能無性生殖)，生了之後就沒有生育能力，但也死不了。假設兩隻昆蟲同時佔有相同方格，那麼這兩隻蟲會彼此吞噬對方而亡；若三隻昆蟲同時佔有相同方格，則此格視為存在一蟲。觀察每一子代的生長情況，探索蟲子生長的規律性。

三、在特殊格子中如圖(一)，探討蟲子生長的規律性。



圖(一)

貳、研究目的：

一、解決甄試題目中，每一子代數目及一定時間後全部子代數目的總和，並歸納其規律性與巴斯卡定理的關聯。

二、歸納出空間方格的規律性及推廣。

三、歸納出特殊方格的規律性。

四、利用特殊方格性質，探討*Motzkin*路徑與解決兩人、三人「一路領先」得票過程對應與應用。

五、探討「立體*Motzkin* 」路徑與解決四人「一路領先」得票過程對應與應用。

参、研究設備：

一、電腦、模型、紙筆

二、*Dev C* + + 4.9.9.2

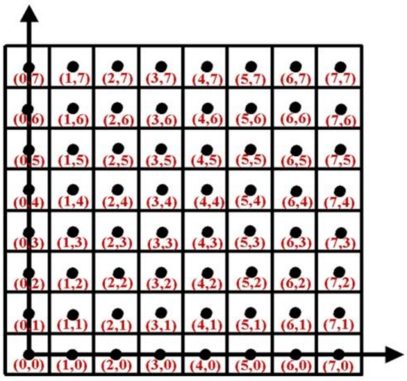
三、*Wolfram Mathematica* 6.0

肆、研究過程：

一、(一). 名詞解釋與符號：

1. 名詞解釋：

(1). 坐標格子：因為蟲子每次生殖的方向是往右、往上，所以可將棋盤格子置於直角坐標中，且定義棋盤格子的左下角(即母蟲所在位置)為(0,0) 。而每一格子皆有唯一坐標(x, y) 與之對應，如圖(二)，我們稱此為二維的坐標格子。同理，可推廣到n維坐標。



圖(二)

(2).一路領先：將物品分給n個人，且過程中恆符合第一人所得物品數$\geq$第二人所得物品數$\geq \ldots \geq$ 第n人所得物品數，稱之為一路領先。

(3). Motzkin路徑：從(0,0)走到$(n,0)$，規定走法為依循向量(1,0)、(1,1)或(1,−1)，且不走到x軸下方所形成的路徑。

(4). Motzkin數：從(0,0) 走到$(n,0)$，規定走法為依循向量(1,0)、(1,1)或 (1,−1)，且不走到x軸下方的方法數。

(5).立體Motzkin路徑：從(0,0,0) 走到$(n,0,0)$，規定走法為依循向量(1,0,0)、(1,1,0)、(1,−1,0) 、(1,1,−1) 或(1,−1,1) ，且行進中的向量終點坐標(x, y, z) 必須恆是$x\geq 0$，$y\geq 0$，$z\geq 0$的路徑。我們稱之為「立體*Motzkin* 」路徑。

(6). n維*Motzkin*：從$(0,0,\ldots,0)$ 走到$(k,0,\ldots,0)$ ，規定走法為依循向量$(1,0,\ldots,0)$ ， $(1,1,\ldots,0)$ ， $(1,-1,\ldots,0)$ ， $(1,1,-1,\ldots,0)$ ， $(1,-1,1,\ldots,0)$ ， $(1,0,1,-1,\ldots,0)$ ， $(1,0,-1,1,\ldots,0)$ ，…， $(1,0,\ldots,1,-1,0)$ ， $(1,0,\ldots,-1,1,0)$ ， $(1,0,\ldots,1,-1)$ ， $(1,0,\ldots,-1,1)$ $(2n-1)$ 種n維向量，且行進中的向量終點坐標$(a_1, a_2,\ldots,a_n)$必須滿足$a_i\geq 0$，$\forall i = 1,2,\ldots, n$ 。我們稱之為「n維*Motzkin* 」。特別規定$n\in N$且$n\geq 4$。

2. 符號：

(1). 二維

(a). $W_{m,n}$：其值為檢驗坐標格子(m,n)，$m,n\in N\cup\{0\}$是否有蟲子存在。

(b). $T_{m,n}$：其值為檢驗坐標格子(m,n)，$m,n\in N\cup\{0\}$是否有蟲子存在。其中坐標格子(m,n) 均需滿足$m\geq n$ 。

(2). 三維

(a). $W_{m,n,l}$：其值為檢驗坐標格子(m,n,l)，$m,n,l\in N\cup\{0\}$是否有蟲子存在。

(b). $T_{m,n,l}$：其值為檢驗坐標格子(m,n,l)，$m,n,l\in N\cup\{0\}$是否有蟲子存在。其中坐標格子(m,n,l) 均需滿足$m\geq n\geq l$ 。

(3). n維

(a). $W_{a_1,a_2,\ldots a_n}$ ：其值為檢驗坐標格子$(a_1, a_2,\ldots,a_n)$，$a_1, a_2,\ldots,a_n\in N\cup\{0\}$是否有蟲子存在。

(b). $T_{a_1,a_2,\ldots a_n}$ ：其值為檢驗坐標格子$(a_1, a_2,\ldots,a_n)$，$a_1, a_2,\ldots,a_n\in N\cup\{0\}$是否有蟲子存在。其中坐標格子$(a_1, a_2,\ldots,a_n)$均需滿足$a_1\geq a_2\geq\ldots\geq a_n$。

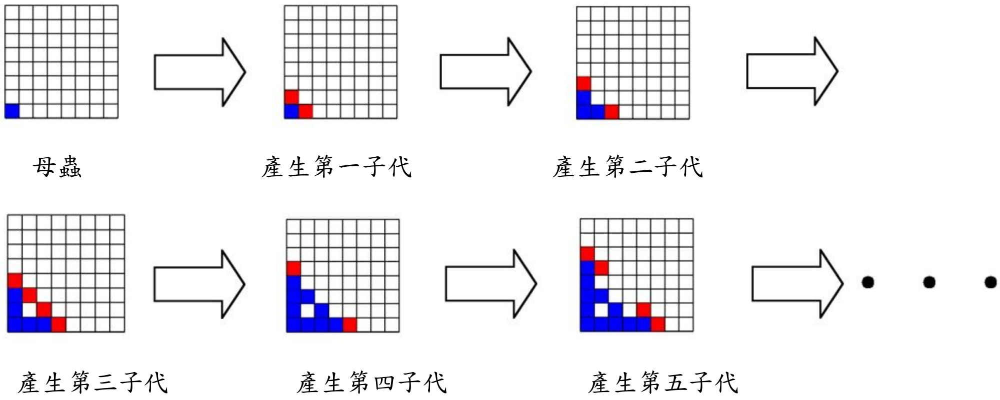
(4). 組合：$\begin{pmatrix} & m+n+l & \\ m, & n, & l\end{pmatrix}=\dfrac{(m+n+l)!}{m!n!l!}$ ，其中$m,n,l\in N\cup\{0\}$。

(5). M_n：從$(0,0)$ 走到$(n,0)$ ，規定走法為依循向量$(1,0)$ 、$(1,1)$ 或$(1,-1)$ ，且不走到x軸下方的方法數，即為*Motzkin*數。

(6). L_n：給定總票數n，$n\in N\cup\{0\}$，試求甲得票數「一路領先」乙得票數，乙得票數「一路領先」丙得票數，且丙得票數「一路領先」丁得票數之方法總數。

(二). 規定原題目第一代昆蟲為母蟲，第二代昆蟲為母蟲產生的第一子代，第三代為第二子代，以此類推。

1.畫出母體和每一子代的增長情形，並記錄下來，如圖(三)：



圖(三)

2. 根據遊戲規則，推導出某一格子中的蟲子是否存在。

(1). 從定義推導，可知：

(a). $\forall(m,0)$ 格子，其中$m\in N\cup\{0\}$，皆有存在蟲子。

(b). $\forall(0,n)$ 格子，其中$n\in N\cup\{0\}$，皆有存在蟲子。

(c). 不在兩軸邊界上的坐標格子(即$xy\neq0$)：

(Ⅰ). 若坐標格子左方及下方的坐標格子皆存在或皆不存在蟲子時，則此坐標格子必不存在蟲子。

(Ⅱ). 若此坐標格子之左方及下方恰只有一格有蟲子，則此坐標格子必存在蟲子。

由(a). (b). (c). 結果，我們定義：

$$\begin{cases} \text{若坐標格子}(m,n)\ (\text{其中}m,n\in N\cup\{0\})\text{有蟲子，則}W_{m,n}=-1 \\ \text{若坐標格子}(m,n)\ (\text{其中}m,n\in N\cup\{0\})\text{無蟲子，則}W_{m,n}=1 \end{cases}$$

且$W_{m,n}=W_{m,n-1}\times W_{m-1,n}$，其中$m,n\in N$， $W_{0,0}=-1$。

於此，對兩軸邊界上的坐標格子而言，為了滿足遞迴關係

$W_{m,n}=W_{m,n-1}\times W_{m-1,n}$，即$W_{m,0}=W_{m,-1}\times W_{m-1,0}$或$W_{0,n}=W_{0,n-1}\times W_{-1,n}$

定義：$m=-1$ 或$n=-1$ 時， $W_{m,n}=1$。

(2). 證明：$W_{m,n}=(-1)^{C_m^{m+n}}$，其中$m,n\in N\cup\{0\}$：

分三步驟證明如下：

(a). 證明：$W_{1,n}=(-1)^{n+1}=(-1)^{C_1^{n+1}}$ ，其中$n\in N$：(證明同(c)，於此省略)

(b). 證明：$W_{2,n}=(-1)^{C_2^{n+2}}$ ，其中$n\in N$。(證明同(c)，於此省略)

(c). 證明：給定n ，$W_{m,n}=(-1)^{C_n^{m+n}}$ ，其中$m\in N$ 。

pf：當$m=1$時，$W_{m,n}=(-1)^{C_1^{n+1}}$ ，$\therefore m=1$成立

假設$m=k$ 時成立，即$W_{k,n}=(-1)^{C_n^{k+n}}$ 則$m=k+1$時，

$\because W_{m,n}=W_{m,n-1}\times W_{m-1,n}$

$$\therefore \begin{cases} W_{k+1,n}=W_{k+1,n-1}\times W_{k,n} \\ W_{k+1,n-1}=W_{k+1,n-2}\times W_{k,n-1} \\ \quad\vdots \\ W_{k+1,1}=W_{k+1,0}\times W_{k,1} \end{cases}$$

$$\begin{aligned}\therefore W_{k+1,n}&=W_{k+1,0}\times(W_{k,1}\times W_{k,2}\times\cdots\times W_{k,n})\\ &=W_{k,0}\times(W_{k,1}\times W_{k,2}\times\cdots\times W_{k,n})=\prod_{i=0}^{\mathrm{n}}W_{k,i}\\ &=(-1)^{C_k^k}\times(-1)^{C_k^{k+1}}\times\cdots\times(-1)^{C_k^{k+n}}\\ &=(-1)^{C_{k+1}^{(k+1)+n}}\end{aligned}$$

$\therefore$由數學歸納法得知本題得證

可得, $W_{m,n}=(-1)^{C_m^{m+n}}$ ，其中$m,n\in N\cup\{0\}$ 。

因此，對二維平面的任意坐標格子，皆可得知坐標格子中是否存在蟲子。

即任一坐標格子(m,n) ，$m,n\in N\cup\{0\}$ ，

$$\begin{cases} \text{若}W_{m,n}\text{ 的冪次 }C_n^{m+n}\text{ 為奇數，則此坐標格子存在蟲子；} \\ \text{若}W_{m,n}\text{ 的冪次 }C_n^{m+n}\text{ 為偶數，則此坐標格子蟲子不存在。} \end{cases}$$

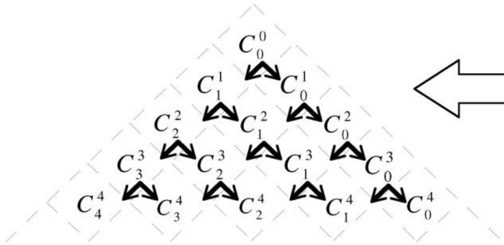
(3). 性質：

(a). 若只考慮坐標格子對應的, $W_{m,n}=(-1)^{C_m^{m+n}}$ 的冪次值，則發現每一坐標格

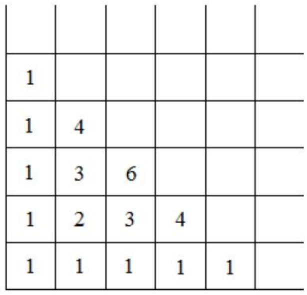
子所對應之冪次 C_n^{m+n} ，與*Pascal Triangle*中之數字相互對應。

(b). 根據巴斯卡定理：$C_m^{m+n}=C_{m-1}^{m+n-1}+C_m^{m+n-1}$， $\forall m\geq n\geq 1$發現此算式恰是蟲子生長$W_{m,n}= W_{m-1,n}\times W_{m,n-1}$，其中$m,n\geq 1$所對應的冪次關係，如圖(四)。

(*c*). 坐標格子所對應的冪次之值恰是由坐標格子**(**0,0**)** 到坐標格子(*m*,*n*) 的捷徑走法，如圖(五)。

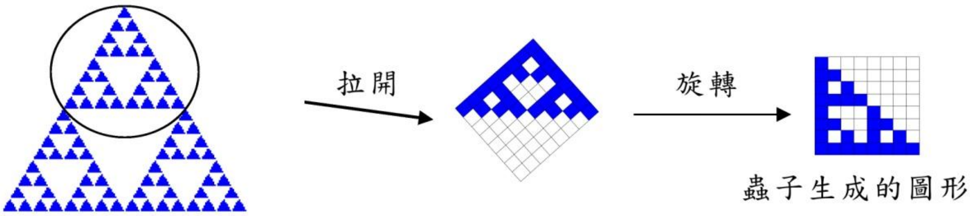


圖（四）



圖（五）

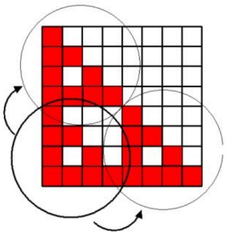
(*d*).若把有蟲子的坐標格子塗上顏色，我們發現其圖形具有自我重複性，也就是「自我碎形」！根據相關資料顯示蟲子生長所形成的圖形恰與*Sierpinski gasket* 形成對應關係，如圖(六)。



Sierpinski gasket

圖(六)

由於蟲子存在的位置有著「自我碎形」的特性。我們觀察到其圖形會由起始點算起以($2^n\times 2^n$)正方形，（$n\in N\cup\{0\}$），向上向右複製相同的圖形，如圖(七)。



圖(七)

由以上性質，針對這道甄試的口試題目：

(Ⅰ)對$G(2^n)$，已知$G(2^0)=1$， $G(2^1)=2$，則$G(2^2)=2\times G(2^1)=2\times 2=2^2 \Rightarrow G(2^n)=2^n$。

(Ⅱ). 對$T(2^n)$，已知 $T(2^0)$， $T(2^1)=3$，又因為「自我碎形」關係，

$\therefore T(2^2)=(2+1)\times T(2^1)=3\times 3=3^2 \Rightarrow T(2^n)=3^n$。

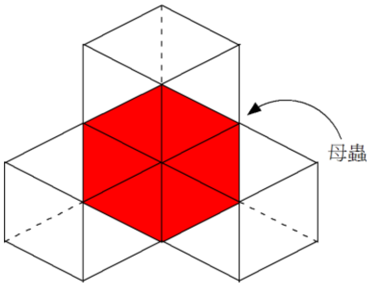
(Ⅲ). $\because T(1027)=T(1024)+G(1025)+G(1026)+G(1027)$

$=3^{10}+G(1025)+G(1026)+G(1027)$

由自我碎形的特性：$\therefore G(1026)=4$ ，$G(1027)=4$

$\therefore T(1027)=3^{10}+2+4+4=3^{10}+10=59059$

二、(一). 改變昆蟲在同一時間於其所在處的上方、右方與前方各生下一個昆蟲(可能無性生殖)，生了之後就沒有生育能力，但也死不了。假設兩隻昆蟲同時佔有相同方格，那麼這兩隻蟲會彼此吞噬對方而亡，若三隻昆蟲同時佔有相同方格，則此格視為存在一蟲。觀察每一子代的生長情況，探索蟲子生長的規律性。如圖(八)所示。



圖(八)

(二). 根據遊戲規則及仿照一.(二). 2.之觀念，

1. (1). $\forall (m,n,0)$，$m,n\in N\cup\{0\}$。因為蟲子生長只受兩個方向的坐標格子影響，

所以狀況等同於平面。故其坐標格子是否有蟲子可以$(-1)^{C_m^{m+1}}$表之。

(2).同(1).，$\forall (m,0,l)$，$m,l\in N\cup\{0\}$，坐標格子是否有蟲子可以$(-1)^{C_m^{m+1}}$表之。

同理，$\forall (0,n,l)$，$n,l\in N\cup\{0\}$，坐標格子是否有蟲子可以$(-1)^{C_n^{m+1}}$表之。

(3). $\forall (m, n, l)$坐標格子，其中$m, n, l \in N$，我們可知影響其是否有蟲子的坐標格子為$(m-1, n, l)$，$(m, n-1, l)$，$(m, n, l-1)$，所以：

(a). 若坐標格子$(m-1, n, l)$，$(m, n-1, l)$，$(m, n, l-1)$中都沒蟲子，則坐標格子(m, n, l) 必不存在蟲子。

(b). 若坐標格子$(m-1, n, l)$，$(m, n-1, l)$，$(m, n, l-1)$中恰有一格有蟲子，則坐標格子(m, n, l) 必存在蟲子。

(c). 若坐標格子$(m-1, n, l)$，$(m, n-1, l)$，$(m, n, l-1)$中恰有兩格有蟲子，則坐標格子(m, n, l)必不存在蟲子。

(d). 若坐標格子$(m-1, n, l)$，$(m, n-1, l)$，$(m, n, l-1)$ 中三格均有蟲子，則坐標格子(m, n, l) 必存在蟲子。

由(1). (2). (3). 結果，我們定義：

$$\begin{cases} \text{若坐標格子}(m,n,l)\ (m,n,l \in N \cup \{0\})\ \text{有蟲子，則} W_{m,n,l} = -1 \\ \text{若坐標格子}(m,n,l)\ (m,n,l \in N \cup \{0\})\ \text{無蟲子，則} W_{m,n,l} = 1 \end{cases}$$

且$W_{m,n,l} = W_{m-1,n,l} \times W_{m,n-1,l} \times W_{m,n,l-1}$，其中$m, n, l \in N$，$W_{0,0,0} = -1$。

2. 證明：$W_{m,n,l} = (-1)^{\frac{(m+n+l)!}{m!n!l!}}$，其中$m, n, l \in N \cup \{0\}$(見附錄)

因此，對於三維空間的任意坐標格子，皆可得知其坐標格子中是否存在蟲子。

即任一坐標格子(m,n,l)，其中$m,n,l \in N \cup \{0\}$，

若$W_{m,n,l}$ 的冪次$\begin{pmatrix} & m+n+l & \\ m, & n, & l \end{pmatrix}$值為奇數，則此坐標格子存在蟲子；

若$W_{m,n,l}$ 的冪次$\begin{pmatrix} & m+n+l & \\ m, & n, & l \end{pmatrix}$值為偶數，則此坐標格子蟲子不存在。

3. 性質：

(1). 參考第四十七屆科展國中組作品：「立體」巴斯卡。此時，若只考慮坐標格子對應之$W_{m,n,l} = (-1)^{\begin{pmatrix} & m+n+l & \\ m, & n, & l \end{pmatrix}}$的冪次，則發現每一坐標格子所對應之冪次$\begin{pmatrix} & m+n+l & \\ m, & n, & l \end{pmatrix}$之值與「立體」巴斯卡中之數字相互對應。

(2). 依據二維巴斯卡定理：

$C_m^{m+n-1}+C_{m-1}^{m+n-1}=C_m^{m+n}$ ，其中 $\forall m\geq n\geq 1$ 的型式，

我們不妨稱算式：

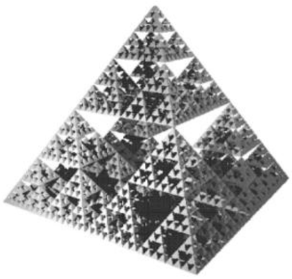
$$\binom{m+n+l-1}{m-1,\quad n,\quad l}+\binom{m+n+l-1}{m,\quad n-1,\quad l}+\binom{m+n+l-1}{m,\quad n,\quad l-1}=\binom{m+n+l}{m,\quad n,\quad l}$$

其中 $m,n,l\geq 1$，為「立體」巴斯卡定理，而此算式恰是蟲子生長

$W_{m,n,l}=W_{m-1,n,l}\times W_{m,n-1,l}\times W_{m,n,l-1}$，其中 $m,n,l\geq 1$，所對應的冪次關係。

(3). 坐標格子所對應的冪次值恰是由坐標格子 $(0,0,0)$ 到坐標格子 (m,n,l) 的捷徑走法。

(4). 若把有蟲子的坐標格子塗上顏色，我們發現圖形具有自我重複性，也就是「自我碎形」！根據相關資料顯示蟲子生長所形成的圖形恰與 *Sierpinski gasket* 形成對應關係。如圖(九)所示。



圖(九)

此圖源自http://local.wasp.uwa.edu.au/~pbourke/fractals/gasket/

(三). 1. 仿二維與三維狀況，我們定義 n 維：

$$\begin{cases}\text{若坐標格子}(a_1,a_2,\ldots,a_n)\text{，}a_1,a_2,\ldots,a_n\in N\cup\{0\}\text{有蟲子，則}W_{a_1,a_2,\ldots,a_n}=-1\\ \text{若坐標格子}(a_1,a_2,\ldots,a_n)\text{，}a_1,a_2,\ldots,a_n\in N\cup\{0\}\text{無蟲子，則}W_{a_1,a_2,\ldots,a_n}=1\end{cases}$$

且 $W_{a_1,a_2,\cdots,a_n}=W_{a_1,a_2,\cdots,a_n}\times W_{a_1,a_2-1,\cdots,a_n}\times\ldots\times W_{a_1,a_2,\cdots,a_n-1}$

其中 $a_1,a_2,\ldots a_n\in N$，$W_{0,0,\ldots,0}=-1$。

2. 證明：$\binom{(a_1-1)+a_2+\ldots a_n}{a_1-1,a_2,\ \ldots,\ a_{n-1},a_n}+\binom{a_1+(a_2-1)+\ldots+a_n}{a_1,a_2-1,\ \ldots,\ a_{n-1},a_n}$

$+\ldots+\binom{a_1+a_2+\ldots+(a_n-1)}{a_1,a_2,\ \ldots,\ a_{n-1},a_n-1}=\binom{a_1+a_2+\ldots+\ldots a_n}{a_1,a_2,\ \ldots,\ a_{n-1},a_n}$

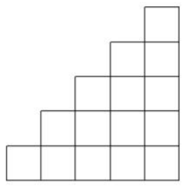
pf :於此省略。

為了一致性，我們稱此性質為「n 維」巴斯卡定理。由此推論：

$$W_{a_1,a_2,\cdots,a_n}=(-1)^{\binom{a_1+a_2+\dots+\dots a_n}{a_1,a_2,\ \cdots\ ,a_{n-1},a_n}},$$

其中$a_1, a_2, \dots a_n \in N \cup \{0\}$。所以，我們可以推論出n維巴斯卡的架構，進而標示出蟲子所存在的位置及關係。

三、(一).於二維中改變規則，限定每個坐標格子(m,n)皆符合$m \geq n$，$m,n \in N \cup \{0\}$，因此我們將圖形改變，如圖(十)所示：



圖(十)

從遊戲規則可知，格子是否有蟲子存在同樣受其左方及下方格子影響。

我們仿照 $W_{m,n}$：

1.定義：

$$\begin{cases} 若座標格子(m,n)\ (m,n \in N \cup \{0\})有蟲子，則\ T_{m,n}=-1 \\ 若座標格子(m,n)\ (m,n \in N \cup \{0\})無蟲子，則\ T_{m,n}=1 \end{cases}$$

2.證明：$T_{m,n}=(-1)^{C_n^{m+n}-C_{n-1}^{m+n}}$，$\forall m \geq n \geq 0$（見附錄）

$$\begin{cases} 當T_{m,n}的冪次C_n^{m+n}-C_{n-1}^{m+n}為奇數，則坐標格子存在蟲子； \\ 當T_{m,n}的冪次C_n^{m+n}-C_{n-1}^{m+n}為偶數，則坐標格子蟲子不存在。 \end{cases}$$

3.性質：

(1). 因為$T_{m,n}=(-1)^{C_n^{m+n}-C_{n-1}^{m+n}}$，$\forall m \geq n \geq 1$

當$n=0$ 時，$T_{m,n}=T_{m-1,0}\times T_{m,-1}=T_{m-1,0}$，而$T_{m,0}$下方並無坐標格子，所以依定義令$T_{m,-1}=1$。

(2). 因為$T_{m,n}=(-1)^{C_n^{m+n}-C_{n-1}^{m+n}}$，其中$m \geq n \geq 0$，則$T_{m,n}$的冪次之值恰是由**(0,0)** 走到**($m$, n)** 的「一路領先」方法數。

(3). 若令$m=n$ 代入，其中$m,n \in N \cup \{0\}$，則$T_{n,n}$之冪次值

$C_n^{m+n}-C_{n-1}^{n+n}=\frac{1}{n+1}C_n^{2n}$，即為$Catalan$ 數。

(二). 於三維中改變立體形狀，限制任一坐標格子(m, n, l)皆滿足$m\geq n\geq l\geq 0$。仿(一)

1. 定義：$T_{m,n,l}=T_{m-1,n,l}\times T_{m,n-1,l}\times T_{m,n,l-1}$，其中$m>n>l>0$，$T_{0,0,0}=-1$

2. 證明：

$$T_{m,n,l}=(-1)^{\binom{m+n+l}{m,\ n,\ l}-\binom{m+n+l}{m,\ n+1,\ l-1}+\binom{m+n+l}{m+1,\ n+1,\ l-2}-\binom{m+n+l}{m+1,\ n-1,\ l}}$$

$$\times(-1)^{\binom{m+n+l}{m+2,\ n-1,\ l-1l}-\binom{m+n+l}{m+2,\ n,\ ll-2}}$$

其中$m, n, l\in N\cup\{0\}$ 且$m\geq n\geq l$。（見附錄）

$$\begin{cases}\text{若}T_{m,n,l}\text{的冪次值為奇數，則此坐標格子存在蟲子。}\\ \text{若}T_{m,n,l}\text{的冪次值為偶數，則此坐標格子蟲子不存在。}\end{cases}$$

3. 性質：

(1). $T_{m,n,l}=(-1)^{\binom{m+n+l}{m,\ n,\ l}-\binom{m+n+l}{m,\ n+1,\ l-1}+\binom{m+n+l}{m+1,\ n+1,\ l-2}-\binom{m+n+l}{m+1,\ n-1,\ l}}$

$\times(-1)^{\binom{m+n+l}{m+2,\ n-1,\ l-1l}-\binom{m+n+l}{m+2,\ n,\ ll-2}}$，其中$m\geq n\geq l\geq 0$，

則$T_{m,n,l}$ 的冪次值恰是由(0,0,0) 走到(m,n, l) 的「一路領先」方法數。

(2). 仿二維情況，若甲乙丙三人得票數依序為x, y, z，其中$x, y, z\in N\cup\{0\}$，

且甲得票數「一路領先」乙得票數α張，乙得票數「一路領先」丙得票數β張，

其中$\alpha,\beta\in N\cup\{0\}$，則可令$\acute{x}=x-(\alpha+\beta)$，$\acute{y}=y-\beta$，$\acute{z}=z$

$$\therefore T_{x-(\alpha+\beta),y-\beta,z}=(-1)^{\binom{x+y+z-\alpha-2\beta}{x-\alpha-\beta,\ y-\beta,\ z}-\binom{x+y+z-\alpha-2\beta}{x-\alpha-\beta,\ y-\beta+1,\ z-1}+\binom{x+y+z-\alpha-2\beta}{x-\alpha-\beta+1,\ y-\beta+1,\ z-2}}$$

$$\times(-1)^{-\binom{x+y+z-\alpha-2\beta}{x-\alpha-\beta+1,\ y-\beta-1,\ z}+\binom{x+y+z-\alpha-2\beta}{x-\alpha-\beta+2,\ y-\beta-1,\ z-1}-\binom{x+y+z-\alpha-2\beta}{x-\alpha-\beta+2,\ y-\beta,\ z-2}}$$

(三).n維情形：

由$W_{m,n}$，$W_{m,n,l}$，$T_{m,n}$，$T_{m,n,l}$的冪次發現，$W_{m,n}$，$T_{m,n}$為坐標格子$(m-1,n)$、$(m,n-1)$所對應之冪次值相加，故其冪次值為從坐標格子(0,0) 走到坐標格子為(m,n)之捷徑走法數(空間亦同)。因此，我們可以仿二維、三維作法，推廣到n維的「一路領先」。

1. 推廣到n 維狀況：

(1). 觀察二維中冪次值的算式：

$$C_n^{m+n}-C_{n-1}^{m+n}=(m+n)!\begin{vmatrix}\dfrac{1}{m!} & \dfrac{1}{(m+1)!}\\ \dfrac{1}{(n-1)!} & \dfrac{1}{n!}\end{vmatrix}$$

(2). 觀察三維中冪次值的算式：

$$\begin{aligned}&\binom{m+n+l}{m,\quad n,\quad l}-\binom{m+n+l}{m,\quad n+1,\quad l-1}+\binom{m+n+l}{m+1,\quad n+1,\quad l-2}\\&-\binom{m+n+l}{m+1,\quad n-1,\quad l}+\binom{m+n+l}{m+2,\quad n-1,\quad l-1}-\binom{m+n+l}{m+2,\quad n,\quad l-2}\\&=(m+n+l)!\begin{vmatrix}\dfrac{1}{m!} & \dfrac{1}{(m+1)!} & \dfrac{1}{(m+2)!}\\ \dfrac{1}{(n-1)!} & \dfrac{1}{n!} & \dfrac{1}{(n+1)!}\\ \dfrac{1}{(l-2)!} & \dfrac{1}{(l-1)!} & \dfrac{1}{l!}\end{vmatrix}\end{aligned}$$

(3). 推論$T_{a_1,a_2,\dots,a_n}$之冪次值為：

$$(a_1+a_2+\cdots+a_n)!\begin{vmatrix}\dfrac{1}{a_1!} & \dfrac{1}{(a_1+1)!} & \cdots & & \dfrac{1}{[a_1+(n-1)]!}\\ \dfrac{1}{(a_2-1)!} & \dfrac{1}{a_2!} & \cdots & & \dfrac{1}{[a_2+(n-2)]!}\\ \vdots & \vdots & & \vdots & \vdots\\ & & \cdots & \dfrac{1}{a_{n-1}!} & \dfrac{1}{(a_{n-1}+1)!}\\ \dfrac{1}{[a_n-(n-1)]!} & \dfrac{1}{[a_n-(n-2)]!} & \cdots & \dfrac{1}{(a_n-1)!} & \dfrac{1}{a_n!}\end{vmatrix},n\geq 2\cdots\cdots\cdots\cdots\cdots(*)$$

其中$a_1\geq a_2\geq\cdots\geq a_n$，$a_1,a_2\cdots$，$a_n\in N\cup\{0\}$。（為了方便呈現，我們稱上式(*)為$T_{a_1,a_2,\dots,a_n}$之冪次值公式）

pf ：(*a*). 若σ為集合$\{1,2,\cdots,n\}$上的一對一函數，則$\{\sigma(1),\sigma(2),\cdots,\sigma(n)\}=\{1,2,\cdots,n\}$，這樣的函數，我們稱為一個$n$個文字的排列，有時將一個排列σ以下面的方式表示：

$\sigma=\begin{pmatrix}1 & 2 & \cdots & n-1 & n\\ \sigma(1) & \sigma(2) & \cdots & \sigma(n-1) & \sigma(n)\end{pmatrix}$，例如$\sigma=\begin{pmatrix}1 & 2 & 3\\ 2 & 3 & 1\end{pmatrix}$時，

$a_{1\sigma(1)}a_{2\sigma(2)}a_{3\sigma(3)}$就表$a_{12}a_{23}a_{31}$。通常以$S_n$表示所有$n$個文字的排列全體集合，顯然$S_n$中共有$n!$個元素。而前述的$\sigma=\begin{pmatrix}1 & 2 & 3\\ 2 & 3 & 1\end{pmatrix}$，可以這樣合成：

$\begin{pmatrix}1 & 2 & 3\\ 1 & 2 & 3\end{pmatrix}\rightarrow\begin{pmatrix}1 & 2 & 3\\ 2 & 1 & 3\end{pmatrix}\rightarrow\begin{pmatrix}1 & 2 & 3\\ 2 & 3 & 1\end{pmatrix}$，即一個排列若可經由偶數次「互換」的合成而得，則稱偶排列；否則稱為奇排列。以sgnσ來表示σ的奇偶性如下：

$$\text{sgn}\sigma=\begin{cases}1\text{，若}\sigma\text{為偶排列}\\ -1\text{，若}\sigma\text{為奇排列}\end{cases}$$

$$\therefore\begin{vmatrix}a_{11} & a_{12} & \cdots & a_{1n}\\ a_{21} & a_{22} & \cdots & a_{2n}\\ \vdots & \vdots & & \vdots\\ a_{n1} & a_{n2} & \cdots & a_{nn}\end{vmatrix}=\sum_{\sigma\in S_n}(\text{sgn}\,\sigma)a_{1\sigma(1)}a_{2\sigma(2)}\cdots a_{n\sigma(n)}\ ,$$

(b). 當$n=2$時，T_{a_1,a_2}之冪次值滿足(*)，已證畢。

假設$n=k$時成立，即$T_{a_1,a_2,\cdots,a_k}$之冪次值滿足(*)

則$n=k+1$時，當$a_{k+1}=0$時，$T_{a_1,a_2,\cdots,a_k,0}$之冪次值與$T_{a_1,a_2,\cdots,a_k}$相同。

即在$(k+1)$維情況下，有些坐標格子$(a_1,a_2,\cdots,a_{k+1})$之冪次值滿足(*)。

假設存在不滿足$T_{a_1,a_2,\cdots,a_{k+1}}$之冪次值公式之坐標格子所形成之集合為

S，即$S=\{(a_1,a_2,\cdots,a_{k+1})|T_{a_1,a_2,\cdots,a_{k+1}}\text{之冪次值不滿足}(*)\}$

若$S=\emptyset$，則本題得證。

若$S\neq\emptyset$，在集合S中元素，針對第$(k+1)$坐標找到最小的a'_{k+1}，使得

$T_{a'_1,a'_2,\cdots,a'_{k+1}}$之冪次值不滿足(*)。

同理固定第$(k+1)$坐標a'_{k+1}，依序針對第k坐標找到最小的a'_k，使得

$T_{a'_1,a'_2,\cdots,a'_{k+1}}$之冪次值不滿足(*)。

如此類推，我們可以依序找到最小的$a'_{k+1}, a'_k, \cdots, a'_2, a'_1$使得$T_{a'_1,a'_2,\cdots,a'_{k+1}}$之冪次值不符合(*) 。

$\because T_{a'_1,a'_2,\cdots,a'_{k+1}} = T_{a'_1-1,a'_2,\ldots,a'_{k+1}} \times T_{a'_1,a'_2-1,\ldots,a'_{k+1}} \times \cdots \times T_{a'_1,a'_2-1,\ldots,a'_{k+1}-1}$

又 $T_{a'_1-1,a'_2,\cdots,a'_{k+1}} \times T_{a'_1,a'_2,-1,\ldots,a'_{k+1}} \times \cdots \times T_{a'_1,a'_2,\cdots,a'_{k+1}-1} \notin S$

$\therefore T_{a'_1-1,a'_2,\cdots,a'_{k+1}}$之冪次值為

$$\left[(a_1'-1)+a_2'+\cdots+a_k'+a_{k+1}'\right]!\begin{vmatrix} \frac{1}{(a_1'-1)!} & \frac{1}{a_1'!} & \cdots & & \frac{1}{\left[a_1'+(k-1)\right]!} \\ \frac{1}{(a_2'-1)!} & \frac{1}{a_2'!} & \cdots & & \frac{1}{\left[a_2'+(k-1)\right]!} \\ \vdots & \vdots & & \vdots & \vdots \\ & & \cdots & \frac{1}{a_k'!} & \frac{1}{(a_k'+1)!} \\ \frac{1}{(a_{k+1}'-k)!} & \frac{1}{\left[a_{k+1}'-(k-1)\right]!} & \cdots & \frac{1}{(a_{k+1}'-1)!} & \frac{1}{a_{k+1}'!} \end{vmatrix}$$

$\therefore T_{a'_1,a'_2,-1,\ldots,a'_{k+1}}$之冪次值為

$$\left[(a_1'-1)+a_2'+\cdots+a_k'+a_{k+1}'\right]!\begin{vmatrix} \frac{1}{a_1'!} & \frac{1}{(a_1'+1)!} & \cdots & & \frac{1}{(a_1'+k)!} \\ \frac{1}{(a_2'-2)!} & \frac{1}{(a_2'-1)!} & \cdots & & \frac{1}{(a_2'+k-2)!} \\ \vdots & \vdots & & \vdots & \vdots \\ & & \cdots & \frac{1}{a_k'!} & \frac{1}{(a_k'+1)!} \\ \frac{1}{(a_{k+1}'-k)!} & \frac{1}{\left[a_{k+1}'-(k-1)\right]!} & \cdots & \frac{1}{(a_{k+1}'-1)!} & \frac{1}{a_{k+1}'!} \end{vmatrix},$$

以此類推。

則$T_{a'_1-1,a'_2,\cdots,a'_{k+1}} \times T_{a'_1,a'_2-1,\ldots,a'_{k+1}} \times \cdots \times T_{a'_1,a'_2,\cdots,a'_{k+1}-1}$之冪次值為

$$
(a_1'+a_2'+\cdots+a_{k+1}'-1)!\left[\begin{vmatrix}
\frac{1}{(a_1'-1)!} & \frac{1}{a_1'!} & \cdots & & \frac{1}{[a_1'+(k-1)]!} \\
\frac{1}{(a_2'-1)!} & \frac{1}{a_2'!} & \cdots & & \frac{1}{[a_2'+(k-1)]!} \\
\vdots & \vdots & & \vdots & \vdots \\
 & & \cdots & \frac{1}{a_k'!} & \frac{1}{(a_k'+1)!} \\
\frac{1}{(a_{k+1}'-k)!} & \frac{1}{[a_{k+1}'-(k-1)]!} & \cdots & \frac{1}{(a_{k+1}'-1)!} & \frac{1}{a_{k+1}'!}
\end{vmatrix}+\begin{vmatrix}
\frac{1}{a_1'!} & \frac{1}{(a_1'+1)!} & \cdots & & \frac{1}{(a_1'+k)!} \\
\frac{1}{(a_2'-2)!} & \frac{1}{(a_2'-1)!} & \cdots & & \frac{1}{[a_2'+(k-2)]!} \\
\vdots & \vdots & & \vdots & \vdots \\
 & & \cdots & \frac{1}{a_k'!} & \frac{1}{(a_k'+1)!} \\
\frac{1}{(a_{k+1}'-k)!} & \frac{1}{[a_{k+1}'-(k-1)]!} & \cdots & \frac{1}{(a_{k+1}'-1)!} & \frac{1}{a_{k+1}'!}
\end{vmatrix}\right.
$$

$$
\left.+\cdots+\begin{vmatrix}
\frac{1}{a_1'!} & \frac{1}{(a_1'+1)!} & \cdots & & \frac{1}{(a_1'+k)!} \\
\frac{1}{(a_2'-1)!} & \frac{1}{a_2'!} & \cdots & & \frac{1}{[a_2'+(k-1)]!} \\
\vdots & \vdots & & \vdots & \vdots \\
 & & \cdots & \frac{1}{a_k'!} & \frac{1}{(a_k'+1)!} \\
\frac{1}{[a_{k+1}'-(k+1)]!} & \frac{1}{(a_{k+1}'-k)!} & \cdots & \frac{1}{(a_{k+1}'-2)!} & \frac{1}{(a_{k+1}'-1)!}
\end{vmatrix}\right]\cdots\cdots(**)
$$

於(**)中，針對此$(k+1)$個行列式，令

$$
\begin{vmatrix}
\frac{1}{(a_1'-1)!} & \frac{1}{a_1'!} & \cdots & & \frac{1}{[a_1'+(k-1)]!} \\
\frac{1}{(a_2'-1)!} & \frac{1}{a_2'!} & \cdots & & \frac{1}{[a_2'+(k-1)]!} \\
\vdots & \vdots & & \vdots & \vdots \\
 & & \cdots & \frac{1}{a_k'!} & \frac{1}{(a_k'+1)!} \\
\frac{1}{(a_{k+1}'-k)!} & \frac{1}{[a_{k+1}'-(k-1)]!} & \cdots & \frac{1}{(a_{k+1}'-1)!} & \frac{1}{a_{k+1}'!}
\end{vmatrix}=\sum_{\sigma\in S_{k+1}}(\operatorname{sgn}\sigma)\,a_{1\sigma(1)}^{(1)}a_{2\sigma(2)}^{(1)}\cdots a_{(k+1)\sigma(k+1)}^{(1)}
$$

$$
\begin{vmatrix}
\frac{1}{a_1'!} & \frac{1}{(a_1'+1)!} & \cdots & & \frac{1}{(a_1'+k)!} \\
\frac{1}{(a_2'-2)!} & \frac{1}{(a_2'-1)!} & \cdots & & \frac{1}{[a_2'+(k-2)]!} \\
\vdots & \vdots & & \vdots & \vdots \\
 & & \cdots & \frac{1}{a_k'!} & \frac{1}{(a_k'+1)!} \\
\frac{1}{(a_{k+1}'-k)!} & \frac{1}{[a_{k+1}'-(k-1)]!} & \cdots & \frac{1}{(a_{k+1}'-1)!} & \frac{1}{a_{k+1}'!}
\end{vmatrix}=\sum_{\sigma\in S_{k+1}}(\operatorname{sgn}\sigma)\,a_{1\sigma(1)}^{(2)}a_{2\sigma(2)}^{(2)}\cdots a_{(k+1)\sigma(k+1)}^{(2)}
$$

，以此類推。

$$
\begin{vmatrix}
\frac{1}{a_1'!} & \frac{1}{(a_1'+1)!} & \cdots & & \frac{1}{(a_1'+k)!} \\
\frac{1}{(a_2'-1)!} & \frac{1}{a_2'!} & \cdots & & \frac{1}{[a_2'+(k-1)]!} \\
\vdots & \vdots & & \vdots & \vdots \\
 & & \cdots & \frac{1}{a_k'!} & \frac{1}{(a_k'+1)!} \\
\frac{1}{(a_{k+1}'-k)!} & \frac{1}{[a_{k+1}'-(k-1)]!} & \cdots & \frac{1}{(a_{k+1}'-1)!} & \frac{1}{a_{k+1}'!}
\end{vmatrix}=\sum_{\sigma\in S_{k+1}}(\operatorname{sgn}\sigma)\,a_{1\sigma(1)}a_{2\sigma(2)}\cdots a_{(k+1)\sigma(k+1)}\cdots\cdots\cdots\cdots(***)
$$

考慮(***)與(**)中$(k+1)$個行列式：

對任一$\sigma \in S_{k+1}$，$sgn\,\sigma$為同號，且觀察$a_{1\sigma(1)}^{(i)}a_{2\sigma(2)}^{(i)}\cdots a_{(k+1)\sigma(k+1)}^{(i)}$，$1\le i\le k+1$，

(Ⅰ). 考慮$\sigma_{p\sigma(p)}^{(1)},\sigma_{p\sigma(p)}^{(2)},\cdots,\sigma_{p\sigma(p)}^{(k+1)}$，$1\le p\le k+1$

若$\sigma_{p\sigma(p)}^{(i)}=\dfrac{1}{\left[a_p-(p-1)+(\sigma(p)-1)-1\right]!}=\dfrac{1}{\left[a_p-p-\sigma(p)-1\right]!}$，

則$\forall j\ne i$，$\sigma_{p\sigma(p)}^{(j)}=\dfrac{1}{\left[a_p-p-\sigma(p)\right]!}$，其中$1\le i,j\le k+1$

(Ⅱ). 考慮$\sigma_{1\sigma(1)}^{(i)}\sigma_{2\sigma(2)}^{(i)}\cdots\sigma_{k+1\sigma(k+1)}^{(i)}$，$1\le i\le k+1$

若$\sigma_{q\sigma(q)}^{(i)}=\dfrac{1}{\left[a_q-q-\sigma(q)-1\right]!}$，$1\le q\le k+1$

則$\sigma_{j\sigma(j)}^{(i)}=\dfrac{1}{\left[a_j-j-\sigma(j)\right]!}$，$1\le j\le k+1$且$j\ne q$

$$(\operatorname{sgn}\sigma)\left(a_{1\sigma(1)}^{(1)}a_{2\sigma(2)}^{(1)}\cdots a_{(k+1)\sigma(k+1)}^{(1)}+a_{1\sigma(1)}^{(2)}a_{2\sigma(2)}^{(2)}\cdots a_{(k+1)\sigma(k+1)}^{(2)}+\cdots+a_{1\sigma(1)}^{(k+1)}a_{2\sigma(2)}^{(k+1)}\cdots a_{(k+1)\sigma(k+1)}^{(k+1)}\right)$$
$$=(\operatorname{sgn}\sigma)\left[\left(a_1'+a_2'+\cdots+a_{k+1}'\right)a_{1\sigma(1)}a_{2\sigma(2)}\cdots a_{(k+1)\sigma(k+1)}\right]\ \text{(由 } n \text{ 維巴斯卡定理)}$$

$$\therefore\left(a_1'+a_2'+\cdots+a_{k+1}'-1\right)!\sum_{i=1}^{k+1}\sum_{\sigma\in S_{k+1}}(\operatorname{sgn}\sigma)a_{1\sigma(1)}^{(i)}a_{2\sigma(2)}^{(i)}\cdots a_{(k+1)\sigma(k+1)}^{(i)}$$
$$=\left(a_1'+a_2'+\cdots+a_{k+1}'-1\right)!\sum_{\sigma\in S_{k+1}}(\operatorname{sgn}\sigma)\left(a_1+a_2+\cdots+a_{k+1}\right)a_{1\sigma(1)}a_{2\sigma(2)}\cdots a_{(k+1)\sigma(k+1)}$$
$$=(a_1+a_2+\cdots+a_{k+1})!\sum_{\sigma\in S_{k+1}}(\operatorname{sgn}\sigma)a_{1\sigma(1)}a_{2\sigma(2)}\cdots a_{(k+1)\sigma(k+1)}\text{，矛盾 。}$$

$\therefore S=\phi$。由數學歸納法得知本題得證

(四). 1. 給定票數n，$n\in N\cup\{0\}$，試求甲得票數「一路領先」乙得票數之方法總數

為 $C_{\left[\frac{n}{2}\right]}^{n}$。

pf :平面中，已知$T_{m,n}=(-1)^{C_{m-1}^{m+n-1}-C_{n-2}^{m+n-1}}$

對 $m+n=k$ ，代入

總數： $(C_{k-1}^{k-1}-C_{-2}^{k-1})+(C_{k-2}^{k-1}-C_{-1}^{k-1})+\dots+(C_{k-1-\left[\frac{k}{2}\right]}^{k-1}-C_{-2+\left[\frac{k}{2}\right]}^{k-1})$

$$=(C_{k-1}^{k-1}-C_{0}^{k-1})+(C_{k-2}^{k-1}+C_{1}^{k-1})+\dots+(C_{\frac{k}{2}+1}^{k-1}-C_{-2+\frac{k}{2}}^{k-1})+C_{k-1-\left[\frac{k}{2}\right]}^{k-1}+C_{k-\left[\frac{k}{2}\right]}^{k-1}$$

$$=C_{k-\left[\frac{k}{2}\right]-1}^{k-1}+C_{k-\left[\frac{k}{2}\right]}^{k-1}$$

(1). 若 $2|k$ ，

$$C_{k-\left[\frac{k}{2}\right]-1}^{k-1}+C_{k-\left[\frac{k}{2}\right]}^{k-1}=\frac{(k-1)!}{\frac{k}{2}!(\frac{k}{2}-1)!}+\frac{(k-1)!}{\frac{k}{2}!(\frac{k}{2}-1)!}=2\times\frac{k!}{\frac{k}{2}!\frac{k}{2}!}\times\frac{\frac{k}{2}}{k}=C_{\frac{k}{2}}^{k}$$

(2). 若 $2\vee k$ ，

$$C_{k-\left[\frac{k}{2}\right]-1}^{k-1}+C_{k-\left[\frac{k}{2}\right]}^{k-1}=C_{k-\frac{k-1}{2}-1}^{k-1}+C_{k-\frac{k-1}{2}}^{k-1}=\frac{(k-1)!}{(\frac{k-1}{2})!(\frac{k-1}{2})!}+\frac{(k-1)!}{(\frac{k-3}{2})!(\frac{k+1}{2})!}$$

$$=\left[\frac{(k-1)!}{(\frac{k+1}{2})!(\frac{k-1}{2})!}\right](\frac{k+1}{2}+\frac{k-1}{2})=C_{\frac{k-1}{2}}^{k}$$

綜合(1).(2).可知，總數即為$C_{\left[\frac{k}{2}\right]}^{k}$

若A_n為給定票數n的兩人「一路領先」方法總數，由此，我們推得其遞迴式

為$A_{n+1}=\frac{2_n+3+(-1)^{n+1}}{n+2}A_n$，其中 $n\in N$

四、(一).1. 給定票數 n，$n\in N\cup\{0\}$，試求甲得票數「一路領先」乙得票

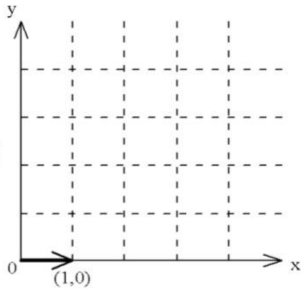
數，且乙得票數「一路領先」丙得票數之方法總數為$\sum_{K\geq 0}\binom{n}{2k}C_k$ ，其中

$C_k=\frac{1}{k+1}\binom{2k}{k}$（即 *Catalan* 數）。

pf :利用電腦程式檢驗出甲乙丙三人「一路領先」總方法數的前幾項，發

現與*Motzkin* 數一一對應，於是思考三人「一路領先」與*Motzkin* 路徑的關聯。觀察M_n，$n \in N$與「給定總票數且甲得票≥乙得票≥丙得票的方法數」之間關係，並令甲乙丙得票分別為*abc*:

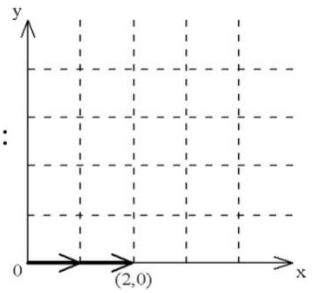
M_1：



（甲，乙，丙）得票過程 ⟷ (1, 0, 0)

得票順序： *a*

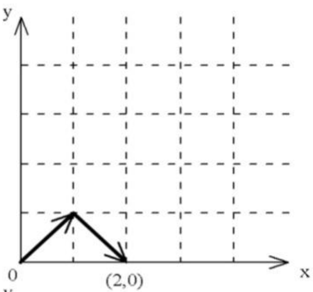
M_2：



（甲，乙，丙）得票過程 ⟷ (1, 0, 0)

→ (2, 0, 0)

得票順序： *a a*

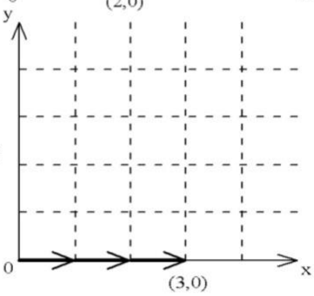


（甲，乙，丙）得票過程 ⟷ **(1, 0, 0)**

→ (1, 1, 0)

得票順序： *ab*

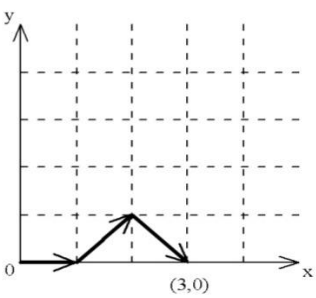
M_3：



（甲，乙，丙）得票過程 ⟷ (1, 0, 0) → (2, 0, 0)

→ (3, 0, 0)

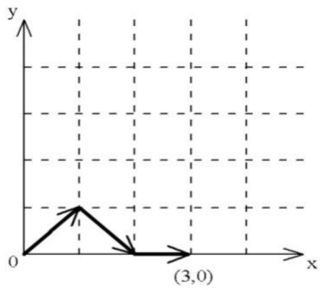
得票順序： *aaa*



（甲，乙，丙）得票過程 ⟷ (1, 0, 0) → (2, 0, 0)

→ (2, 1, 0)

得票順序： *aab*

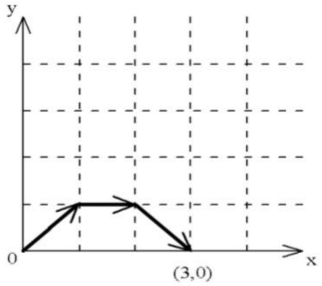


（甲，乙，丙）得票過程

⟷ (1, 0, 0) → (1, 1, 0)

⟶ (2, 1, 0)

得票順序： *aba*



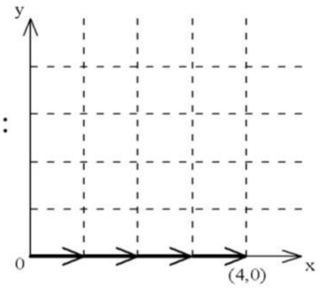
（甲，乙，丙）得票過程

⟷ (1, 0, 0) → (1, 1, 0)

⟶ (1, 1, 1)

得票順序： *abc*

M_4 ：

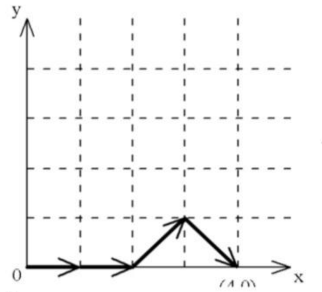


（甲，乙，丙）得票過程

⟷ (1, 0, 0) → (2, 0, 0)

⟶ (3, 0, 0) → (4, 0, 0)

得票順序： *aaaa*

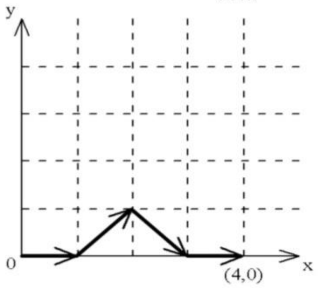


（甲，乙，丙）得票過程

⟷ (1, 0, 0) → (2, 0, 0)

⟶ (3, 0, 0) → (3, 1, 0)

得票順序： *aaab*

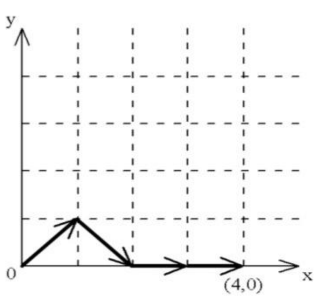


（甲，乙，丙）得票過程

⟷ (1, 0, 0) → (2, 0, 0)

⟶ (2, 1, 0) → (3, 1, 0)

得票順序： *aaba*

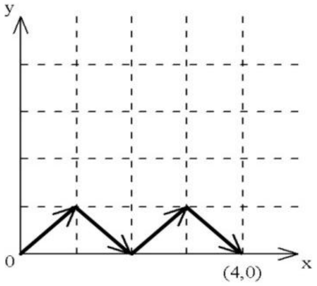


（甲，乙，丙）得票過程

⟷ (1, 0, 0) → (1, 1, 0)

⟶ (2, 1, 0) → (3, 1, 0)

得票順序： *abaa*

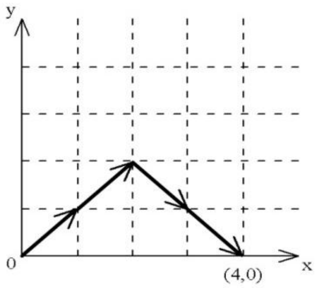


（甲，乙，丙）得票過程

⟷ (1, 0, 0) → (1, 1, 0)

→ (2, 1, 0) → (2, 2, 0)

得票順序： *abab*

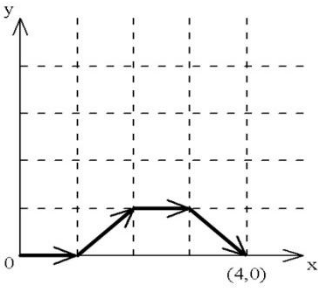


（甲，乙，丙）得票過程

⟷ (1, 0, 0) → (2, 0, 0)

→ (2, 1, 0) → (2, 2, 0)

得票順序： *aabb*

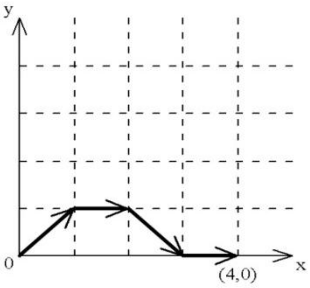


（甲，乙，丙）得票過程

⟷ (1, 0, 0) → (2, 0, 0)

→ (2, 1, 0) → (2, 1, 1)

得票順序： *aabc*

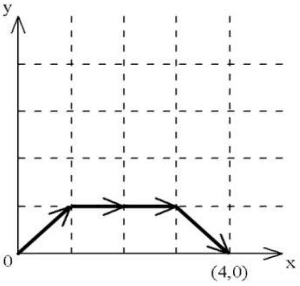


（甲，乙，丙）得票過程

⟷ (1, 0, 0) → (1, 1, 0)

→ (1, 1, 1) → (2, 1, 1)

得票順序： *abca*



（甲，乙，丙）得票過程

⟷ (1, 0, 0) → (1, 1, 0)

→ (2, 1, 1) → (2, 1, 1)

得票順序： *abac*

(1). 「一路領先」到*Motzkin* 路徑：由甲乙丙「一路領先」過程觀察可得：

(*a*). 每一個丙所對應的向量定為**(1,−1)**。

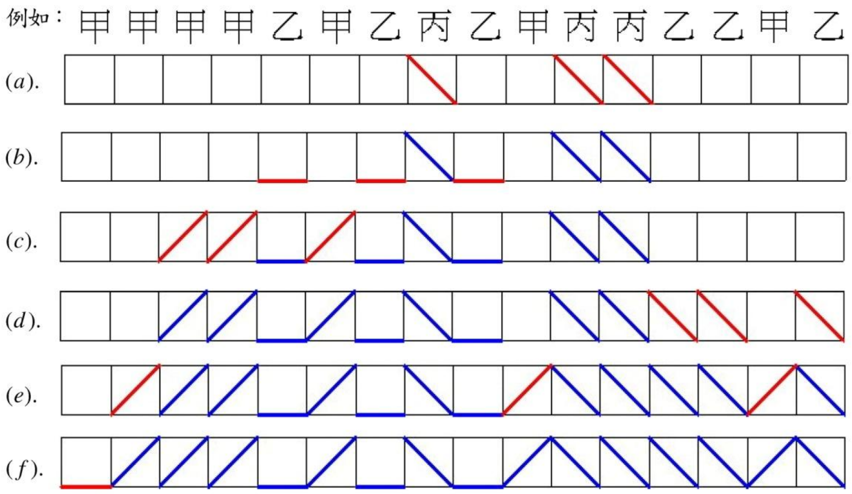
(*b*). 承(*a*).，每一個丙往左找最相近的乙，其對應向量定為**(1,0)**。

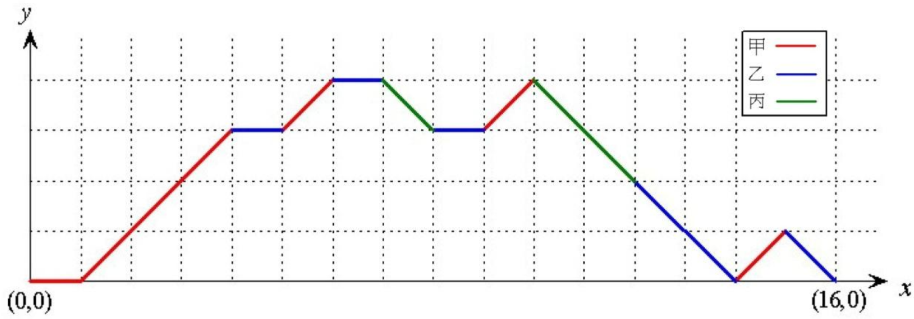
(*c*). 承(*b*).，每一個乙往左找最相近的甲，其對應向量定為**(1,1)**。

(*d*). 承(*a*). (*b*). (*c*).，若有剩餘的乙，其對應向量定為**(1,−1)**。

(*e*). 承(*d*).，每一個乙往左找最相近的甲，其對應向量定為**(1,1)**。

(*f*). 承(*c*). (*d*). (*e*).，若有剩餘的甲，其對應向量定為**(1,0)**。





(Ⅰ). 在步驟**(*a*). (*b*). (*c*).**對應中，將若干組(甲乙丙)的得票對應到向量走法分別為**(1,1) , (1,0) , (1,−1)**。且所對應出的向量 **(1,1)**和**(1,−1)**走法數相同，且先有**(1,1)**再有**(1,−1)**。

(Ⅱ). 在步驟**(*d*). (*e*). (*f*).**對應中，在(Ⅰ).之後將若干組(甲乙)的得票對應到向量走法分別為**(1,1) , (1,−1)**，且必先有**(1,1)**再有**(1,−1)**。

由(Ⅰ). (Ⅱ).，因對應向量**(1,1) , (1,0) , (1,−1)**走法三種，且對應路徑為一起點**(0,0)**，終點(*n*,0)且不走到*x*軸下方所形成的路徑，故為一*Motzkin* 路徑。

∴對於任意一個甲乙丙「一路領先」得票過程，均可找到一個且唯一的*Motzkin* 路徑與之對應。

(2). 由*Motzkin*路徑到「一路領先」：

(方法一)由*Motzkin* 路徑觀察可得：

(*a*).向量**(**1,1**)**及位於x軸上的向量**(1, 0)**定為甲。

(*b*).向量**(**1,0**)**可為甲或乙。

(*c*).向量**(**1,−1**)**可為乙或丙。

(*d*).承(*c*).：

對任一 *Motzkin* 路徑，由左至右考慮由一個或連續數個向量(1,−1)所組成的下降區間$[c,d]$。由區間$[c,d]$往左找到最相近的一始點位於x軸的向量**(1,1)**，設此向量**(1,1)**位於區間$[a,b]$上。

(Ⅰ). 若區間$[b,c]$中無水平向量**(1,0)**，則在區間$[c,d]$的向量(1,−1)全部填乙。

(Ⅱ). 若區間$[b,c]$中存在水平向量**(1,0)**。

(i). 由左至右考慮由一個或連續數個向量**(1,0)**所組成任一水平區間$[g,h]$。

①由區間$[g,h]$往左找到最相近向量**(1,0)**的甲，設此向量**(1,0)**位於區間$[e,f]$上，且在區間$[f,g]$中，甲比乙多k個。

ⓐ若區間$[g,h]$的向量(1,0)個數$\geq$k時，則在區間[g,h]上由左至右填入k 個乙，剩餘者填甲。

ⓑ若區間$[g,h]$的向量(1,0)個數$<$ k時，則在區間[g,h]的向量(1,0)全部填乙。

②由區間[g,h]往左找，若無向量(1,0)的甲時，則替換成往左找一始點位於x軸的向量(1,1)，設此向量(1,1)位於區間$[e',f']$上，且在區間$[e', g]$中，甲比乙多k'個，討論情形與上ⓐⓑ同。

(ii). 考慮下降區間$[c, d]$：

①由區間$[c,d]$往左找到最相近向量$(1,-1)$的乙，設此向量$(1,-1)$位於區間$[i, j]$上，且在區間$[j,c]$中，乙比丙多l個。

ⓐ若區間$[c,d]$的向量$(1,-1)$個數$\geq l$時，則在區間$[c,d]$由上往下填入l個丙，剩餘者填乙。

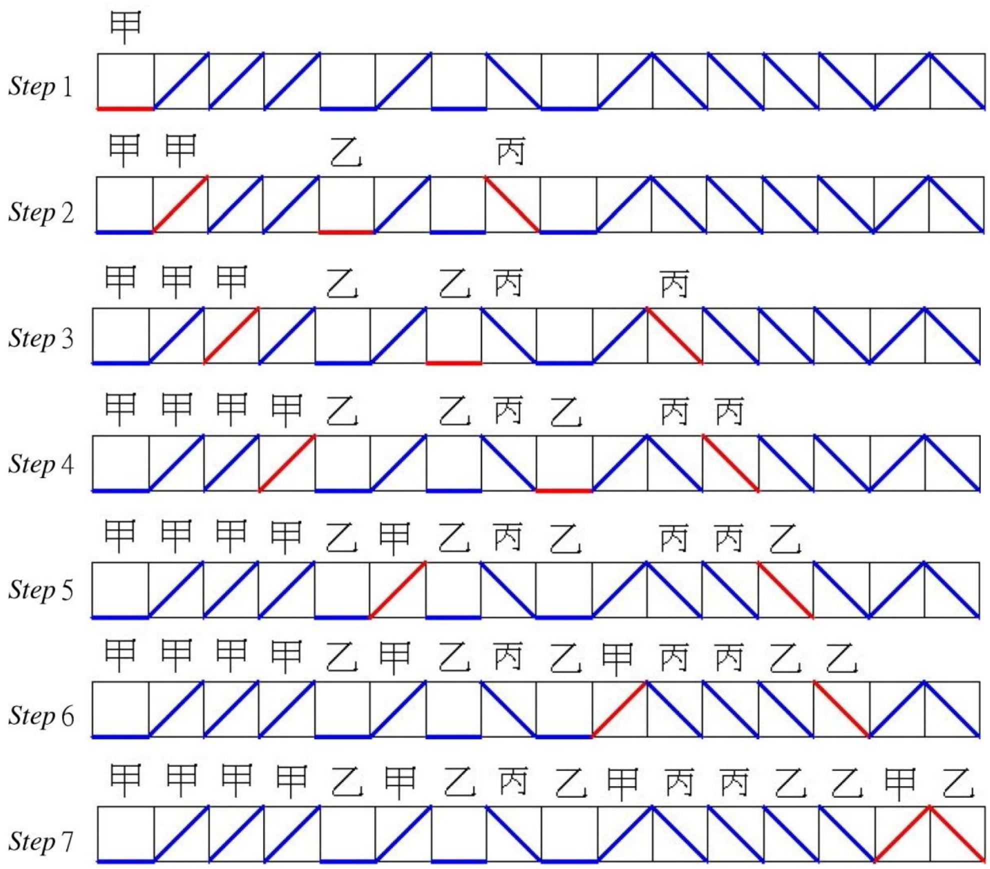
ⓑ若區間$[c,d]$的向量$(1,-1)$個數$< l$時，則在區間$[c,d]$的向量$(1,-1)$全部填丙。

②由區間$[c,d]$往左找，若無向量$(1,-1)$的乙時，則替換成往左找一始點位於x軸的向量$(1,1)$，設此向量$(1,1)$位於區間$[i', j']$上，且在區間$[i',c]$中，乙比丙多l'個，討論情況與上ⓐⓑ同。

由此，對於任意一個*Motzkin*路徑，均可找到一個且唯一的甲乙丙「一路領先」得票過程與之對應。所以，給定票數n，$n\in N\cup\{0\}$之甲乙丙「一路領先」方法數即為M_n，由文獻知$M_n=\sum_{k\geq 0}\binom{n}{2k}C_k$，其中$C_k=\frac{1}{k+1}\binom{2k}{k}$即*Catalan*數。

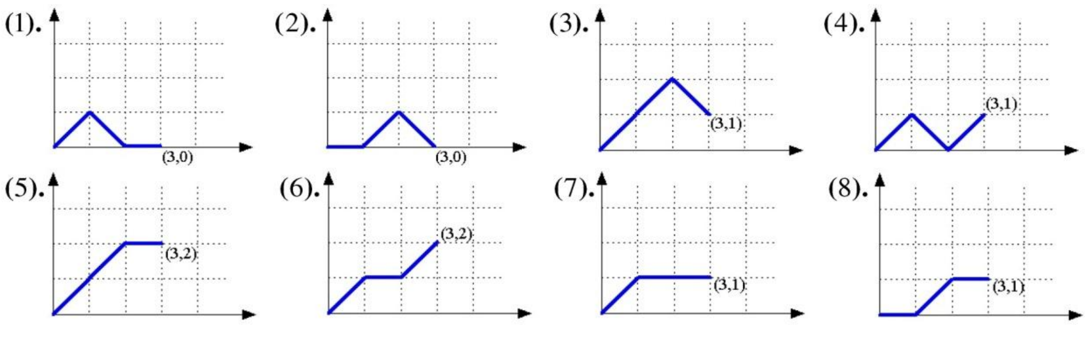
此外，若在*Motzkin*路徑中限制向量$(1,0)$走法僅可在x軸上出現時，此即和甲乙兩人「一路領先」得票過程形成一一對應。

(方法二)從*Motzkin*路徑對回「一路領先」得票順序時，由左往右，若向量$(1,1)$後非$(1,1)$的向量先遇到$(1,0)$，則先找$(1,1)$，$(1,0)$，$(1,-1)$配對為甲乙丙。若先找到$(1,-1)$則先找$(1,1)$，$(1,-1)$配對為甲乙。單獨的$(1,0)$即為甲。此種對應方式也可以將*Motzkin*路徑對回「一路領先」得票順序，1-1 *onto*。例如：



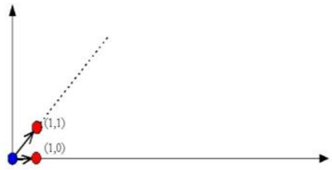
2. 參考文獻(十四)中提到「*A quarter-century later [R2], he conjetured（essentially）the following statement：Regev's Conjecture*：$A_{2,1,0}(n) = M(n-1) - M(n-3)$」，作者*Doron Zeilberger*用一個「非對應」的方式證明這個猜想，而他的*final remarks*中提到：「*Regev* [*R*2] *raises the question of finding a bijective proof of his* $M(n-1) - M(n-3)$ *conjecture.*」！因為證明了*Motzkin*路徑和「一路領先」對應方式，我們可以用一個「對應」的方式證明這個猜想。為了與前述的表示方式相同，我們以M_n代替$M_{(n)}$。

證明：由*Motzkin*路徑，我們可將甲得兩票，乙得一票的情形分成8種情況：

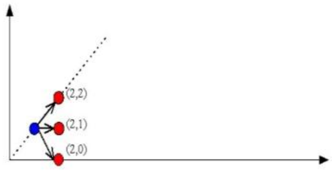


圖（十一）

其中，(1)(2)暫止於(3,0)，(3)(4)(7)(8)暫止於(3,1)，(5)(6)暫止於(3,2)，如圖（十一）。



圖（十二）



圖（十三）

(*a*). 如圖（十二），可知$(0,0)\to(n,0)$的方法數為以下兩種方法數的總和：

(Ⅰ). $(1,0)\to(n,0)$的方法數，即為M_{n-1}。

(Ⅱ). $(1,1)\to(n,0)$的方法數。

已知$(0,0)\to(n,0)$的方法數為M_n，可知$(1,1)\to(n,0)$的方法數為M_n-M_{n-1}。因此，$(3,1)\to(n,0)$的方法數為$M_{n-2}-M_{n-3}$。

(*b*). 如圖（十三），可知$(1,1)\to(n,0)$的方法數為以下三種方法數的總和：

(Ⅰ). $(2,0)\to(n,0)$的方法數，即為M_{n-2}。

(Ⅱ). $(2,1)\to(n,0)$的方法數，即為$M_{n-1}-M_{n-2}$。

(Ⅲ). $(2,2)\to(n,0)$的方法數。

已知$(1,1)\to(n,0)$的方法數為M_n-M_{n-1}，可知$(2,2)\to(n,0)$的方法數為$\left(M_n-M_{n-1}\right)-\left(M_{n-1}-M_{n-2}\right)-M_{n-2}=M_{n-2}-M_{n-1}$，因此，$(3,2)\to(n,0)$的方法數為$M_{n-1}-2M_{n-2}$。

將此八種狀況的方法數相加：

$2(M_{n-3})+4(M_{n-2}-M_{n-3})+2(M_{n-1}-2M_{n-2})=2M_{n-1}-2M_{n-3}$

但對於(1)(2)，(3)(4)，(5)(6)，(7)(8)四組，其後面的*Motzkin*路徑若相同，則同組之後續對應的「一路領先」順序相同。在不考慮前三票甲乙的得票順序時，此四組中兩個應視為同類。因此我們將結果除以2，得到$A_{2,1,0}(n)=\mathrm{M}_{n-1}-\mathrm{M}_{n-3}$，其中$A_{2,1,0}(n)$為前三票甲兩票乙一票，再給$(n-3)$票三人的一路領先方法數。

五、由四.知道*Motzkin* 路徑和三人「一路領先」得票過程形成對應。若限制水平向量只出現在*x*軸時，即為二人「一路領先」的得票過程。於是，思考能否架構*Motzkin*路徑到立體空間，使得從(0,0,0)走到$(n,0,0)$的方法數與四人、五人，或者更多人的「一路領先」總方法數形成一一對應。

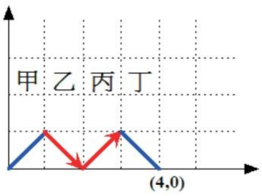
根據平面二維的推測以及利用電腦程式檢驗，發現從(0,0,0)走到(n,0,0)，走法為「依循向量(1,0,0)、(1,1,0)、(1,−1,0)、(1,−1,1)、(1,1,−1)，且行進中的向量終點坐標(x, y, z)必須恆是$x \geq 0$，$y \geq 0$，$z \geq 0$」的方法數，經由前幾項的觀察，發現其路徑走法數與甲乙丙丁戊五人「一路領先」總方法數形成一一對應！若將此路徑稱為「立體*Motzkin*」，則不難發現在「立體*Motzkin*」路徑中，若限制向量(1,0,0) 走法僅可在xy平面上出現時，經由前幾項的觀察從(0,0,0)走到(n,0,0)的方法數和四人「一路領先」總方法數形成一一對應！以下驗證此性質：

(為了方便呈現，我們將圖形投影至xy平面，並以紅色表示立體向量(1,−1,1)和(1,1,−1)走法)

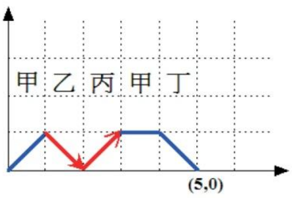
(一).「立體*Motzkin*」路徑與四人「一路領先」得票順序的對應關係：

由圖形觀察：

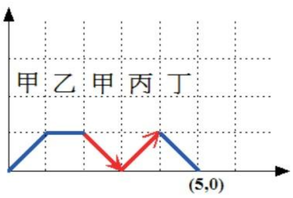
(1).



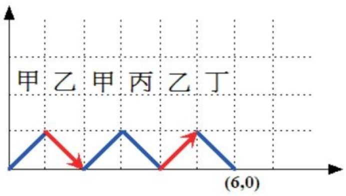
(2).



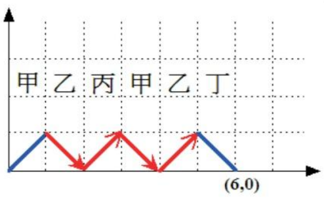
(3).



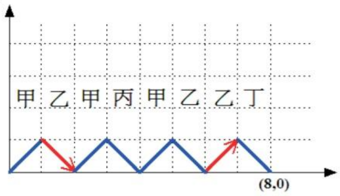
(4).



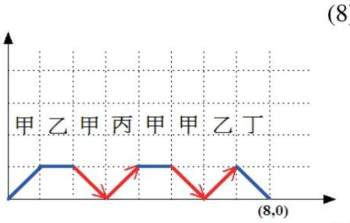
(5).



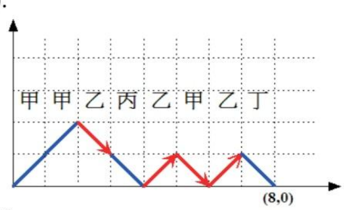
(6).



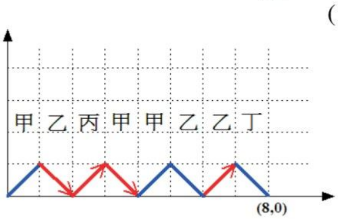
(7).



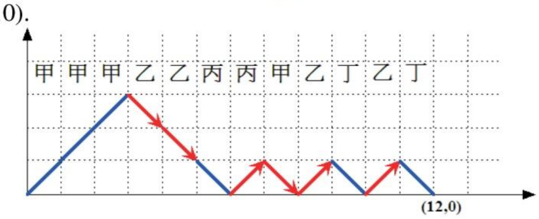
(8).



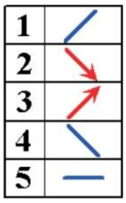
(9).



(10).



1. 甲乙丙丁四人一路領先得票過程(四人得票情形分別以a、b、c、d表之)到「立體 *Motzkin*」路徑(以下為方便呈現，將向量(1,1,0)、(1,-1,1)、(1,1,-1)、 (1,-1,0)、(1,0,0) 分別以向量1，2，3，4，5 表示)



(1). 配對$abcd$ ：由右往左依序一個d 往左找到最鄰近的c，再以此c往左找到最鄰近的b，再以此b往左找到最鄰近的a，直至所有d均已被配對為止。。將所配成若干組的$abcd$ ， 對應向量分別定為1234 。

(2). 配對abc：對於剩餘的abc，由右往左依序一個c 往左找到最鄰近的b，再以此b往左找到最鄰近的a，直至剩餘的c均已被配對為止。將所配成若干組的abc，對應向量分別定為154。

(3). 配對ab：對於剩餘的ab ，由右往左依序一個b 往左找到最鄰近的a，直至剩餘的b 均已被配對為止。將所配成若干組的ab ，對應向量分別定為14 。

(4). 承步驟(1).(2).(3).，剩餘的a 對應向量定為5。

(5). 由右往左考慮步驟(1).所配對$abcd$ 組：

(a) .考慮向量$3(c)$與d，若之間存在由步驟(2).(3).對應的向量4(c)或4(b)，則考慮步驟(2).(3).所填入的向量14(ac)，與向量14(ab)：

(Ⅰ). 若向量$3(c)$的左方「步驟(2).(3).所填入的向量1(a)個數> 步驟(2).(3).所填入的向量4(c)與4(b)個數」：將向量$3(c)$與其右方最鄰近的向量4(c)或4(b)的位置置換。置換後，若此向量$3(c)$右方至d之間仍有向量4(c)或4(b)，就繼續考慮步驟(5). (a) .。

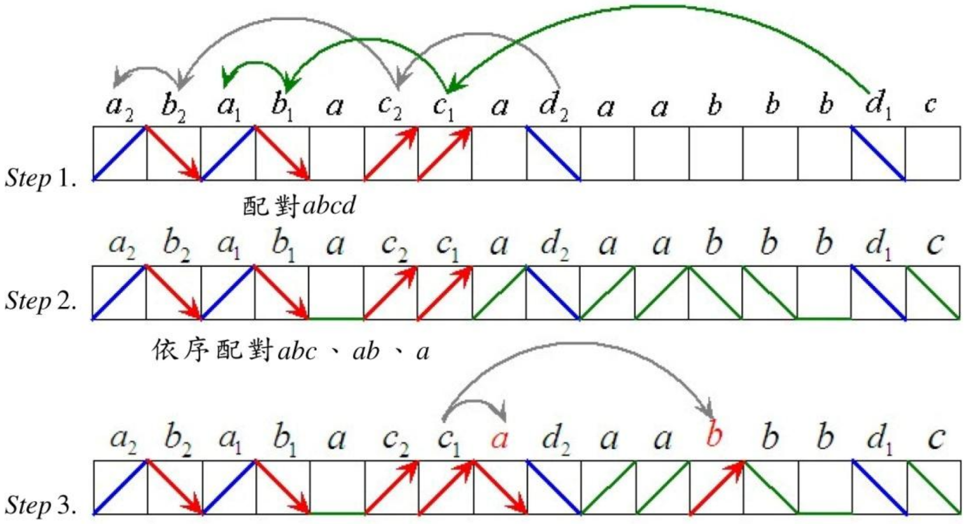
(Ⅱ). 若向量$3(c)$的左方「步驟(2).(3).所填入的向量1(a)個數= 步驟(2).(3).所填入的向量4(c)與4(b)個數」：將此向量$3(c)$右方最鄰近的向量1(a)和向量「4(c)或4(b)」，分別改為向量$2(a)$和「$3(c)$或$3(b)$」。再以此被更改的向量「$3(c)$或$3(b)$」與d 繼續

考慮步驟(5). (*a*) .。

(*b*) . 由右往左，對於目前在步驟(5)中該組*abcd* 的向量*23*(*bc*)，和步驟 (5). (*a*) .(Ⅱ).所置換出的向量*2*(*a*)和「*3*(*c*)或*3*(*b*)」：一個向量*3* 往左找一個向量*2*，其中向量*2* 不可於下次被重複找到，若向量*23*間，「向量*2* 之右方最鄰近的向量5最初來自*a*」，則將向量*2*和此向量5的位置做置換，並以此向量*2*繼續往右考慮，直到此向量*23*間，「向量*2*右方最鄰近的向量5 最初並非來自*a*」為止。此步驟直到所有的向量*23*被考慮為止。

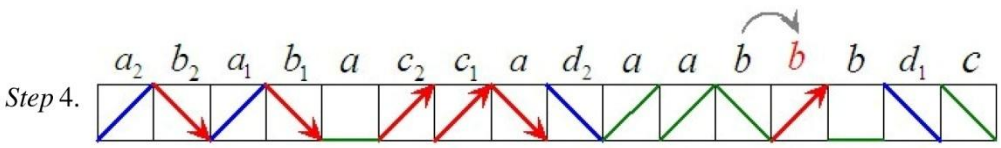
(6). 步驟(5).直到所有 *abcd* 對被考慮為止。

例如:

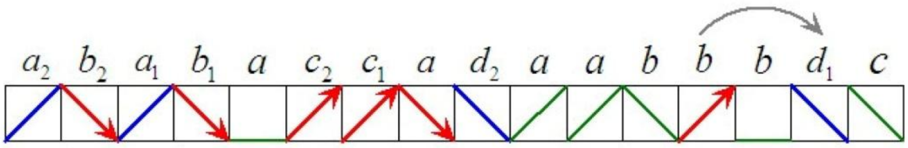


考慮 $a_1b_1c_1d_1$，向量3(c_1)和向量d_1之間存在向量4(b)。又向量3(c_1)

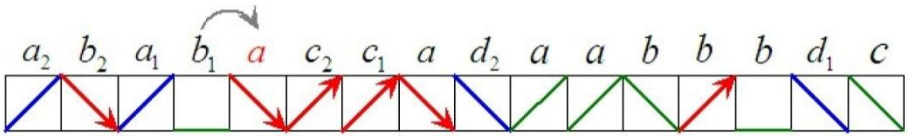
左方於 *Step* 2.填左方於 *Step* 2.填入的「向量 1(*a*)個數」=「向量 4(*b*)與 4(*c*)個數」，故將向量 3(c_1)右方鄰近的向量 1(*a*)與 4(*b*)分別更改為向量 2(*a*)和 3(*b*)



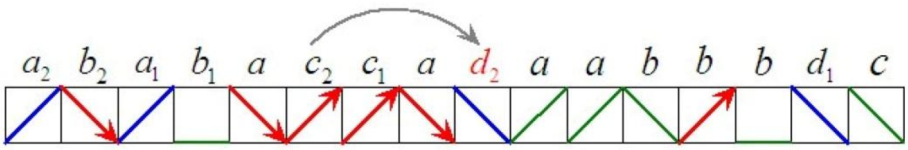
以*Step* 3.被更改的向量3(b)繼續考慮。向量3(b)和d_1之間存在向量4(b)，又向量3(b)左方於*Step* 2.填入的「向量1(a)個數」>「向量4(b)與4(c)個數」，故將向量3(b)與其右方鄰近的向量4(b)進行置換。

Step 5. 

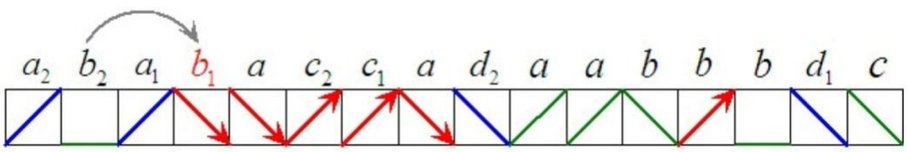
以*Step* 4.被更改的向量3(b)繼續考慮。向量3(b)和d_1之間不存在向量4(b)或4(c)，故不進行置換。

Step 6. 

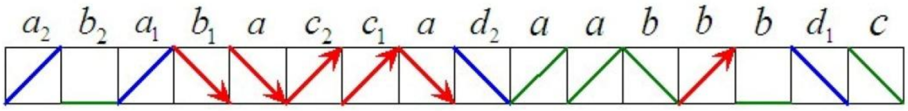
考慮$a_1\,b_1\,c_1\,d_1$ 原本的向量23($b_1\,c_1$)，和*Step* 3.所置換出的向量23(ab)。其中，向量2(b_1)與3(c_1)之間，存在向量5(a)。故將向量2(b_1)與其右方鄰近的向量 5(a)進行置換。

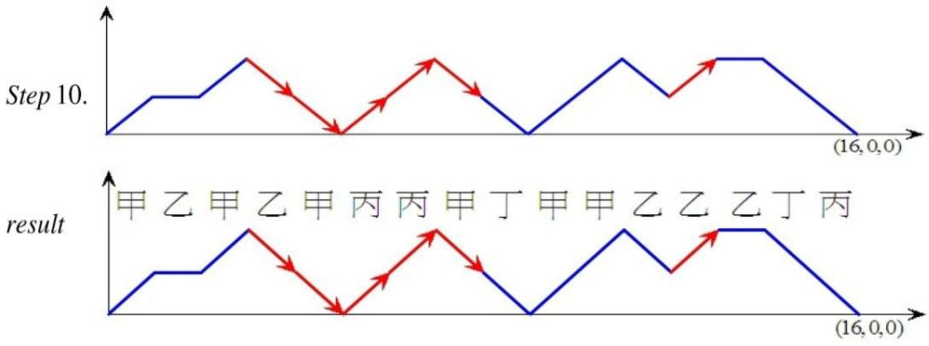
Step 7. 

考慮$a_2\,b_2\,c_2\,d_2$ ，向量3(c_2)和d_2 之間不存在向量4(b)或4(c)，故不進行置換。

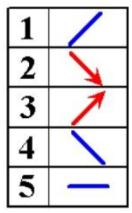
Step 8. 

考慮$a_2\,b_2\,c_2\,d_2$ 原本的向量23($b_2\,c_2$)，之間存在最初來自於a的向量5，故將向量3(b_2)和5 進行置換。

Step 9. 



2. 「立體*Motzkin*」路徑到甲乙丙丁四人「一路領先」得票過程(以下為方便表示，將向量(1,1,0)、(1,-1,1)、(1,1,-1)、(1,-1,0)、(1,0,0)分別以向量1，2，3，4，5 表示，四人得票情形分別以a、b、c、d 表之，且區間$[\alpha,\beta]$表在立體空間中，$x=\alpha$ 和$x=\beta$兩平面所夾中間部分的空間。而「特殊立體對」由於性質特殊，故須特別考慮。)



(1). 向量1 和位於x 軸上的向量5 直接定為a

(2). 由左往右考慮尚未對應到得票過程的路徑向量。

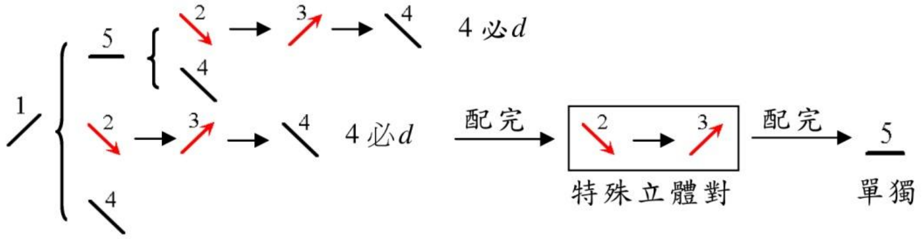
(a). 遇到連續向量2時，套用規則(Ⅰ)。

(b). 遇到連續向量5時，套用規則(Ⅱ)。

(c). 遇到連續向量3時，套用規則(Ⅲ)。

(d). 遇到連續向量4時，套用規則(Ⅳ)。

(Ⅰ). 對於連續數個向量2所組成的區間$[a,b]$：在區間$[0,b]$由左往右進行配對，先遇到先配，如圖（十四）。



圖（十四）

配對完後，若區間$[a,b]$中尚有未被配對到的向量2，且其位於區間$[a',b]$，則其為「特殊立體對」之「左半」，且強制將其對應為a。再考慮區間$[a,a']$的對應：

(i). 由區間$[a,a']$往左找一最鄰近的向量5(a)或2(a)，其中向量2(a)不可為「特殊立體對」。設此向量位於區間$[b'-1,b']$，且在區間$[b', a]$中，非「特殊立體對」的a比非「特殊立體對」的b多k個：

①. 若區間$[a, a']$的向量2個數$\geq k$時，則在區間$[a, a']$上由左至右填入k個b，剩餘者填a 。

②. 若區間$[a, a']$的向量2個數$< k$時，則在區間$[a, a']$的向量2全部填b 。

(ii). 由區間$[a,a']$往左找，若無向量 5(a)或非特殊立體對的向量2(a)時，則替換成往左找一始點位於 x 軸的向量 1，設此向量 1 位於區間$[c',c'+1]$上，且在區間$[c', a]$中非「特殊立體對」的 a 比非「特殊立體對」的 b 多 k'個時，再往下討論，其情形與上①. ② .同。

(Ⅱ). 對於連續數個向量5 所組成的區間$[c,d]$：同(Ⅰ).(i).和(Ⅰ). (ii).

(Ⅲ). 對於連續數個向量3 所組成的區間$[e, f]$：在區間$[0, f]$進行配對，先遇到先配，如上頁圖（十四）。配對完後，若區間$[e, f]$中有被「特殊立體對」之「左半」配對到的向量3，且其位於區間$[e', f]$，則其為「特殊立體對」之「右半」且強制將其對應為b。再考慮區間$[e, e']$的對應：

(i). 由區間$[e, e']$往左找一最鄰近的向量4(b)或3(b)，其中向量3(b)不可為「特殊立體對」。設此向量位於區間$[f'-1, f']$，且在區間$[f', e]$中，非「特殊立體對」的b比c多l個：

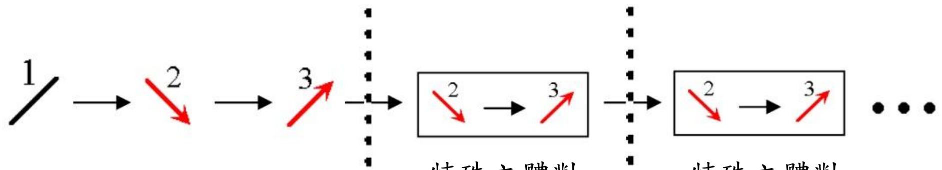
①. 若區間$[e, e']$的向量3個數$\geq l$時，則在區間$[e, e']$上由左

至右填入l個c ，剩餘者填b 。

②. 若區間$[e, e']$的向量3個數$< l$時，則在區間$[e, e']$的向量3全部填c 。

(ii). 由區間$[e, e']$往左找，若無向量4(b)或非「特殊立體對」的向量3(b)時，則替換成往左找一始點位於x軸的向量1，設此向量1位於區間$[g', g'+1]$上，且在區間$[g', e]$中非「特殊立體對」的b比c多l'個時，再往下討論，其情形與上①.②同。

(Ⅳ). 對於連續數個向量4所組成的區間$[g, h]$：在區間$[0, g]$由左往右進行配對，如圖（十五）。



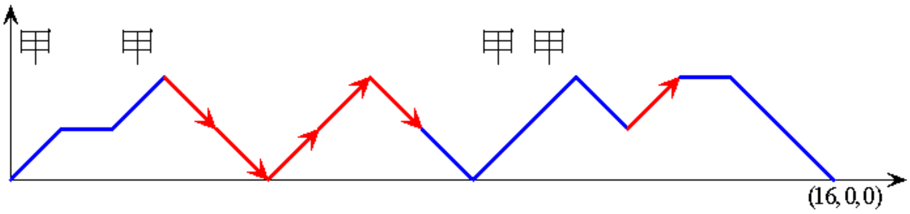
圖(十五)

針對配對的每一組，若其在如圖（十五）的節線處尚未配到d，且其沒有配對「特殊立體對」或其配對的「特殊立體對」為完整（即非只有左半），則在區間$[g, h]$左方填入一個d。考慮完能填d的個數後，區間$[g, h]$若有剩餘的向量4位於$[g', h]$，則考慮區間$[g', h]$的對應：同(Ⅲ). (i).和(Ⅲ). (ii).。

由此，對於任意一個限制(1,0,0) 只能在xy平面上出現的「立體*Motzkin* 」路徑，均可找到一個且唯一的甲乙丙丁「一路領先」得票過程與之對應。因此，從(0,0,0)走到(n,0,0)，規定走法為依循向量(1,0,0)、(1,1,0)、(1,−1,0)、(1,1,−1)或(1,−1,1)，且走法限於空間（$x\geq 0$且$y\geq 0$且$z\geq 0$），並規定向量(1,0,0)只能出現在xy 平面上的方法數為給定票數n 之甲乙丙丁四人「一路領先」方法數，即為L_n，由文獻知$L_n = C_{\left[\frac{n}{2}+\frac{1}{2}\right]} C_{\left[\frac{n}{2}+1\right]}$，其中$C_k = \frac{1}{k+1}\binom{2k}{k}$即*Catalan*數。

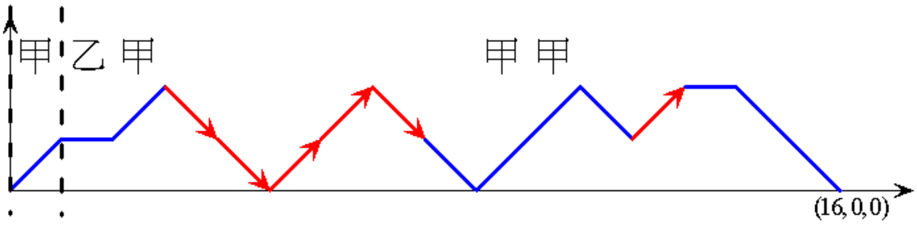
例如：

Step 1



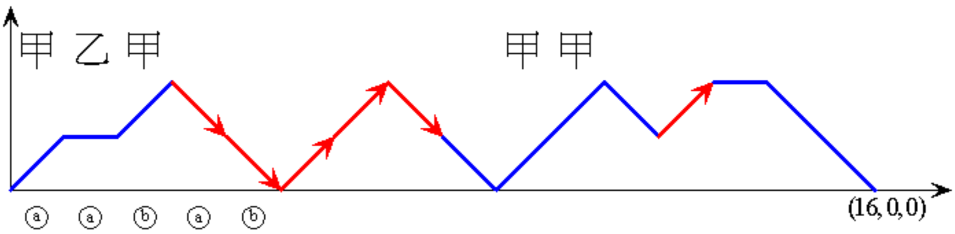
向量1和位於x軸上的向量5直接定為a

Step 2



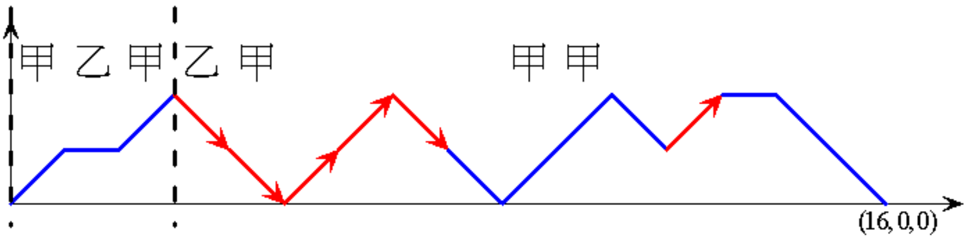
對於連續向量5，往左找一最鄰近的向量5(a)或2(a)，其中向量2(a)不可為「特殊立體對」。若無向量5(a)或非「特殊立體對」的向量2(a)時，則替換成往左找一始點位於x 軸的向量1，考慮區間。區間中非「特殊立體對」的a 比非「特殊立體對」的b 多1 個，則由左往右填入一個b，剩餘填a。

Step 3



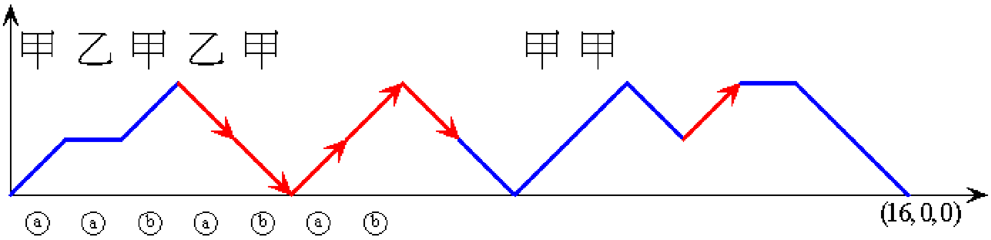
遇到連續向量2，要由左往右配對。圖中兩向量2均有被配對，故均非「特殊立體對」。

Step 4



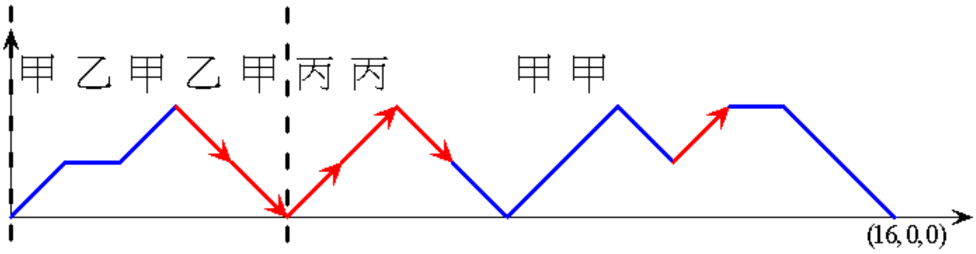
對於非「特殊立體對」的連續向量2，往左找一最鄰近的向量5(a)或2(a)，其中向量2(a)不可為「特殊立體對」。若無向量5(a)或非「特殊立體對」的向量2(a)時，則替換成往左找一始點位於x 軸的向量1，考慮區間。區間中非「特殊立體對」的a比非「特殊立體對」的b多1個，則由左往右填入一個b，剩餘填a 。

Step 5



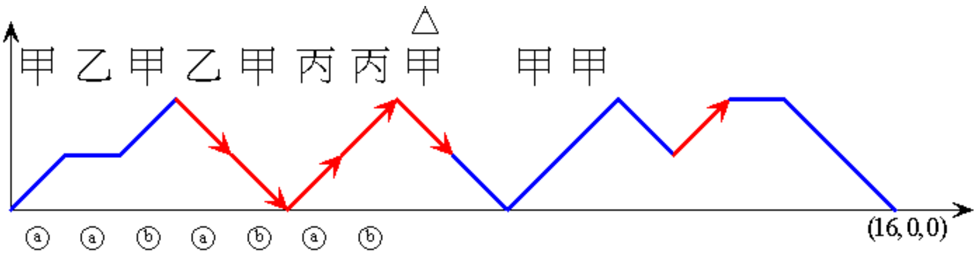
遇到連續數個向量3，要由左往右配對。圖中兩向量均未被「特殊立體對」之「左半」配到，故均非「特殊立體對」。

Step 6



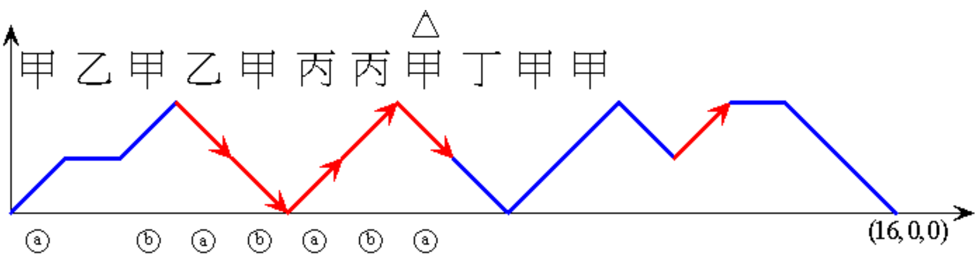
對於非「特殊立體對」的連續向量3，往左找一最鄰近的向量4(b)或3(b)，其中向量3(b)不可為「特殊立體對」。若無向量4(b)或非「特殊立體對」的向量3(b)時，則替換成往左找一始點位於x軸的向量1，考慮區間。區間中非「特殊立體對」的b比c多2 個，則由左往右填入一個c，剩餘填b 。

Step 7



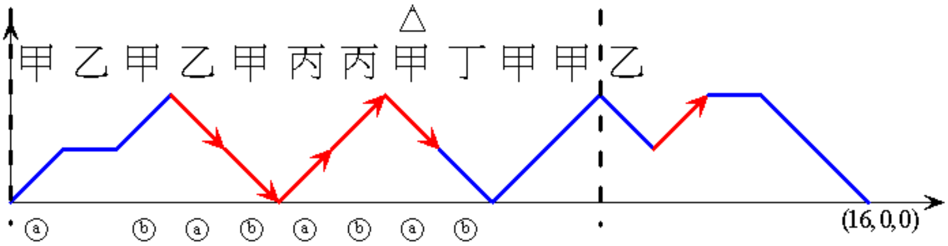
遇到連續向量2，要由左往右配對。圖中向量2未有被配對，故其為「特殊立體對」之「左半」，強制定其對應為a。

Step 8



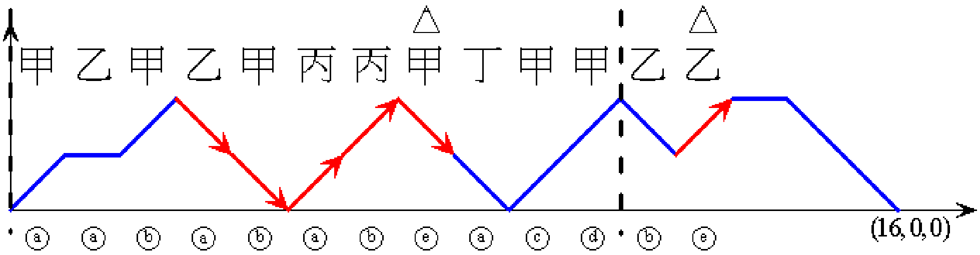
遇到連續數個向量4，要由左往右配對。其中組ⓐ的「特殊立體對」不完整，不可填d。組ⓑ沒有配對「特殊立體對」，可以填d，故此連續向量4 可以填入一個d 。

Step 9



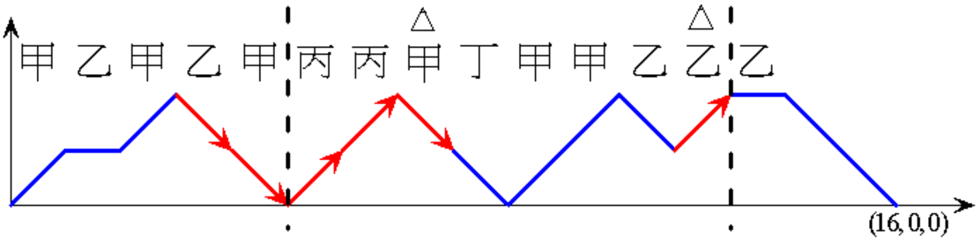
遇到連續數個向量4，要由左往右配對。其中組ⓐ的「特殊立體對」不完整，不可填d。組ⓑ配對到d，不可填d。故此連續向量4無法填入d。對於不能填d的連續向量4，往左找一最鄰近的向量4(b)或3(b)，其中向量3(b)不可為「特殊立體對」。若無向量4(b)或非「特殊立體對」的向量3(b)時，則替換成往左找一始點位於x軸的向量1，考慮區間。區間中非「特殊立體對」的b比c多0個，則由左往右填入0個c，剩餘填b 。

Step 10



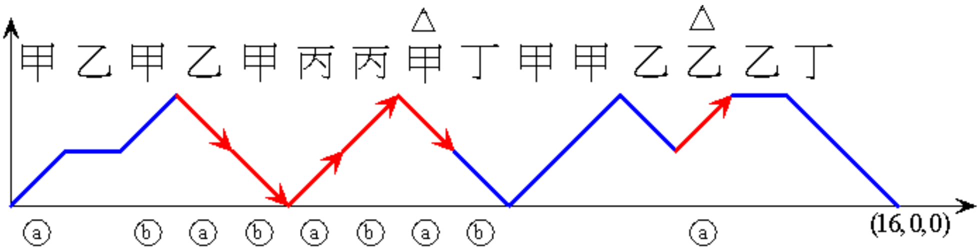
遇到連續向量3，要由左往右配對。圖中向量3 被「特殊立體對」之「左半」配對，故其為「特殊立體對」之「右半」，且強制定其對應為b。

Step 11



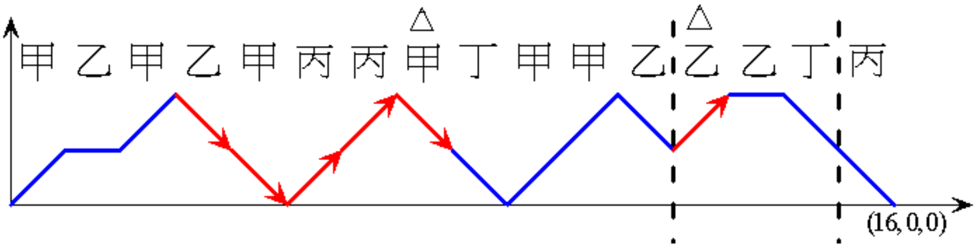
對於連續向量5，往左找一最鄰近的向量5(a)或2(a)，其中向量2(a)不可為「特殊立體對」。若無向量5(a)或非「特殊立體對」的向量2(a)時，則替換成往左找一始點位於x軸的向量1，考慮區間。區間中非「特殊立體對」的a比非「特殊立體對」的b多1個，則由左往右填入一個b，剩餘填a 。

Step 12



遇到連續數個向量4，要由左往右配對。其中組ⓐ的「特殊立體對」完整，可以填d。組ⓑ已配對到d，不可填d，故此連續向量4可以填入1個d 。

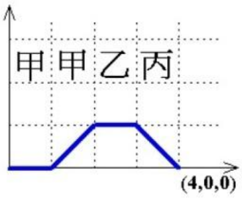
Step 13

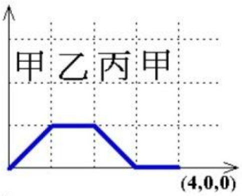


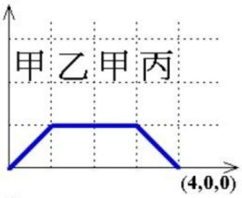
對於不能填 d 的連續向量 4，往左找一最鄰近的向量 4(b)或 3(b)，其中向量 3(b)不可為「特殊立體對」。考慮區間。區間中非「特殊立體對」的 b 比 c 多 1 個，則由左往右填入一個 c，剩餘填 b。

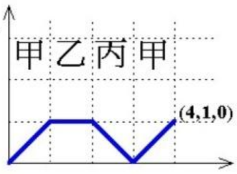
(二). 仿 *Regev's Conjecture* 。由於已經架構出「立體 *Motzkin* 」路徑，同理亦可得到 $L_{2,1,1,0}(n)=L_{n-2}-L_{n-4}$ ，其中 $L_{2,1,1,0}(n)$ 為前四票中，甲得兩票，乙得一票，丙得一票，再給 $(n-4)$ 票四人的「一路領先」方法數。

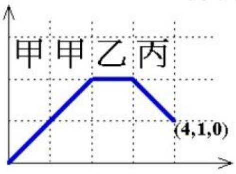
證明：由「立體 *Motzkin*」路徑，我們可將甲得兩票，乙得一票，丙得一票的情形分成14 種情況：

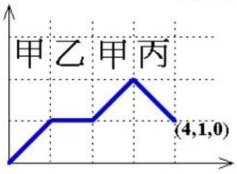
(1). 

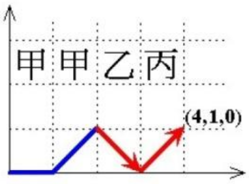
(2). 

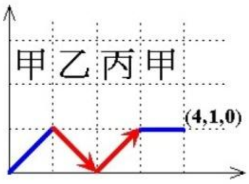
(3). 

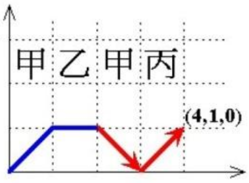
(4). 

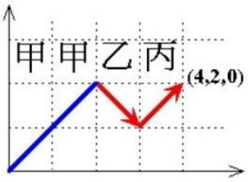
(5). 

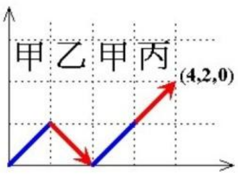
(6). 

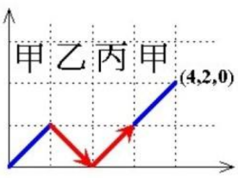
(7). 

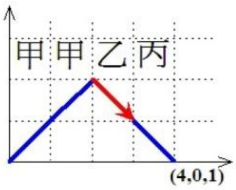
(8). 

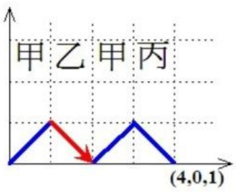
(9). 

(10). 

(11). 

(12). 

(13). 

(14). 

(1)(2)(3) 暫止於(4,0,0) ，其方法數為L_{n-4}。(4)(5)(6)(7)(8)(9) 暫止於(4,1,0) ，其方法數為L_{n-3}- L_{n-4}。(10)(11)(12) 暫止於(4,2,0) 。(13)(14) 暫止於(4,0,1)。對於(1)(2)(3)，(4)(5)(6)，(7)(8)(9)，(10)(11)(12)，(13)(14)四組，其後面的「立體*Motzkin* 」路徑若相同，則同組之後續對應的「一路領先」得票順序相同。各組中各種狀況應視為同類。又暫止於(4,2,0) 的方法數和暫止於(4,0,1) 的方法數之總和為 L_{n-2} -L_{n-3}，得到：

$L_{2,1,1,0}(n) = (L_{n-2} - 2L_{n-3}) + 2(L_{n-3} - L_{n-4}) + (L_{n-4}) = L_{n-2} - L_{n-4}$，其中 $L_{2,1,1,0}(n)$ 為前四票中，甲得兩票，乙得一票，丙得一票，再給$(n-4)$票四人的「一路領先」方法數。

(三). 由數列搜尋器（*The Online Encyclopedia of Integer Sequences*），發現以下的遊戲：「*Number of underdiagonal lattice paths in the first quadrant, going from* (0,0) *to a point on the x-axis and consisting of* $n+1$ *steps from* $\{E = (1,0), W = (-1,0), N = (0,1), S = (0,-1)\}$ 」所形成的數列，與四人「一路領先」所形成的數列一樣！因此依照「立體*Motzkin* 」路徑的對應方法，來印證兩者的關聯！

證明：因為遊戲規則需由$(0, 0)$ 走到$(n, 0)$ ，其中$n \in N$ 。

E	→
N	↑
S	↓
W	←

情況一：若出現一種向量走法，必為$E = (1, 0)$ ，因此定E 為甲得票。

情況二：若出現兩種向量走法，必為$E = (1, 0)$ 與$W = (-1, 0)$ ，因此定EW為甲乙得票。

情況三：若出現三種向量走法，必為$E = (1, 0)$，$N = (0,1)$，$S = (0, -1)$，因此定ENS 為甲乙丙得票。

情況四：若出現四種向量走法，即為$E = (1, 0)$，$N = (0,1)$，$S = (0, -1)$ 與$W = (-1, 0)$ ，因此定$ENSW$ 為甲乙丙丁得票。

(於此規定甲乙丙丁得票以*abcd* 呈現)

1. 由「一路領先」得票順序到*ENSW*

(1). 由右往左，考慮d→由此d繼續往左考慮，找到最鄰近此d的c→同理，找到最鄰近c的b 和最鄰近b的a 。

(2). 由右往左，考慮未在(1).中的c→由右往左繼續考慮，找到最鄰近此c的b→同理，找到最鄰近此b的a。

(3). 將(1).找到的$abcd$ 組定為$ENSW$ 以及由(2).找到的abc組定為ENS 。

(4). 考慮剩下的a,b，分別定為E,W 。

(5). 由右往左依序考慮每個S，若其右方不存在W，則此S 考慮結束，繼續考慮其左方之下一個S。若其右方最鄰近的W不為d，則考慮步驟(4).所填入的E,W ：

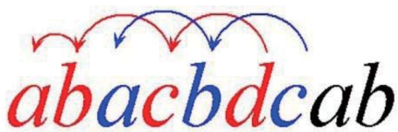
(a). 若S的左方「E 的個數>W 的個數」：將此S 和其右方最鄰近不為d的W 位置置換。並以此S 繼續考慮其右方。

(b). 若S 的左方「E 的個數=W 的個數」：將此S 右方最鄰近S 的E 和W ，分別改為N 和S 。再以此被更改的S 繼續考慮其右方。

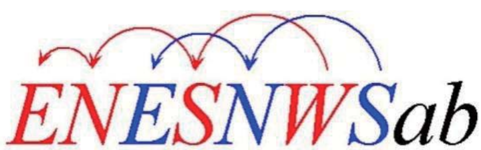
(c). 在(a) (b) 的過程中，若S 右方最鄰近的W 為d ，則此S 置換步驟結束，並於考慮下一個S 的過程中不得再考慮此W 。

(6). 若步驟(5).考慮完所有的S ，則此對應完成。例：$abacbdcab$

(a). 考慮上述對應步驟(1).、(2).，配組如下：



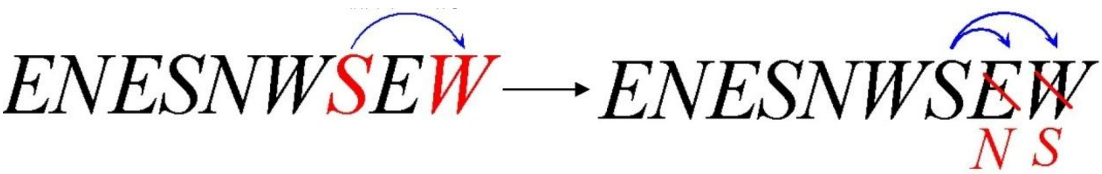
(b). 將其換成 ENSW 及 ENS ，如下：



(c). 考慮上述對應步驟(4).，定 a 為 E 、b 為 W ：

(d). 考慮上述對應步驟(5).：

非 d 的 W 前面的 EW 一樣多



(e). 繼續考慮上述對應步驟(5).：

此 W 為 d，「不置換」!

(f). 所以 abacbdcab 的答案即為 ENESNWSNS 。

2. 由$ENSW$ 到「一路領先」得票順序

(1). 由左往右，依序考慮每個向量（即$ENSW$ ）

(a) . 若為E，定其為a 。

(b) . 若為N ，往右找離此N 最近且未被配對的S 配為一組：

(Ⅰ). 若此NS 組的N 左方存在未被配對的E ，則由左往右找一個未被配對的E與此組NS 配成ENS組，並對應為abc 。

(Ⅱ). 若此NS 組的N 左方不存在未被配對的E，將此組對應為ab。

(c). 若為W，考慮其左方所存在完整的ENS 組數和包夾著此W的$NS(ab)$ 組數。

(Ⅰ). 若「ENS 組數」>「包夾著此W的$NS(ab)$組數」，定此W為d，並由左往右找一個未被配對的$S(c)$ 與其配對。且此$S(c)$所存在的ENS 組不得於下次繼續考慮步驟(c).。

(Ⅱ). 若「ENS組數」=「包夾著此W的$NS(ab)$ 組數」，定此W為b，並由左往右找一個未被配對的$E(a)$ 與其配對。且此$E(a)$不得於下次繼續考慮步驟(b).(c).。

(d). 步驟(a).(b).(c).對應完後，由左往右考慮每個$c(S)$，若其左方離此$c(S)$ 最近

的b不為N，則將此兩得票順序置換，考慮至其左方最近的b為N為止，並於考慮下個$c(S)$ 時不得考慮此$b(N)$。考慮完所有$c(S)$ 即結束對應。

例如：

Step 1：由左往右，依步驟(*a*) .定為 *a* 。

Step 2：遇到N，往後找最近的S。前面有多餘的a，依步驟(*b*). (Ⅰ).對應為*abc* 。

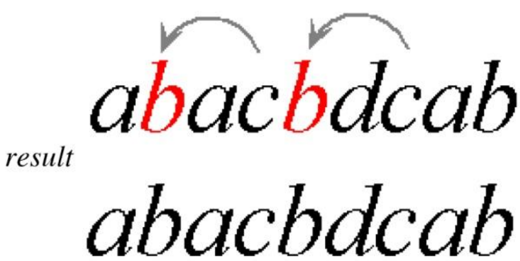
Step 3：同*Step* 1，定E為a。

Step 4：同*Step* 2，N往後找最近的S，前有未配對的a，依步驟(*b*). (Ⅰ).對應為*abc*。

Step 5：遇到W，前有一組*ENS*組，不存在*NS*(*ab*)包夾著此W，依步驟 (*c*). (Ⅰ).對應為d 。

Step 6：遇到N，往後找一最近的S，前無未配對的a，依步驟(*b*). (Ⅱ).對應為*ab*。

Step 7：考慮步驟(*d*).，每個c 往前找到$b(N)$ 即停止。



伍、研究結果

一、對二維的坐標格子(m,n) ，定義蟲子生殖的遞迴關係$W_{m,n}=W_{m-1,n}\times W_{m,n-1}$，其中$m,n\in N\cup\{0\}$ 。

(一). 1. 對任意坐標格子0 0 (m,n) ，我們有$W_{m0,n0}=(-1)^{C_{m0}^{m0+n0}}$，其中$m_0,n_0\in N\cup\{0\}$，$W_{0,0}=-1$。

2. 若$C_{m_0}^{m_0+n_0}$為奇數，則此坐標格子有蟲子； 若$C_{m_0}^{m_0+n_0}$為偶數，則此坐標格子蟲子不存在。

(二). 由蟲子生殖遞迴關係導出的冪次性質，恰與巴斯卡定理$C_m^{m+n}=C_{m-1}^{m+n-1}+C_m^{m+n-1}$形成對應，其中$m,n\in N$ 。

(三). $W_{m,n}=(-1)^{C_m^{m+n}}$，其中$m,n\in N\cup\{0\}$ 。冪次值即從原點$(0,0)$ 走到(m,n) 之捷徑走法數。

(四). 標示出蟲子存在的位置，形成一美麗的「自我碎形」圖案。

二、對三維空間中坐標格子(m,n,l) ，定義蟲子生殖的遞迴關係

$W_{m,n,l}=W_{m-1,n,l}\times W_{m,n-1,l}\times W_{m,n,l-1}$，其中$m,n,l\in N\cup\{0\}$， $W0,0,0=-1$

(一). 1. 對任意坐標格子 (m_0,n_0,l_0) ，我們有$W_{m0,n0,l0}=(-1)^{\binom{m_0+n_0+l_0}{m_0,\ n_0,\ l_0}}$

其中$m_0,n_0,l_0\in N\cup\{0\}$

2. 若$\binom{m_0+n_0+l_0}{m_0,\ n_0,\ l_0}$為奇數，則此坐標格子存在蟲子；

若$\binom{m_0+n_0+l_0}{m_0,\ n_0,\ l_0}$為偶數，則此坐標格子中蟲子不存在。

(二). 由蟲子生長的遞迴關係導出的冪次性質，我們仿二維，恰與「立體」巴斯卡定理：

$$\binom{m+n+l-1}{m-1,\ n,\ l}+\binom{m+n+l-1}{m,\ n-1,\ l}+\binom{m+n+l-1}{m,\ n,\ l-1}=\binom{m+n+l}{m,\ n,\ l}$$

形成對應，其中$m,n,l\in N\cup\{0\}$。

(三). $W_{m,n,l}=(-1)^{\binom{m_0+n_0+l_0}{m_0,\ n_0,\ l_0}}$，其中$m,n,l\in N\cup\{0\}$。冪次值即從原點(0,0,0)走到$(m,n,l)$之捷徑走法數。

三、根據二維及三維的公式及空間關係，我們推論n維空間內也存在著規律。對坐標格子$(a_1, a_2, \ldots, a_n)$，定義蟲子生殖的遞迴關係$W_{a1, a2,\ldots,an}= W_{a1-1, a2,\ldots,an}\times W_{a1, a2-1,\ldots,an}\times\ldots\times W_{a1, a2,\ldots,an-1}$，其中$a_1, a_2, \ldots, a_n\in N\cup\{0\}$，$W_{0,0,\cdots,0} = -1$

(一). 1. 對任意坐標格子 $(a_1', a_2',\ldots, a_n')$，我們有

$$W_{a1', a2',\ldots, an'}=(-1)^{\binom{m_0+n_0+l_0}{a1,\ a2,\ldots,\ an-1}}$$，其中$a_1', a_2',\ldots, a_n'\in N\cup\{0\}$

2. 若$\binom{a_1'+a_2'+\cdots+a_n'}{a_1',\ a_2',\ \cdots,\ a_{n-1}',\ a_n'}$為奇數，則此坐標格子中存在蟲子；

若$\binom{a_1'+a_2'+\cdots+a_n'}{a_1',\ a_2',\ \cdots,\ a_{n-1}',\ a_n'}$為偶數，則此坐標格子中蟲子不存在。

(二). 由蟲子生長的遞迴關係導出的冪次性質，我們仿二維、三維，恰與「n維」巴斯卡定理

$$\binom{a_1+a_2+\cdots+a_n-1}{a_1-1,\ a_2,\ \cdots,\ a_{n-1},\ a_n}+\binom{a_1+a_2+\cdots+a_n-1}{a_1,\ a_2-1,\ \cdots,\ a_{n-1},\ a_n}+\cdots+\binom{a_1+a_2+\cdots+a_n-1}{a_1,\ a_2,\ \cdots,\ a_{n-1},\ a_n-1}=\binom{a_1+a_2+\cdots+a_n}{a_1,\ a_2,\ \cdots,\ a_{n-1},\ a_n}$$

形成對應，其中$a_1, a_2,\ldots,a_n\in N\cup\{0\}$

(三). $W_{a1, a2,\ldots,an}=(-1)^{\binom{a_1+a_2+\cdots+a_n}{a_1,\ a_2,\ \cdots,\ a_{n-1},\ a_n}}$，其中$a_1, a_2,\ldots,a_n\in N\cup\{0\}$。冪次值即為從原點(0,0,⋯,0) 走到 $(a_1, a_2,\ldots,a_n)$之捷徑走法數。

四、改變蟲子所走的格子形狀，對二維平面中任意坐標格子(m, n) 恆有$m\geq n$之關係。

定義蟲子生殖的遞迴關係$T_{m,n} = T_{m-1,n}\times T_{m,n-1}$，$T_{0,0} = -1$，其中$m,n\in N\cup\{0\}$

(一). 1. 對任意坐標格子(m_0, n_0)，我們有 $T_{m0,n0}=(-1)^{C_{n0}^{m0+n0}-C_{n0-1}^{m0+n0}}$，其中 $m_0\geq n_0$ 且$m_0, n_0\in N\cup\{0\}$。

2. 若$C_{n0}^{m0+n0}-C_{n0-1}^{m0+n0}$為奇數，則此坐標格子存在蟲子；若$C_{n0}^{m0+n0}-C_{n0-1}^{m0+n0}$為偶數，則此坐標格子中蟲子不存在。

(二). 1. $T_{m,n}=(-1)^{C_n^{m+n}-C_{n-1}^{m+n}}$，其中$m_0, n_0 \in N\cup\{0\}$ 。其冪次值即為從原點(0,0) 走到 (m,n) 之捷徑走法數。我們可應用其於求一路領先的方法數上。

2. 若限制 $m_0-d\geq n_0$，我們令$m'=m_0-d$，$n'=n_0$，$T_{m',n'}$ 即為「一路領先」之方法數。

(三). 1. 若給定票數n，其中$n\in N\cup\{0\}$，則甲得票數「一路領先」乙得票數之方法總數為$C^{n}_{\left[\frac{n}{2}\right]}$

2. 若A_n為給定票數n的兩人「一路領先」方法總數，則其遞迴式為$A_{n+1}=\dfrac{2n+3+(-1)^{n+1}}{n+2}A_n$，其中$n\in N$。

3. 在Motzkin 路徑中，若限制向量(1,0) 走法(即僅可在x 軸上出現)，會與二人「一路領先」得票過程形成一一對應。

五、改變蟲子所走的格子形狀，對三維空間中任意坐標格子(m, n, l) 恆有$m\geq n\geq l$之關係，定義蟲子生殖的遞迴關係$T_{m,n,l}=T_{m-1,n,l}\times T_{m,n-1,l}\times T_{m,n,l-1}$，其中$m, n, l\in N\cup\{0\}$。

$$T_{0,0,0}=-1\text{，其中}m, n, l\in N\cup\{0\}$$

(一). 對任意坐標格子$(m0, n0, l0)$ ，我們有

$$T_{m_0,n_0,l_0}=(-1)^{\binom{m_0+n_0+l_0}{m_0,\ n_0,\ l_0}-\binom{m_0+n_0+l_0}{m_0,\ n_0+1,\ l_0-1}+\binom{m_0+n_0+l_0}{m_0+1,\ n_0+1,\ l_0-2}-\binom{m_0+n_0+l_0}{m_0+1,\ n_0-1,\ l_0}}$$
$$\times(-1)^{\binom{m_0+n_0+l_0}{m_0+2,\ n_0-1,\ l_0-1}-\binom{m_0+n_0+l_0}{m_0+2,\ n_0,\ l_0-2}}$$

其中$m_0\geq n_0\geq l_0$ 且$m_0, n_0, l_0\in N\cup\{0\}$

(二). 承(一).：

若Tm_0, n_0, l_0之冪次值為奇數，則此坐標格子中存在蟲子；

若Tm_0, n_0, l_0 之冪次值為偶數，則此坐標格子中蟲子不存在。

(三). 承(一).，對任意坐標格子(m,n,l) ，其中$m,n,l\in N\cup\{0\}$，$T_{m,n,l}$冪次值即為從原點 **(0,0,0)** 走到(m, n, l) 之捷徑走法數。我們可應用其於求「一路領先」之方法數上。

(四). 若限制$m_0-d_1\geq n_0$ 且$n_0-d_2\geq l_0$，我們令$m'=m_0-d_1-d_2$，$n'=n_0-d_2$，$l'=l_0$，$T_{m',n',l'}$，即為「一路領先」之方法數。

(五). 1. 若給定票數n，其中$n\in N\cup\{0\}$，則甲得票數「一路領先」乙得票數，乙得票數「一路領先」丙得票數之得票過程與由(0,0) 走到$(n,0)$ 的*Motzkin*路徑形成一一對應。

2. 參考文獻可得給定票數n，$n\in N\cup\{0\}$之三人「一路領先」其方法總數 $M_n=\sum_{k\geq 0}\binom{n}{2k}C_k$，其中$C_k=\dfrac{1}{k+1}\binom{2k}{k}$即*Catalan* 數。

六、改變蟲子所走的格子形狀，對n維空間中任意坐標格子$(a_1, a_2, \cdots. a_n)$ 恆有$a_1\geq a_2\geq\cdots\geq a_n$之關係，定義蟲子生殖的遞迴關係$T_{a1,a2,\ldots,an}=T_{a1-1,a2,\ldots,an}\times T_{a1,a2-1,\ldots,an}\times T_{a1,a2,\ldots,an-1}$，其中$a_1, a_2, \cdots a_n\in N\cup\{0\}$。

(一). 對任意坐標格子$(a_1, a_2, \ldots, a_n)$，若令

$$D_n=(a_1'+a_2'+\cdots+a_n')!\begin{vmatrix} \frac{1}{a_1'!} & \frac{1}{(a_1'+1)!} & \cdots & & \frac{1}{[a_1'+(n-1)]!} \\ \frac{1}{(a_2'-1)!} & \frac{1}{a_2'!} & \cdots & & \frac{1}{[a_2'+(n-2)]!} \\ \vdots & \vdots & & \vdots & \vdots \\ & & \cdots & \frac{1}{a_{n-1}'!} & \frac{1}{(a_{n-1}'+1)!} \\ \frac{1}{[a_n'-(n-1)]!} & \frac{1}{[a_n'-(n-2)]!} & \cdots & \frac{1}{(a_n'-1)!} & \frac{1}{a_n'!} \end{vmatrix}$$

我們有$T_{a'_1, a'_2, \ldots, a'_n}=(-1)^{D_n}$，其中$a_1'\geq a_2'\geq\ldots\geq a_n'$且$a_1', a_2', \ldots, a_n'\in N\cup\{0\}$。

（二）. 若D_n為奇數，則此坐標格子中存在蟲子；若D_n為偶數，則此坐標格子中蟲子不存在。

(三). $T\,a'_1, a'_2, \ldots, a'_n = (-1)^{D_n}$，其中$a'_1, a'_2, \ldots, a'_n$ $a_n' \in N \cup \{0\}$。冪次值即為從原點(0,0, ⋯,0) 走到$(a_1', a_2', \ldots, a_n')$之捷徑走法數。我們可應用其於求「一路領先」之方法數上。

七、利用圖形對應，證明出「一路領先」得票順序與*Motzkin*路徑的對應關係與應用。

(一). 1. 利用圖形對應，證明三人「一路領先」和*Motzkin* 路徑方法數相同。

2. 利用圖形對應，證明兩人「一路領先」和限制向量(1,0) 只能出現在x軸上的*Motzkin* 路徑方法數相同。

(二). 利用圖形對應，證明四人「一路領先」和從(0,0,0) 走到(n,0,0)，規定走法為依循向量(1,0,0)、(1,1,0)、(1,−1,0)、(1,1,−1) 或(1,−1,1)，且走法限於空間（$x\geq 0$且$y\geq 0$且$z\geq 0$），並規定向量(1,0,0) 只能出現在xy平面上的方法數相同。

(三). 利用圖形對應解決*Regev's Conjecture* :「若前四票得票為甲兩票、乙一票，再給$(n-3)$票之一路領先方法數為M_n-1-M_n-3，其中M_n是*Motzkin*數」。

(四). 利用四人「一路領先」與「立體*Motzkin* 」路徑的對應，證明：

$L_{2,1,1,0}(n) = (L_{n-2} - 2L_{n-3}) + 2(L_{n-3} - L_{n-4}) + (L_{n-4}) = L_{n-2} - L_{n-4}$，其$L_{a,b,c,d}(n)$為四人「一路領先」甲已得$a$票，乙已得$b$票，丙已得$c$票，丁已得$d$票，再給$n-(a+b+c+d)$票的「一路領先」得票方法數，其中$a\geq b\geq c\geq d$。

(五). 利用對應證明四人「一路領先」方法數和遊戲「*Number of underdiagonal lattice paths in the first quadrant, going from* **(**0,0**)** *to a point on the x-axis and consisting of n* +1 *steps from* **{** E = **(**1,0**)** , W = **(**−1,0**)** , N = **(**0,1**)** , S = **(**0,−1**)** **}** 」的方法數相同，並得出之間存在的1−1*onto* 對應關係。

陸、討論

已知兩人「一路領先」與*Motzkin*路徑中，限制向量$(1,0)$ 只能出現在x軸的路徑形成對應。三人「一路領先」與*Motzkin* 路徑形成對應。四人「一路領先」與「立體*Motzkin*」路徑中，限制向量$(1,0,0)$ 只能出現在xy平面的路徑形成對應。至於五人「一路領先」是經由程式驗證發現，其給定總票數「一路領先」方法數與「立體*Motzkin*」路徑形成對應。

甚至程式驗證，發現給定票數六人與七人「一路領先」與從$(0,0,0,0)$ 走到$(n,0,0,0)$，走法為依循向量$(1,0,0,0)$、$(1,1,0,0)$、$(1,-1,0,0)$、$(1,1,-1,0)$、$(1,-1,1,0)$、$(1,0,1,-1)$ 或$(1,0,-1,1)$，且向量群在行進中的向量終點坐標 (a_1,a_2,a_3,a_4) 必須滿足$a_1,a_2,a_3,a_4\ \geq 0$，的路徑走法數形成對應！因此，猜測應該有以下推論存在，將之稱為$3HTC$ 猜想。

$3HTC$ 猜想：由$2n-1$種n維向量$(1,0,0,0,\cdots,0,0)$，$(1,1,0,0,\cdots,0,0)$，$(1,-1,0,0,\cdots,0,0)$，$(1,1,-1,0,\cdots,0,0)$，$(1,-1,1,0,\cdots,0,0)$，$(1,0,1,-1,\cdots,0,0)$，$(1,0,-1,1,\cdots,0,0)$，$\cdots$，$(1,0,0,\cdots,1,-1,0)$，$(1,0,0,\cdots,-1,1,0)$，$(1,0,0,\cdots,0,1,-1)$，$(1,0,0,\cdots,0,-1,1)$，從$(0,0,\cdots,0)$ 走到$(k,0,\cdots,0)$，且維持向量向量群在行進中的向量終點坐標$(a_1,a_2,\ldots,a_n)$必須滿足$a_i\geq 0$，$\forall i=1,2,\cdots,n$ 條件下，行走的路徑方法數即為給定票數$2n-2$ 與$2n-1$人的「一路領先」方法數。其中$2n-2$人的「一路領先」方法數為限制向量$(1,0,0,0,\cdots,0,0)$ 只能出現在第n維坐標為0的路徑走法數。

每增加一維度即增加兩種向量，並保留原本已存在的向量。所增加的兩個向量除第一維度及最後兩維度外皆為0，最後兩維度分別為1、−1和−1、1。

對於這些向量群，增加維度（即人數）並不會增減原先已存在的向量，代表「一路領先」在較低維度（即人數）的時候仍然成立！以程式驗證到12、13 人「一路領先」時皆正確！因此推論，任意人數的一路領先應符合此猜想！

柒、參考資料

一、余文卿 編著 / 高中數學課本第一冊、高中數學課本第三冊、高中數學課本第四冊龍騰文化事業公司

二、廖思善 編著 / 動手玩碎形 / 天下文化

三、第47 屆科展立體巴斯卡、第45 屆科展*Catalan* 數列的聯想

四、洪維恩 編著 / *Mathematica*5 數學運算大師 / 旗標出版社

五、整數數列搜尋器（*The On − line Encyclopedia of Integer Sequences* ），
http://www.research.att.com/~njas/sequences/

六、*Sierpinski Gasket* http://local.wasp.uwa.edu.au/~pbourke/fractals/gasket/

七、*Sierpinski Gasket and Carpet*
http://www.atlas-zone.com/complex/fractals/classic/sierpinski.htm

八、伯特朗選票問題 / 取自機率名題二則漫談
http://episte.math.ntu.edu.tw/articles/mm/mm_04_4_03/page2.html

九、*Motzkin Numbers* / 取自小素民族
http://hk.geocities.com/goodprimes/OMinor.htm

十、*Motzkin Numbers* / 取自 *Wolfram MathWorld*
http://mathworld.wolfram.com/MotzkinNumber.html

十一、*Sen − Peng Eu* & *Shu −Cheng Liu* &*Yeong − Nan Yeh* / 2006 /*Catalan and Motzkin Numbers* mod *ulo* 4*and*8
http://www.math.sinica.edu.tw/www/file_upload/mayeh/MotzkinMod_f.pdf

十二、王元元 王慶瑞 黃紀麟 編著 / 組合數學-原理及題解 / 中央圖書出版社 / 2000

十三、*Robert Donaghey* & *Louis W. Shapiro* / 1977 / *Motzkin Numbers* /*Journal of Combinatorial Theory* / *Series*A23 / *P*.291− *P*.301

十四、*Proof of a Conjecture of Amitai Regev about Three-Rowed Young Tableaux (and much more!)* / 2006 */ Shalosh B. EKHAD and Doron ZEILBERGER*

http://www.math.rutgers.edu/~zeilberg/mamarim/mamarimPDF/regev.pdf

十五、*R. P. Stanley / Enumerative Combinatorics, Cambridge Vol. 2 / 1999 / p. 452.*

十六、四人一路領先 *http://www.research.att.com/~njas/sequences/A005817*

十七、五人一路領先*http://www.research.att.com/~njas/sequences/A049401*

捌、附錄

一、

第六頁 2. 證明：$W_{m,n,l}=(-1)^{\frac{(m+n+l)!}{m!n!l!}}$ ，其中$m,n,l\in N\cup\{0\}$

(一). 若m,n,l中有一為0，則與二維平面結果一致，即

$$W_{m,n,0}=W_{m-1,n,0}\times W_{m,n-1,0}=(-1)^{C_n^{m+n}}=(-1)^{\frac{(m+n)!}{m!n!}}=(-1)^{\binom{m+n}{m,\ n,\ 0}}$$

同理$W_{m,0,l}=(-1)^{\binom{m+l}{m,\ 0,\ l}}$ ，$W_{0,n,l}=(-1)^{\binom{n+l}{0,\ n,\ l}}$

(二). 1. 考慮$n=l=1$時

$\because W_{m,n,l}=W_{m-1,n,l}\times W_{m,n-1,l}\times W_{m,n,l-1}$ ，其中$m\in N\cup\{0\}$

$$\therefore \begin{cases} W_{m,1,1}=W_{m-1,1,1}\times W_{m,0,1}\times W_{m,1,0} \\ W_{m-1,1,1}=W_{m-2,1,1}\times W_{m-1,0,1}\times W_{m-1,1,0} \\ \quad\vdots \\ W_{1,1,1}=W_{0,1,1}\times W_{1,0,1}\times W_{1,1,0} \end{cases}$$

$$\therefore W_{m,1,1}=W_{0,1,1}\times\prod_{i=1}^{m}W_{i,0,1}\times\prod_{i=1}^{m}W_{i,1,0}=(-1)^{2+2(C_1^2+C_2^3+\ldots+C_m^{m+1})}$$
$$=(-1)^{2+2(C_m^{m+2}-1)}=(-1)^{2\times C_2^{m+2}}=(-1)^{\binom{m+2}{m,\ 1,\ 1}}$$

2. 考慮$n=2$ ，$l=1$時

$\because W_{m,n,l}=W_{m-1,n,l}\times W_{m,n-1,l}\times W_{m,n,l-1}$ ，其中$m\in N\cup\{0\}$

$$\therefore \begin{cases} W_{m,2,1}=W_{m-1,2,1}\times W_{m,1,1}\times W_{m,2,0} \\ W_{m-1,2,1}=W_{m-2,2,1}\times W_{m-1,1,1}\times W_{m-1,2,0} \\ \quad\vdots \\ W_{1,2,1}=W_{0,2,1}\times W_{1,1,1}\times W_{1,2,0} \end{cases}$$

$$\therefore W_{m,2,1}=W_{0,2,1}\times\prod_{i=1}^{m}W_{i,1,1}\times\prod_{i=1}^{m}W_{i,2,0}$$
$$=(-1)^{C_1^3}\times(-1)^{\binom{3}{1,\ 1,\ 1}+\binom{4}{2,\ 1,\ 1}+\cdots+\binom{m+2}{m,\ 1,\ 1}}\times(-1)^{C_1^3+C_2^4+\cdots+C_m^{m+2}}$$
$$=(-1)^{3+(C_m^{m+3}-1)+2!\cdot(C_m^{m+3}-1)}=(-1)^{3+3(C_m^{m+3}-1)}=(-1)^{\binom{m+3}{m,\ 2,\ 1}}$$

3. pf ：$W_{m,n,1}=(-1)^{\binom{m+n+1}{m,\ n,\ 1}}$ ，其中$m,n\in N\cup\{0\}$

給定m ，

當$n=1$時，由 1. 可得$W_{m,1,1}=(-1)^{\binom{m+2}{m,\ 1,\ 1}}$ $\therefore n=1$時成立

假設 $n=n_0\geq 1$ 時成立，即 $W_{m,n_0,1}=(-1)^{\binom{m+n_0+1}{m,\ n_0,\ 1}}$

則 $n=n_0+1$ 時，

$\because W_{m,n,l}=W_{m-1,n,l}\times W_{m,n-1,l}\times W_{m,n,l-1}$

$$\therefore \begin{cases} W_{m,n_0+1,1}=W_{m-1,n_0+1,1}\times W_{m,n_0,1}\times W_{m,n_0+1,0} \\ W_{m-1,n_0+1,1}=W_{m-2,n_0+1,1}\times W_{m-1,n_0,1}\times W_{m-1,n_0+1,0} \\ \qquad\vdots \\ W_{1,n_0+1,1}=W_{0,n_0+1,1}\times W_{1,n_0,1}\times W_{1,n_0+1,0} \end{cases}$$

$$\therefore W_{m,n_0+1,1}=W_{0,n_0+1,1}\times\prod_{i=1}^{m}W_{i,n_0,1}\times\prod_{i=1}^{m}W_{i,n_0+1,0}$$

$$=(-1)^{C_{n_0+1}^{n_0+2}}\times(-1)^{\binom{n_0+2}{1,\ n_0,\ 1}+\binom{n_0+3}{2,\ n_0,\ 1}+\cdots+\binom{m+n_0+1}{m,\ n_0,\ 1}}$$

$$\times(-1)^{C_1^{n_0+2}+C_2^{n_0+3}+\cdots+C_m^{m+n_0+1}}=(-1)^{\binom{m+(n_0+1)+1}{m,\ n_0+1,\ 1}}$$

$\therefore$由數學歸納法得知本命題得證

同理可證：$W_{m,1,l}=(-1)^{\binom{m+1+l}{m,\ 1,\ l}}$，$W_{1,n,l}=(-1)^{\binom{1+n+l}{1,\ n,\ l}}$

4. (1). 預備定理：$\forall m,n\in N$

$$\binom{m+n+l-1}{m-1,\ n,\ l}+\binom{m+n+l-1}{m,\ n-1,\ l}+\binom{m+n+l-1}{m,\ n,\ l-1}=\binom{m+n+l}{m,\ n,\ l}$$

pf：左式 $=(m+n+l-1)!\times\left[\dfrac{1}{(m-1)!n!l!}+\dfrac{1}{m!(n-1)!l!}+\dfrac{1}{m!n!(l-1)!}\right]$

$=(m+n+l-1)!\times\dfrac{m+n+l}{m!n!l!}=$ 右式

$\therefore$本預備定理成立。

(2). pf：給定 m，$W_{m,n,2}=(-1)^{\frac{(m+n+2)!}{m!n!2!}}$，其中 $m,n\in N$

當 $n=1$ 時，由 3. 可得 $W_{m,1,2}=(-1)^{\binom{m+3}{m,\ 1,\ 2}}$ $\therefore n=1$ 時成立

假設 $n=k$ 時成立，即 $W_{m,k,2}=(-1)^{\frac{(m+k+2)!}{m!k!2!}}$

則 $n=k+1$ 時，$\because W_{m,n,l}=W_{m-1,n,l}\times W_{m,n-1,l}\times W_{m,n,l-1}$

$$\therefore \begin{cases} W_{m,k+1,2}=W_{m-1,k+1,2}\times W_{m,k,2}\times W_{m,k+1,1} \\ W_{m-1,k+1,2}=W_{m-2,k+1,2}\times W_{m-1,k,2}\times W_{m-1,k+1,1} \\ \qquad\vdots \\ W_{1,k+1,2}=W_{0,k+1,2}\times W_{1,k,2}\times W_{1,k+1,1} \end{cases}$$

$$\therefore W_{m,k+1,2}=W_{0,k+1,2}\times\prod_{i=1}^{m}W_{i,k,2}\times\prod_{i=1}^{m}W_{i,k+1,1}$$

$$=(-1)^{\frac{(k+3)!}{(k+1)!2!}}\times(-1)^{\begin{pmatrix} & k+3 & \\ 1, & k, & 2\end{pmatrix}+\begin{pmatrix} & k+4 & \\ 2, & k, & 2\end{pmatrix}+\cdots+\begin{pmatrix} & m+k+2 & \\ m, & k, & 2\end{pmatrix}}$$

$$\times(-1)^{\begin{pmatrix} & k+3 & \\ 1, & k+1, & 1\end{pmatrix}+\begin{pmatrix} & k+4 & \\ 2, & k+1, & 1\end{pmatrix}+\cdots+\begin{pmatrix} & m+k+2 & \\ m, & k+1, & 1\end{pmatrix}}$$

$$=(-1)^{\begin{pmatrix} & k+3 & \\ 0, & k+1, & 2\end{pmatrix}+\begin{pmatrix} & k+3 & \\ 1, & k, & 2,\end{pmatrix}+\begin{pmatrix} & k+3 & \\ 1, & k+1, & 1\end{pmatrix}}$$

$$\times(-1)^{\begin{pmatrix} & k+4 & \\ 2, & k, & 2\end{pmatrix}+\begin{pmatrix} & k+4 & \\ 2, & k+1, & 1\end{pmatrix}+\cdots+\begin{pmatrix} & m+k+2 & \\ m, & k+1, & 1\end{pmatrix}}$$

$$=(-1)^{\begin{pmatrix} & m+(k+1)+2 & \\ m, & k+1, & 2\end{pmatrix}}$$

$\therefore$由數學歸納法得知本題得證

同理可證：$W_{2,n,l}=(-1)^{\begin{pmatrix} & n+l+2 & \\ 2, & n, & l\end{pmatrix}}$，$W_{m,2,l}=(-1)^{\begin{pmatrix} & m+l+2 & \\ m, & 2, & l\end{pmatrix}}$

5. pf ：$W_{m,n,l}=(-1)^{\begin{pmatrix} & m+n+l & \\ m, & n, & l\end{pmatrix}}$，其中$m,n,l\in N$

當$l=1$時，於 3. 已證畢， $\therefore l=1$時成立

假設$l=l_0\geq 1$時成立，即$W_{m,n,l_0}=(-1)^{\begin{pmatrix} & m+n+l & \\ m, & n, & l_0\end{pmatrix}}$

則$l=l_0+1$時：再針對n討論

當$n=1$時，$\because W_{m,1,l_0+1}=(-1)^{\begin{pmatrix} & m+l_0+2 & \\ m, & 1, & l_0+1\end{pmatrix}}$ ，

$\therefore n=1$成立

假設n=n_0時成立，即$W_{m,n_0,l_0+1}=(-1)^{\begin{pmatrix} & m+n_0+l_0+1 & \\ m, & n_0, & l_0+1\end{pmatrix}}$

則n=n_0+1時，

$$\because\begin{cases} W_{m,n_0+1,l_0+1}=W_{m-1,n_0+1,l_0+1}\times W_{m,n_0,l_0+1}\times W_{m,n_0+1,l_0} \\ W_{m-1,n_0+1,l_0+1}=W_{m-2,n_0+1,l_0+1}\times W_{m-1,n_0,l_0+1}\times W_{m-1,n_0+1,l_0} \\ \qquad\vdots \\ W_{1,n_0+1,l_0+1}=W_{0,n_0+1,l_0+1}\times W_{1,n_0,l_0+1}\times W_{1,n_0+1,l_0}\end{cases}$$

$$\therefore W_{m,n_0+1,l_0+1}=W_{0,n_0+1,l_0+1}\times\prod_{i=1}^{m}W_{i,n_0,l_0+1}\times\prod_{i=1}^{m}W_{i,n_0+1,l_0}$$

$$=(-1)^{\frac{(n_0+l_0+2)!}{(n_0+1)!(l_0+1)!}}$$

$$\times(-1)^{\begin{pmatrix} & n_0+l_0+2 & \\ 1, & n_0, & l_0+1\end{pmatrix}+\begin{pmatrix} & n_0+l_0+3 & \\ 2, & n_0, & l_0+1\end{pmatrix}+\cdots+\begin{pmatrix} & m+n_0+l_0+1 & \\ m, & n_0, & l_0+1\end{pmatrix}}$$

$$\times(-1)^{\begin{pmatrix} & n_0+l_0+2 & \\ 1, & n_0+1, & l_0\end{pmatrix}+\begin{pmatrix} & n_0+l_0+3 & \\ 2, & n_0+1, & l_0\end{pmatrix}+\cdots+\begin{pmatrix} & m+n_0+l_0+1 & \\ m, & n_0+1, & l_0\end{pmatrix}}$$

$$= (-1)^{\binom{m+n_0+l_0+2}{m,\ n_0+1,\ l_0+1}} \quad \therefore n = n_0 + 1 \text{時成立}$$

$$\therefore \text{由數學歸納法得知} W_{m,n,l_0+1} = (-1)^{\binom{m+n+l_0+1}{m,\ n,\ l_0+1}} \text{成立}$$

$$\therefore \text{由數學歸納法得知} W_{m,n,l} = (-1)^{\binom{m+n+l}{m,\ n,\ l}} \text{成立}$$

二、第八頁 2.

(一). 證明：$T_{m,n} = (-1)^{C_{m-1}^{m+n-1} - C_{n-2}^{m+n-1}}$，其中 $m \ge n \ge 1$，分五步驟證明如下：

1. 證明：$T_{m,1} = (-1)^m$，$\forall m \in N$

   pf：$\because T_{m,n} = T_{m,n-1} \times T_{m-1,n}$，其中 $m > n \ge 0$

$$\therefore \begin{cases} T_{1,1} = T_{1,0} \\ T_{2,1} = T_{1,1} \times T_{2,0} \\ T_{3,1} = T_{2,1} \times T_{3,0} \\ \quad \vdots \\ T_{m,1} = T_{m-1,1} \times T_{m,0} \end{cases}$$

$$\therefore T_{m,1} = \prod_{i=1}^{m} T_{i,0} = \prod_{i=1}^{m} W_{i,0} = (-1)^{C_0^1 + C_0^2 + \cdots + C_0^m} = (-1)^m$$

2. 證明：$T_{m,2} = (-1)^{C_2^{m+1} - 1}$，$\forall m \in N$ 且 $m \ge 2$

   pf：仿 (a). $\therefore T_{m,2} = \prod_{i=2}^{m} T_{i,1} = (-1)^{2+3+\cdots+m} = (-1)^{C_2^{m+1} - 1}$

3. 證明：$T_{m,3} = (-1)^{C_3^{m+2} - C_1^{m+2}}$，$\forall m \in N$ 且 $m \ge 3$

   pf：由 (b).

$$\therefore T_{m,3} = \prod_{i=3}^{m} T_{i,2} = (-1)^{(C_2^4 - 1) + (C_3^5 - 1) + \cdots + (C_{m-1}^{m+1} - 1)}$$

$$= (-1)^{(C_2^4 + C_3^5 + \cdots + C_{m-1}^{m+1}) - (m-2)} = (-1)^{C_{m-1}^{m+2} - C_1^4 - (m-2)} = (-1)^{C_3^{m+2} - C_1^{m+2}}$$

4. 由以上的規律，我們推測：$T_{m,n} = (-1)^{C_{m-1}^{m+n-1} - C_{n-2}^{m+n-1}}$，$\forall m \ge n \ge 1$

   pf：當 $n = 1$ 時，(a). 已證畢， $\therefore n = 1$ 時成立

   假設 $n = k$ 時成立，即 $T_{m,k} = (-1)^{C_{m-1}^{m+k-1} - C_{k-2}^{m+k-1}}$

   則 $n = k+1$ 時

$$T_{m,k+1} = \prod_{i=k+1}^{m} T_{i,k} = T_{k+1,k} \times T_{k+2,k} \times \cdots \times T_{m,k}$$

$$= (-1)^{(C_k^{2k} - C_{k-2}^{2k}) + (C_{k+1}^{2k+1} - C_{k-2}^{2k+1}) + \cdots + (C_{m-1}^{m+k-1} - C_{k-2}^{m+k-1})}$$

$$= (-1)^{(C_k^{2k} + C_{k+1}^{2k+1} + \cdots + C_{m-1}^{m+k-1}) - (C_{k+2}^{2k} + C_{k+3}^{2k+1} + \cdots + C_{m+1}^{m+k-1})}$$

$$= (-1)^{(C_{m-1}^{m+k} - C_{k+1}^{2k}) - (C_{k+2}^{2k} + C_{k+3}^{2k+1} + \cdots + C_{m+1}^{m+k-1})} = (-1)^{C_{m-1}^{m+k} - (C_{k+1}^{2k} + C_{k+2}^{2k} + \cdots + C_{m+1}^{m+k-1})}$$

$$=(-1)^{C_{m-1}^{m+k}-C_{m+1}^{m+k}}=(-1)^{C_{m-1}^{m+(k+1)-1}-C_{(k+1)-2}^{m+(k+1)-1}}$$

$\therefore$由數學歸納法得知本命題得證

又 $C_{m-1}^{m+n-1}-C_{n-2}^{m+n-1}=C_{m-1}^{m+n-1}+(C_{m}^{m+n-1}-C_{n-1}^{m+n-1})-C_{n-2}^{m+n-1}$

$\qquad\qquad=C_{m}^{m+n}-C_{n-1}^{m+n}=C_{n}^{m+n}-C_{n-1}^{m+n}$

$\therefore T_{m,n}=(-1)^{C_{n}^{m+n}-C_{n-1}^{m+n}}$ ， $m\ge n\ge 1$

在運算過程中，會遇到型如$T_{a,-1}$、$T_{a,-2}$、$T_{-1,b}$之情形，可視爲此坐標格子(a,b)之左方或下方或再下方無格子，所以定義$T_{a,-1}=T_{a,-2}=T_{-1,b}=1$，甚至可定義其對應之冪次值爲$0$ ，$\therefore T_{m,n}=(-1)^{C_{n}^{m+n}-C_{n-1}^{m+n}}$ ，$\forall m\ge n\ge 0$ 。

三、第九頁

(一). 證明：$T_{m,n,l}=(-1)^{\binom{m+n+l}{m,\ n,\ l}-\binom{m+n+l}{m,\ n+1,\ l-1}+\binom{m+n+l}{m+1,\ n+1,\ l-2}-\binom{m+n+l}{m+1,\ n-1,\ l}}$

$\qquad\times(-1)^{\binom{m+n+l}{m+2,\ n-1,\ l-1}-\binom{m+n+l}{m+2,\ n,\ l-2}}$，其中$m,n,l\in N\bigcup\{0\}$且$m\ge n\ge l$ 。

1. pf：$T_{m,1,1}=(-1)^{C_{2}^{m+1}}$ ，其中$m\ge 1$

$$\because\begin{cases}T_{1,1,1}=T_{1,1,0}\\ T_{2,1,1}=T_{1,1,1}\times T_{2,1,0}\\ \quad\vdots\\ T_{m,1,1}=T_{m-1,1,1}\times T_{m,1,0}\end{cases}$$

$$\therefore T_{m,1,1}=\prod_{i=1}^{m}T_{i,1,0}=\prod_{i=1}^{m}T_{i,1}=(-1)^{1+2+\ldots+m}=(-1)^{C_{2}^{m+1}}$$

2. pf：$T_{m,2,1}=(-1)^{2C_{m-1}^{m+2}-C_{m+1}^{m+2}+C_{m+2}^{m+2}}$ ，其中$m\ge 1$

仿 1. $\therefore T_{m,2,1}=(-1)^{2C_{m-1}^{m+2}-C_{m+1}^{m+2}+C_{m+2}^{m+2}}$

3. pf：$T_{m,3,1}=(-1)^{3C_{m-1}^{m+3}-2C_{m+1}^{m+3}+C_{m+2}^{m+3}}$ ，其中$m\ge 1$

仿 1. $\therefore T_{m,3,1}=(-1)^{3C_{m-1}^{m+3}-2C_{m+1}^{m+3}+C_{m+2}^{m+3}}$

4. pf： $T_{m,n,l}=(-1)^{\binom{m+n+l}{m,\ n,\ l}-\binom{m+n+l}{m,\ n+1,\ l-1}+\binom{m+n+l}{m+1,\ n+1,\ l-2}-\binom{m+n+l}{m+1,\ n-1,\ l}}$

$\qquad\times(-1)^{\binom{m+n+l}{m+2,\ n-1,\ l-1}-\binom{m+n+l}{m+2,\ n,\ l-2}}$，其中$m\ge n\ge l\ge 0$

我們觀察： $T_{m,2,1}=(-1)^{2\times C_{m-1}^{m+2}-C_{m+1}^{m+2}+C_{m+2}^{m+2}}$ ， $T_{m,3,1}=(-1)^{3C_{m-1}^{m+3}-2C_{m+1}^{m+3}+C_{m+2}^{m+3}}$

所以推測$T_{m,n,1}=(-1)^{n\times C_{m-1}^{m+n}-(n-1)C_{m+1}^{m+n}+C_{m+2}^{m+n}}$ ，其中$m\ge n\ge 1$

(1). pf：$T_{m,n,1}=(-1)^{n\cdot C_{m-1}^{m+n}-(n-1)\cdot C_{m+1}^{m+n}+C_{m+2}^{m+n}}$ ， $m\ge n\ge 1$

當$n=1$時，1. 已證畢，$\therefore n=1$時成立

假設$n=k$時成立，即$T_{m,k,1}=(-1)^{k\cdot C_{m-1}^{m+k}-(k-1)\cdot C_{m+1}^{m+k}+C_{m+2}^{m+k}}$ 成立

則$n=k+1$時，

$$\because \begin{cases} T_{k+1,k+1,1}=T_{k+1,k,1}\times T_{k+1,k+1,0} \\ T_{k+2,k+1,1}=T_{k+1,k+1,1}\times T_{k+2,k,1}\times T_{k+2,k+1,0} \\ \qquad\vdots \\ T_{m,k+1,1}=T_{m-1,k+1,1}\times T_{m,k,1}\times T_{m,k+1,0} \end{cases}$$

$$\begin{aligned}
\therefore T_{m,k+1,1} &= \prod_{i=k+1}^{m} T_{i,k,1}\times\prod_{i=k+1}^{m} T_{i,k+1,0} \\
&= (-1)^{[k\cdot C_{k}^{2k+1}-(k-1)C_{k+2}^{2k+1}+C_{k+3}^{2k+1}]+[k\cdot C_{k+1}^{2k+2}-(k-1)C_{k+3}^{2k+2}+C_{k+4}^{2k+2}]+\ldots+[k\cdot C_{m-1}^{m+k}-(k-1)C_{m+1}^{m+k}+C_{m+2}^{m+k}]} \\
&\quad \times(-1)^{(C_{k}^{2k+1}-C_{k+2}^{2k+1})+(C_{k+1}^{2k+2}-C_{k+3}^{2k+2})+\ldots+(C_{m-1}^{m+k}-C_{m+1}^{m+k})} \\
&= (-1)^{k(C_{k}^{2k+1}+C_{k+1}^{2k+2}+\ldots+C_{n-1}^{n+k})-(k-1)(C_{k+2}^{2k+1}+C_{k+3}^{2k+2}+\ldots+C_{n+1}^{n+k})+(C_{k+3}^{2k+1}+C_{k+4}^{2k+2}+\ldots+C_{n+2}^{n+k})} \\
&\quad \times(-1)^{(C_{k}^{2k+1}+C_{k+1}^{2k+2}+\ldots+C_{n-1}^{n+k})-(C_{k+2}^{2k+1}+C_{k+3}^{2k+2}+\ldots+C_{n+1}^{n+k})} \\
&= (-1)^{(k+1)(C_{k}^{2k+1}+C_{k+1}^{2k+2}+\ldots+C_{n-1}^{n+k})-k(C_{k+2}^{2k+1}+C_{k+3}^{2k+2}+\ldots+C_{n+1}^{n+k})+(C_{k+3}^{2k+1}+C_{k+4}^{2k+2}+\ldots+C_{n+2}^{n+k})} \\
&= (-1)^{(k+1)(C_{n-1}^{n+k+1}-C_{k-1}^{2k+1})-k(C_{n+1}^{n+k+1}-C_{k+1}^{2k+1})+(C_{n+2}^{n+k+1}-C_{k+2}^{2k+1})} \\
&= (-1)^{[(k+1)C_{n-1}^{n+k+1}-kC_{n+1}^{n+k+1}+C_{n+2}^{n+k+1}]+[-(k+1)C_{k-1}^{2k+1}+k\cdot C_{k+1}^{2k+1}-C_{k+2}^{2k+1}]}
\end{aligned}$$

其中：$-(k+1)C_{k-1}^{2k+1}+k\cdot C_{k+1}^{2k+1}-C_{k+2}^{2k+1}=k(C_{k+1}^{2k+1}-C_{k-1}^{2k+1})-(C_{k-1}^{2k+1}+C_{k+2}^{2k+1})$

$$=k\left(\frac{(2k+1)!}{k!(k+1)!}-\frac{(2k+1)!}{(k+2)!(k-1)!}\right)-2\times C_{k+2}^{2k+1}=k\times\frac{(2k+1)!(k+2-k)}{k!(k+2)!}-2\times C_{k+2}^{2k+1}$$

$$=2\times\frac{(2k+1)!}{(k-1)!(k+2)!}-2\times C_{k+2}^{2k+1}=2\times C_{k+2}^{2k+1}-2\times C_{k+2}^{2k+1}=0$$

$\therefore T_{n,k+1,1}=(-1)^{[(k+1)C_{n-1}^{n+(k+1)}-kC_{n+1}^{n+(k+1)}+C_{n+2}^{n+(k+1)}]}$

$\therefore n=k+1$時成立

∴由數學歸納法得知本命題得證

(2). $pf: T_{m,2,2}=(-1)^{2C_{m-1}^{m+3}-C_{m+1}^{m+3}+C_{m+2}^{m+3}}$ ，$m\geq 2$

仿(1). $\therefore T_{m,2,2}=(-1)^{2C_{m-1}^{m+3}-C_{m+1}^{m+3}+C_{m+2}^{m+3}}$

(3). $pf: T_{m,3,2}=(-1)^{5C_{m-1}^{m+4}-3C_{m+1}^{m+4}+2C_{m+2}^{m+4}}$ ，$m\geq 3$

仿(1). $\therefore T_{m,3,2}=(-1)^{5C_{m-1}^{m+4}-3C_{m+1}^{m+4}+2C_{m+2}^{m+4}}$

(4). $pf: T_{m,n,2}=(-1)^{(C_{2}^{n+1}-1)C_{m-1}^{m+n+1}-C_{2}^{n}C_{m+1}^{m+n+1}+(n-1)C_{m+2}^{m+n+1}}$

由觀察，推測對任意$n\in N$，

$T_{m,n,2}=(-1)^{(C_2^{n+1}-1)C_{m-1}^{m+n+1}-C_2^{n}C_{m+1}^{m+n+1}+(n-1)C_{m+2}^{m+n+1}}$，其中 $m\geq n\geq 2$

當 $n=2$ 時，於(2). 已證畢，$\therefore n=2$ 時成立

假設 $n=k$ 時成立，

即 $T_{m,k,2}=(-1)^{(C_2^{k+1}-1)C_{m-1}^{m+k+1}-C_2^{k}C_{m+1}^{m+k+1}+(k-1)C_{m+2}^{m+k+1}}$，其中 $m\geq k\geq 2$

則 $n=k+1$ 時

$$\because\begin{cases} T_{k+1,k+1,2}=T_{k+1,k,2}\times T_{k+1,k+1,1} \\ T_{k+2,k+1,2}=T_{k+1,k+1,2}\times T_{k+2,k,2}\times T_{k+2,k+1,1} \\ \vdots \\ T_{m,k+1,2}=T_{m-1,k+1,2}\times T_{m,k,2}\times T_{m,k+1,1} \end{cases}$$

$$\therefore T_{m,k+1,2}=\prod_{i=k+1}^{m}T_{i,k,2}\times\prod_{i=k+1}^{m}T_{i,k+1,1}$$

$$=(-1)^{\left[(C_2^{k+1}-1)(C_k^{2k+2}+C_{k+1}^{2k+3}+...+C_{m-1}^{m+k+1})-C_2^{k}(C_{k+2}^{2k+2}+C_{k+3}^{2k+3}+...+C_{m+1}^{m+k+1})+(k-1)(C_{k+3}^{2k+2}+C_{k+4}^{2k+3}+...+C_{m+2}^{m+k+1})\right]}$$

$$\times(-1)^{\left[(k+1)(C_k^{2k+2}+C_{k+1}^{2k+3}+...+C_{m-1}^{m+k+1})-k(C_{k+2}^{2k+2}+C_{k+3}^{2k+3}+...+C_{m+1}^{m+k+1})+(C_{k+3}^{2k+2}+C_{k+4}^{2k+3}+...+C_{m+2}^{m+k+1})\right]}$$

$$=(-1)^{\left[(C_2^{k+1}-1+k+1)(C_{m-1}^{m+k+2}-C_{k-1}^{2k+2})-(C_2^{k}+k)(C_{m+1}^{m+k+2}-C_{k+1}^{2k+2})+(k-1+1)(C_{m+2}^{m+k+2}-C_{k+2}^{2k+2})\right]}$$

$$=(-1)^{\left[(C_2^{k+1}+C_1^{k+1}-1)(C_{m-1}^{m+k+2}-C_{k-1}^{2k+2})-(C_2^{k}+C_1^{k})(C_{m+1}^{m+k+2}-C_{k+1}^{2k+2})+k(C_{m+2}^{m+k+2}-C_{k+2}^{2k+2})\right]}$$

$$=(-1)^{\left[(C_2^{k+2}-1)C_{m-1}^{m+k+2}-C_2^{k+1}\cdot C_{m+1}^{m+k+2}+k\cdot C_{m+2}^{m+k+2}\right]-\left[(C_2^{k+2}-1)C_{k-1}^{2k+2}-C_2^{k+1}C_{k+1}^{2k+2}+k\cdot C_{k+2}^{2k+2}\right]}$$

其中：$(C_2^{k+2}-1)C_{k-1}^{2k+2}-C_2^{k+1}C_{k+1}^{2k+2}+kC_{k+2}^{2k+2}=-C_{k-1}^{2k+2}+C_2^{k+2}C_{k-1}^{2k+2}-C_2^{k+1}C_{k+1}^{2k+2}+kC_{k+2}^{2k+2}$

$$=-C_{k-1}^{2k+2}+\frac{(k+2)!}{2!k!}\times\frac{(2k+2)!}{(k+3)!(k-1)!}-\frac{(k+1)!}{2!(k-1)!}\times\frac{(2k+2)!}{(k+1)!(k+1)!}+\frac{k(2k+2)!}{k!(k+2)!}$$

$$=-C_{k-1}^{2k+2}+\frac{(2k+2)!}{2!k!(k+3)(k-1)!}-\frac{(2k+2)!}{2(k-1)!(k+1)!}+\frac{(2k+2)!}{(k-1)!(k+2)!}$$

$$=-C_{k-1}^{2k+2}+\frac{(2k+2)!(k^2+k-k^2-3k)}{2!k!(k+1)!(k+3)}+\frac{(2k+2)!}{(k-1)!(k+2)!}=-C_{k-1}^{2k+2}+\frac{(2k+2)!(k+3-k-2)}{(k-1)!(k+2)!(k+3)}$$

$$=-C_{k-1}^{2k+2}+C_{k-1}^{2k+2}=0$$

$\therefore n=k+1$ 時成立

$\therefore$由數學歸納法得知本命題得證

5. 由 $T_{m,n,1}$ 及 $T_{m,n,2}$ 型式，我們推測：

$$T_{m,n,l}=(-1)^{(C_l^{n+l-1}-C_{l-1}^{n+l-1})\times C_{m-1}^{m+n+l-1}-(C_l^{n+l-2}-C_{l-3}^{n+l-2})\times C_{m+1}^{m+n+l-1}+(C_{l-1}^{n+l-3}-C_{l-3}^{n+l-3})\times C_{m+2}^{m+n+l-1}}$$

經過化簡後得到：$T_{m,n,l}=(-1)^{\binom{m+n+l}{m,\ n,\ l}-\binom{m+n+l}{m,\ n+1,\ l-1}+\binom{m+n+l}{m+1,\ n+1,\ l-2}-\binom{m+n+l}{m+1,\ n-1,\ l}}$

$$\times(-1)^{\binom{m+n+l}{m+2,\ n-1,\ l-1}-\binom{m+n+l}{m+2,\ n,\ l-2}}$$

(1). pf：(∗)預備定理：

$$\sum_{i=m}^{k}\binom{i+n+l}{i,\ n,\ l}=\frac{(n+l)!}{n!l!}\left(C_m^{m+n+l}+C_{m+1}^{m+1+n+l}+\cdots+C_k^{k+n+l}\right)=\frac{(n+l)!}{n!l!}\left(C_k^{k+n+l+1}-C_{m-1}^{m+n+l}\right)$$

$$=\frac{(n+l)!}{n!l!}\times\left(\frac{(k+n+l+1)!}{k!(n+l+1)!}-\frac{(m+n+l)!}{(m-1)!(n+l+1)!}\right)$$

(2). pf： $T_{m,n,l}=(-1)^{\binom{m+n+l}{m,\ n,\ l}-\binom{m+n+l}{m,\ n+1,\ l-1}+\binom{m+n+l}{m+1,\ n+1,\ l-2}-\binom{m+n+l}{m+1,\ n-1,\ l}}$

$$\times(-1)^{\binom{m+n+l}{m+2,\ n-1,\ l-1}-\binom{m+n+l}{m+2,\ n,\ l-2}}$$

假設$l=l_0$成立，

即$T_{m,n,l_0}=(-1)^{\binom{m+n+l_0}{m,\ n,\ l_0}-\binom{m+n+l_0}{m,\ n+1,\ l_0-1}+\binom{m+n+l_0}{m+1,\ n+1,\ l_0-2}-\binom{m+n+l_0}{m+1,\ n-1,\ l_0}}$

$$\times(-1)^{\binom{m+n+l_0}{m+2,\ n-1,\ l_0-1}-\binom{m+n+l_0}{m+2,\ n,\ l_0-2}}$$

則$l=l_0+1$時

當 $n=l_0+1$

$$\because\begin{cases}T_{l_0+1,l_0+1,l_0+1}=T_{l_0+1,l_0+1,l_0}\\T_{l_0+2,l_0+1,l_0+1}=T_{l_0+1,l_0+1,l_0+1}\times T_{l_0+2,l_0+1,l_0}\\T_{l_0+3,l_0+1,l_0+1}=T_{l_0+2,l_0+1,l_0+1}\times T_{l_0+3,l_0+1,l_0}\\\qquad\vdots\\T_{m,l_0+1,l_0+1}=T_{m-1,l_0+1,l_0+1}\times T_{m,l_0+1,l_0}\end{cases}$$

$$\therefore T_{m,l_0+1,l_0+1}=\prod_{i=l_0+1}^{m}T_{i,l_0+1,l_0}=(-1)^{\sum_{i=l_0+1}^{m}\binom{i+l_0+1+l_0}{i,\ l_0+1,\ l_0}+\sum_{i=l_0+1}^{m}\binom{i+l_0+1+l_0}{i+1,\ l_0+2,\ l_0-2}+\sum_{i=l_0+1}^{m}\binom{i+l_0+1+l_0}{i+2,\ l_0,\ l_0-1}}$$

$$\times(-1)^{-\sum_{i=l_0+1}^{m}\binom{i+l_0+1+l_0}{i,\ l_0+2,\ l_0-1}-\sum_{i=l_0+1}^{m}\binom{i+l_0+1+l_0}{i+1,\ l_0,\ l_0}-\sum_{i=l_0+1}^{m}\binom{i+l_0+1+l_0}{i+2,\ l_0+1,\ l_0-2}}$$

$$=(-1)^{\frac{(2l_0+1)!}{(l_0+1)!l_0!}\left(\frac{(m+2l_0+2)!}{m!(2l_0+2)!}-\frac{(3l_0+2)!}{l_0!(2l_0+2)!}\right)+\frac{(2l_0)!}{(l_0+2)!(l_0-2)!}\left(\frac{(m+2l_0+2)!}{(m+1)!(2l_0+1)!}-\frac{(3l_0+2)!}{(l_0+1)!(2l_0+1)!}\right)+\frac{(2l_0-1)!}{l_0!(l_0-1)!}\left(\frac{(m+2l_0+2)!}{(m+2)!(2l_0)!}-\frac{(3l_0+2)!}{(l_0+2)!(2l_0)!}\right)}$$

$$\times(-1)^{-\frac{(2l_0+1)!}{(l_0+2)!(l_0-1)!}\left(\frac{(m+2l_0+2)!}{m!(2l_0+2)!}-\frac{(3l_0+2)!}{l_0!(2l_0+2)!}\right)-\frac{(2l_0)!}{l_0!l_0!}\left(\frac{(m+2l_0+2)!}{(m+1)!(2l_0+1)!}-\frac{(3l_0+2)!}{(l_0+1)!(2l_0+1)!}\right)-\frac{(2l_0-1)!}{(l_0+1)!(l_0-2)!}\left(\frac{(m+2l_0+2)!}{(m+2)!(2l_0)!}-\frac{(3l_0+2)!}{(l_0+2)!(2l_0)!}\right)}$$

(∵預備定理)，又

$$\frac{(3l_0+2)!}{(l_0+1)!l_0!l_0!(2l_0+2)}+\frac{(3l_0+2)!}{(l_0+2)!(l_0-2)!(l_0+1)!(2l_0+1)}+\frac{(3l_0+2)!}{l_0!(l_0-1)!(l_0+2)!2l_0}-\frac{(3l_0+2)!}{l_0!(l_0+2)!(l_0-1)!(2l_0+2)}-\frac{(3l_0+2)!}{l_0!l_0!(l_0+1)!(2l_0+1)}-\frac{(3l_0+2)!}{(l_0+1)!(l_0-2)!(l_0+2)!2l_0}$$

$$=\frac{(3l_0+2)!\left[\left(l_0+2\right)-l_0\right]}{(l_0+1)!l_0!l_0!(2l_0+2)}+\frac{(3l_0+2)!\left[l_0\left(l_0-1\right)-\left(l_0+1\right)\left(l_0+2\right)\right]}{(l_0+2)!(l_0+1)!l_0!(2l_0+1)}+\frac{(3l_0+2)!\left[\left(l_0+1\right)-\left(l_0-1\right)\right]}{(l_0+2)!(l_0+1)!(l_0-1)!2l_0}$$

$$=\frac{2\times(3l_0+2)!}{(l_0+1)!l_0!l_0!(2l_0+2)}+\frac{2\times(3l_0+2)!}{(l_0+2)!(l_0+1)!(l_0-1)!2l_0}-\frac{\left[4l_0+2\right](3l_0+2)!}{(l_0+2)!(l_0+1)!l_0!(2l_0+1)}=0$$

$$\therefore T_{m,l_0+1,l_0+1}=(-1)^{\frac{(m+2l_0+2)!}{(l_0+1)!l_0!m!(2l_0+2)!}+\frac{(m+2l_0+2)!}{(l_0+2)!(l_0-2)!(m+1)!(2l_0+1)}+\frac{(m+2l_0+2)!}{l_0!(l_0-1)!(m+2)!(2l_0)}}$$

$$\times(-1)^{-\frac{(m+2l_0+2)!}{(l_0+2)!(l_0-1)!m!(2l_0+2)}-\frac{(m+2l_0+2)!}{l_0!l_0!(m+1)!(2l_0+1)}-\frac{(m+2l_0+2)!}{(l_0+1)!(l_0-2)!(m+2)!(2l_0)}}$$

$$=(-1)^{\frac{1}{2}\times\binom{m+2l_0+2}{m\quad l_0+1\quad l_0+1}+\frac{l_0-1}{2l_0+1}\times\binom{m+2l_0+2}{m+1\quad l_0+2\quad l_0-1}+\frac{1}{2}\times\binom{m+2l_0+2}{m+2\quad l_0\quad l_0}-\frac{l_0}{2l_0+2}\times\binom{m+2l_0+2}{m\quad l_0+2\quad l_0}-\frac{l_0+1}{2l_0+1}\times\binom{m+2l_0+2}{m+1\quad l_0\quad l_0+1}-\frac{l_0-1}{2l_0}\times\binom{m+2l_0+2}{m+2\quad l_0+1\quad l_0-1}}$$

$$=(-1)^{\left[\binom{m+l_0+1+l_0+1}{m\quad l_0+1\quad l_0+1}-\binom{m+l_0+1+l_0+1}{m\quad l_0+2\quad l_0}+\binom{m+l_0+1+l_0+1}{m+1\quad l_0+2\quad l_0-1}-\binom{m+l_0+1+l_0+1}{m+1\quad l_0\quad l_0+1}+\binom{m+l_0+1+l_0+1}{m+2\quad l_0\quad l_0}-\binom{m+l_0+1+l_0+1}{m+2\quad l_0+1\quad l_0-1}\right]}$$

$$\times(-1)^{-\left[\frac{1}{2}\times\binom{m+2l_0+2}{m\quad l_0+1\quad l_0+1}+\frac{l_0+2}{2l_0+1}\times\binom{m+2l_0+2}{m+1\quad l_0+2\quad l_0-1}+\frac{1}{2}\times\binom{m+2l_0+2}{m+2\quad l_0\quad l_0}-\frac{l_0+2}{2l_0+2}\times\binom{m+2l_0+2}{m\quad l_0+2\quad l_0}-\frac{l_0}{2l_0+1}\times\binom{m+2l_0+2}{m+1\quad l_0\quad l_0+1}-\frac{l_0+1}{2l_0}\times\binom{m+2l_0+2}{m+2\quad l_0+1\quad l_0-1}\right]}$$

$$=(-1)^{\left[\binom{m+l_0+1+l_0+1}{m\quad l_0+1\quad l_0+1}-\binom{m+l_0+1+l_0+1}{m\quad l_0+2\quad l_0}+\binom{m+l_0+1+l_0+1}{m+1\quad l_0+2\quad l_0-1}-\binom{m+l_0+1+l_0+1}{m+1\quad l_0\quad l_0+1}+\binom{m+l_0+1+l_0+1}{m+2\quad l_0\quad l_0}-\binom{m+l_0+1+l_0+1}{m+2\quad l_0+1\quad l_0-1}\right]}$$

$\therefore n=l_0+1$時成立

假設$n=n_0$時成立，其中$n_0\geq l_0+1$時成立，

即

$$T_{m,n_0,l_0+1}=(-1)^{\left[\binom{m+n_0+l_0+1}{m\quad n_0\quad l_0+1}-\binom{m+n_0+l_0+1}{m\quad n_0+1\quad l_0}+\binom{m+n_0+l_0+1}{m+1\quad n_0+1\quad l_0-1}-\binom{m+n_0+l_0+1}{m+1\quad n_0-1\quad l_0+1}+\binom{m+n_0+l_0+1}{m+2\quad n_0-1\quad l_0}-\binom{m+n_0+l_0+1}{m+2\quad n_0\quad l_0-1}\right]}$$

則$n=n_0+1$時

$$\because\begin{cases}T_{m,n_0+1,l_0+1}=T_{m-1,n_0+1,l_0+1}\times T_{m,n_0,l_0+1}\times T_{m,n_0+1,l_0}\\ T_{m-1,n_0+1,l_0+1}=T_{m-2,n_0+1,l_0+1}\times T_{m-1,n_0,l_0+1}\times T_{m-1,n_0+1,l_0}\\ \qquad\vdots\\ T_{n_0+1,n_0+1,l_0+1}=T_{n_0+1,n_0l_0+1}\times T_{n_0+1,n_0+1,l_0}\end{cases}$$

$$T_{m,n_0+1,l_0+1}=\prod_{i=n_0+1}^{m}T_{i,n_0,l_0+1}\times\prod_{i=n_0+1}^{m}T_{i,n_0+1,l_0}$$

$$=(-1)^{\sum_{i=n_0+1}^{m}\left[\binom{i+n_0+l_0+1}{i,\quad n_0,\quad l_0+1}+\binom{i+n_0+l_0+1}{i+1,\quad n_0+1,\quad l_0-1}+\binom{i+n_0+l_0+1}{i+2,\quad n_0-1,\quad l_0}+\binom{i+n_0+1+l_0}{i,\quad n_0+1,\quad l_0}+\binom{i+n_0+1+l_0}{i+1,\quad n_0+2,\quad l_0-2}+\binom{i+n_0+1+l_0}{i+2,\quad n_0,\quad l_0-1}\right]}$$

$$\times(-1)^{\sum_{i=n_0+1}^{m}\left[-\binom{i+n_0+l_0+1}{i,\quad n_0+1,\quad l_0}-\binom{i+n_0+l_0+1}{i+1,\quad n_0-1,\quad l_0+1}-\binom{i+n_0+l_0+1}{i+2,\quad n_0,\quad l_0-1}-\binom{i+n_0+1+l_0}{i,\quad n_0+2,\quad l_0-1}-\binom{i+n_0+1+l_0}{i+1,\quad n_0,\quad l_0}-\binom{i+n_0+1+l_0}{i+2,\quad n_0+1,\quad l_0-2}\right]}$$

$$=(-1)^{\frac{(n_0+l_0+1)!}{n_0!(l_0+1)!}\left(\frac{(m_0+n_0+l_0+2)!}{m!(n_0+l_0+2)!}-\frac{(2n_0+l_0+2)!}{n_0!(n_0+l_0+2)!}\right)+\frac{(n_0+l_0)!}{(n_0+1)!(l_0-1)!}\left(\frac{(m+n_0+l_0+2)!}{(m+1)!(n_0+l_0+1)!}-\frac{(2n_0+l_0+2)!}{(n_0+1)!(n_0+l_0+1)!}\right)+\frac{(n_0+l_0-1)!}{(n_0-1)!l_0!}\left(\frac{(m+n_0+l_0+2)!}{(m+2)!(n_0+l_0)!}-\frac{(2n_0+l_0+2)!}{(n_0+2)!(n_0+l_0)!}\right)}$$

$$\times(-1)^{\frac{(n_0+l_0)!}{(n_0+2)!(l_0-2)!}\left(\frac{(m+n_0+l_0+2)!}{(m+1)!(n_0+l_0+1)!}-\frac{(2n_0+l_0+2)!}{(n_0+1)!(n_0+l_0+1)!}\right)-\frac{(n_0+l_0)!}{(n_0-1)!(l_0+1)!}\left(\frac{(m+n_0+l_0+2)!}{(m+1)!(n_0+l_0+1)!}-\frac{(2n_0+l_0+2)!}{(n_0+1)!(n_0+l_0+1)!}\right)-\frac{(n_0+l_0+1)!}{(n_0+2)!(l_0-1)!}\left(\frac{(m+n_0+l_0+2)!}{m!(n_0+l_0+2)!}-\frac{(2n_0+l_0+2)!}{n_0!(n_0+l_0+2)!}\right)}$$

$$\times(-1)^{-\frac{(n_0+l_0)!}{n_0!l_0!}\left(\frac{(m+n_0+l_0+2)!}{(m+1)!(n_0+l_0+1)!}-\frac{(2n_0+l_0+2)!}{(n_0+1)!(n_0+l_0+1)!}\right)-\frac{(n_0+l_0-1)!}{(n_0+1)!(l_0-2)!}\left(\frac{(m+n_0+l_0+2)!}{(m+2)!(n_0+l_0)!}-\frac{(2n_0+l_0+2)!}{(n_0+2)!(n_0+l_0)!}\right)}$$

又

$$\frac{(n_0+l_0)!}{(n_0+1)!(l_0-1)!}\left(\frac{(m+n_0+l_0+2)!}{(m+1)!(n_0+l_0+1)!}-\frac{(2n_0+l_0+2)!}{(n_0+1)!(n_0+l_0+1)!}\right)+\frac{(n_0+l_0)!}{(n_0+2)!(l_0-2)!}\left(\frac{(m+n_0+l_0+2)!}{(m+1)!(n_0+l_0+1)!}-\frac{(2n_0+l_0+2)!}{(n_0+1)!(n_0+l_0+1)!}\right)$$

$$=\frac{(n_0+l_0+1)!}{(n_0+2)!(l_0-1)!}\left(\frac{(m+n_0+l_0+2)!}{(m+1)!(n_0+l_0+1)!}-\frac{(2n_0+l_0+2)!}{(n_0+1)!(n_0+l_0+1)!}\right)$$

$$-\frac{(n_0+l_0)!}{(n_0-1)!(l_0+1)!}\left(\frac{(m+n_0+l_0+2)!}{(m+1)!(n_0+l_0+1)!}-\frac{(2n_0+l_0+2)!}{(n_0+1)!(n_0+l_0+1)!}\right)-\frac{(n_0+l_0)!}{n_0!l_0!}\left(\frac{(m+n_0+l_0+2)!}{(m+1)!(n_0+l_0+1)!}-\frac{(2n_0+l_0+2)!}{(n_0+1)!(n_0+l_0+1)!}\right)$$

$$=-\frac{(n_0+l_0+1)!}{n_0!(l_0+1)!}\left(\frac{(m+n_0+l_0+2)!}{(m+1)!(n_0+l_0+1)!}-\frac{(2n_0+l_0+2)!}{(n_0+1)!(n_0+l_0+1)!}\right)$$

又 $\frac{(n_0+l_0+1)!}{(n_0+2)!(l_0-1)!}\left(-\frac{(2n_0+l_0+2)!}{(n_0+1)!(n_0+l_0+1)!}\right)+\left(-\frac{(n_0+l_0+1)!}{n_0!(l_0+1)!}\right)\left(-\frac{(2n_0+l_0+2)!}{(n_0+1)!(n_0+l_0+1)!}\right)+\frac{(n_0+l_0+1)!}{n_0!(l_0+1)!}\left(-\frac{(2n_0+l_0+2)!}{n_0!(n_0+l_0+2)!}\right)+\frac{(n_0+l_0-1)!}{(n_0-1)!l_0!}\left(-\frac{(2n_0+l_0+2)!}{(n_0+2)!(n_0+l_0)!}\right)$

$$-\frac{(n_0+l_0+1)!}{(n_0+2)!(l_0-1)!}\left(-\frac{(2n_0+l_0+2)!}{n_0!(n_0+l_0+2)!}\right)-\left(\frac{(n_0+l_0-1)!}{(n_0+1)!(l_0-2)!}\right)\left(-\frac{(2n_0+l_0+2)!}{(n_0+2)!(n_0+l_0)!}\right)=0$$

$$\therefore T_{m,n_0+1,l_0+1}=(-1)^{\begin{pmatrix} & m+n_0+l_0+2 & \\ m+1, & n_0+2, & l_0-1\end{pmatrix}-\begin{pmatrix} & m+n_0+l_0+2 & \\ m+1, & n_0, & l_0+1\end{pmatrix}+\frac{n_0+1}{n_0+l_0+2}\begin{pmatrix} & m+n_0+l_0+2 & \\ m, & n_0+1, & l_0+1\end{pmatrix}+\frac{n_0}{n_0+l_0}\begin{pmatrix} & m+n_0+l_0+2 & \\ m+2, & n_0, & l_0\end{pmatrix}}$$

$$\times(-1)^{-\frac{l_0}{n_0+l_0+2}\begin{pmatrix} & m+n_0+l_0+2 & \\ m, & n_0+2, & l_0\end{pmatrix}-\frac{l_0-1}{n_0+l_0}\begin{pmatrix} & m+n_0+l_0+2 & \\ m+2, & n_0+1, & l_0-1\end{pmatrix}}$$

$$=(-1)^{\begin{pmatrix} & m+n_0+1+l_0+1 & \\ m, & n_0+1, & l_0+1\end{pmatrix}-\begin{pmatrix} & m+n_0+1+l_0+1 & \\ m, & n_0+2, & l_0\end{pmatrix}+\begin{pmatrix} & m+n_0+1+l_0+1 & \\ m+1, & n_0+2, & l_0-1\end{pmatrix}-\begin{pmatrix} & m+n_0+1+l_0+1 & \\ m+1, & n_0, & l_0+1\end{pmatrix}+\begin{pmatrix} & m+n_0+1+l_0+1 & \\ m+2, & n_0, & l_0\end{pmatrix}}$$

$$\times(-1)^{-\begin{pmatrix} & m+n_0+1+l_0+1 & \\ m+2, & n_0+1, & l_0-1\end{pmatrix}-\left(\frac{l_0+1}{n_0+l_0+2}\times\frac{(m+n_0+l_0+2)!}{m!(n_0+1)!(l_0+1)!}+\frac{l_0}{n_0+l_0}\times\frac{(m+n_0+l_0+2)!}{(m+2)!n_0!l_0!}-\frac{n_0+2}{n_0+l_0+2}\times\frac{(m+n_0+l_0+2)!}{m!(n_0+2)!l_0!}-\frac{n_0+1}{n_0+l_0}\times\frac{(m+n_0+l_0+2)!}{(m+2)!(n_0+1)!(l_0-1)!}\right)}$$

$$=(-1)^{\begin{pmatrix} & m+(n_0+1)+(l_0+1) & \\ m, & (n_0+1), & (l_0+1)\end{pmatrix}-\begin{pmatrix} & m+(n_0+1)+(l_0+1) & \\ m, & (n_0+1)+1, & (l_0+1)-1\end{pmatrix}+\begin{pmatrix} & m+(n_0+1)+(l_0+1) & \\ m+1, & (n_0+1)+1, & (l_0+1)-2\end{pmatrix}}$$

$$\times(-1)^{-\begin{pmatrix} & m+(n_0+1)+(l_0+1) & \\ m+1, & (n_0+1)-1, & (l_0+1)\end{pmatrix}+\begin{pmatrix} & m+(n_0+1)+(l_0+1) & \\ m+2, & (n_0+1)-1, & (l_0+1)-1\end{pmatrix}-\begin{pmatrix} & m+(n_0+1)+(l_0+1) & \\ m+2, & (n_0+1), & (l_0+1)-2\end{pmatrix}}$$

$\therefore n=n_0+1$ 成立

$\therefore l=l_0+1$ 時成立

$\therefore$ 由數學歸納法得知本題得證

四、檢驗「得票數與向量規則對應」之函數是否爲一對一且映成函數之 C++程式碼

(一).由「一路領先」至「E(1,0)、W(-1,0)、N(0,1)、S(0,-1) 向量群」規則之轉換程式：輸入得票之順序，輸出向量之對應結果。(如輸入 aabababcdc，則輸出 EEWENENSWS)

程式碼如下：

```
#include<stdio.h>
#include<string.h>
char last[256];
struct character{
    char c;
    int bottom;
    char original_char;
    int original_count;
    bool used;
};
void copy(character from[],character to[],int len,bool clean=true){
    for(int i=0;i<len;i++){
        to[i]=from[i];
        if(clean)to[i].used=0;
    }
}
void step1_go(char c,int x,character by[],int pos[]){
    while(x>=0){
        if(!by[x].used&&by[x].c==c){
            by[x].used=1;
            pos[by[x].c-'a'+1]=x;
            if(last[c])
                step1_go(last[c],x-1,by,pos);
            break;
        }x--;
    }
}
void step1(character ans[],int len){
    int pos[5],bottom_id=0;
    char tmp;
    for(tmp='d';tmp=='d'||tmp=='c';tmp--)
        for(int i=len-1;i>=0;i--)
            if(!ans[i].used&&(ans[i].c==tmp)){
                pos[0]=ans[i].c-'a'+1;
                step1_go(ans[i].c,i,ans,pos);
                for(int j=1;j<=pos[0];j++)
                    ans[pos[j]].bottom=bottom_id,
                    ans[pos[j]].original_char=ans[pos[j]].c,
                    ans[pos[j]].original_count=pos[0];
                ans[pos[1]].c='E';
                ans[pos[2]].c='N';
                ans[pos[3]].c='S';
                if(pos[0]==4)
                    ans[pos[4]].c='W';
                bottom_id++;
            }
}
bool chk(character str[],int mlen){
    int i,x=0,y=0;
    for(i=0;i<mlen;i++){
        if(str[i].c=='N')y++;
        else if(str[i].c=='S')y--;
        else if(str[i].c=='E')x++;
        else if(str[i].c=='W')x--;
        if(x<y)break;
    }
    return i==mlen;
}
void step2(character ans[],int len){
    int flag=1;
    int i,j,k,s;
    for(i=0;i<len;i++)
```

```
        if(!ans[i].used){
            ans[i].original_count=-2;
            if(ans[i].c=='a')
                ans[i].c='E';
            else if(ans[i].c=='b')
                ans[i].c='W';
        }
}
void step3(character ans[],int len){
    int i,j=-1,k,m,n,YesW,Epos;
    for(k=len-1;k>=0;k--){
        i=k;
    go:
        if(ans[i].c=='S'&&!ans[i].used)
            for(j=i+1;j<len;j++)
                if(ans[j].c=='W'&&!ans[j].used)
                    if(ans[j].original_char!='d'){
                        character tmp=ans[i];
                        ans[i]=ans[j];
                        ans[j]=tmp;
                        int E_count=0,W_count=0;
                        for(int ew=0;ew<i;ew++)
                            if(ans[ew].original_count==-2)
                                if(ans[ew].c=='E')
                                    E_count++;
                                else if(ans[ew].c=='W')
                                    W_count++;
                        if(!chk(ans,len)||E_count<=W_count){
                            ans[j]=ans[i];
                            ans[i]=tmp;
                            for(m=i+1;m<len;m++)
                                if(ans[m].c=='E'&&ans[m].original_count==-2)
                                    break;
                            ans[m].c='N';
                            ans[j].c='S';
                        }
                        i=j;
                        goto go;
                    } else {
                        ans[j].used=1;
                        break;
                    }
    }
}
main(){
    int i,j,k,len;
    char s[101],ans[101];
    character step[4][101];
    last['d']='c',last['c']='b';
    last['b']='a',last['a']='\0';
    while(scanf("%s",s)==1){
        len=strlen(s);
        for(i=0;i<4;i++)
            for(j=0;j<101;j++)
                step[i][j]=(character){0,0,0,-1,0};
        for(i=0;i<len;i++)
            step[0][i].c=s[i];
        copy(step[0],step[1],len);
        step1(step[1],len);
        copy(step[1],step[2],len,0);
        step2(step[2],len);
        copy(step[2],step[3],len);
        step3(step[3],len);
        printf("%s ",s);
        for(i=0;i<len;i++)
            putchar(step[3][i].c);
        putchar('\n');
    }
}
```

(二). 由「E(1,0)、W(-1,0)、N(0,1)、S(0,-1) 向量群」至「一路領先」規則之轉換程式：輸入向量之對應結果，輸出得票之順序。 (如輸入 EEWENENSWS，則輸出 aabababcdc)程式碼如下：

```
#include<stdio.h>
#include<string.h>
main(){
    int i,j,k,r,a,b,len;
    char s[101],ans[101],orig[101];
    bool used[101],flag;
    int bottom[101],bottom_id,count[101],SbyN[101];
    while(scanf("%s",s)==1){
        len=strlen(s);
        strcpy(ans,s);
        strcpy(orig,s);
        for(i=0;i<len;i++){
            bottom[i]=-1;
            SbyN[i]=-1;
        }
        bottom_id=0;
        for(i=0;i<len;i++){
            if(s[i]=='E'&&bottom[i]==-1){
                ans[i]='a';
                count[bottom_id]=1;
                bottom[i]=bottom_id++;
            } else if(s[i]=='N'){
                for(j=i+1;j<len;j++){
                    if(s[j]=='S'&&bottom[j]==-1){
                        SbyN[i]=j;
                        for(k=0;k<i;k++)
                            if(s[k]=='E'&&count[bottom[k]]==1)
                                break;
                        if(k<i){
                            ans[i]='b';
                            ans[j]='c';
                            ans[k]='a';
                            count[bottom_id]=3;
                            bottom[i]=bottom[j]=bottom[k]=bottom_id++;
                        } else {
                            ans[i]='a';
                            ans[j]='b';
                            count[bottom_id]=2;
                            bottom[i]=bottom[j]=bottom_id++;
                        }
                        break;
                    }
                }
            } else if(s[i]=='W'){
                a=b=0;
                for(j=0;j<i;j++){
                    if(s[j]=='N'&&count[bottom[j]]==2){
                        if(SbyN[j]>i){
                            a++;
                        }
                    } else if(s[j]=='S'&&count[bottom[j]]==3)
                        b++;
                }
                if(a<b){
                    ans[i]='d';
                    for(r=i-1;r>=0;r--)
                        if(s[r]=='S'&&count[bottom[r]]==3) //ENS ¥þ¦b W «e
                            break;
                    bottom[i]=bottom[r];
                    count[bottom[r]]=4;
                } else if(a==b){
                    ans[i]='b';
                    for(r=i-1;r>=0;r--)
                        if(s[r]=='E'&&count[bottom[r]]==1)
                            break;
                    bottom[i]=bottom[r];
                    count[bottom[r]]=2;
                }
            }
        }
        for(i=0;i<len;i++)
            used[i]=0;
        for(i=0;i<len;i++){
            k=i;
            flag=1;
            while(flag){
                flag=0;
                if(ans[k]=='c')
                    for(j=k-1;j>=0;j--)
                        if(ans[j]=='b'&&!used[j]){
                            if(s[j]=='W'||s[j]=='S'){
                                s[k]^=s[j]^=s[k]^=s[j];
                                ans[k]^=ans[j]^=ans[k]^=ans[j];
                                used[k]^=used[j]^=used[k]^=used[j];
                                k=j;
                                flag=1;
                            } else
                                used[j]=1;
                            break;
                        }
            }
        }
        printf("%s %s\n",orig,ans);
    }
}
```

(三). 由.「一路領先」至「①(1,1,0)、②(1,-1,1)、③(1,1,-1)、④(1,-1,0)、⑤(1,0,0) 向量群」規則之轉換程式：輸入得票之順序，輸出向量之對應結果。 (如輸入 aababababcdc，則輸出 5141512344。)

程式碼如下：

```
#include<stdio.h>
#include<string.h>
char last[256];
struct character{
  char c;
  int bottom;
  char original_char;
  int original_count;
  bool used;
};
void copy(character from[],character to[],int len,bool clean=true){
  for(int i=0;i<len;i++){
    to[i]=from[i];
    if(clean)to[i].used=0;
  }
}
void step1_go(char c,int x,character by[],int pos[]){
  while(x>=0){
    if(!by[x].used&&by[x].c==c){
      by[x].used=1;
      pos[by[x].c-'a'+1]=x;
      if(last[c]){
        step1_go(last[c],x-1,by,pos);
      }
      break;
    }
    x--;
  }
}
void step1(character ans[],int len){
  int pos[5],bottom_id=0;
  for(char ch='d';ch>='a';ch--){
    for(int i=len-1;i>=0;i--){
      if(!ans[i].used&&(ans[i].c==ch)){
        pos[0]=ch-'a'+1;
        step1_go(ch,i,ans,pos);
        for(int j=1;j<=pos[0];j++){
          ans[pos[j]].bottom=bottom_id;
          ans[pos[j]].original_char=ans[pos[j]].c;
          ans[pos[j]].original_count=pos[0];
        }
        if(pos[0]==1){
          ans[pos[1]].c='5';
        } else if(pos[0]==2){
          ans[pos[1]].c='1';
          ans[pos[2]].c='4';
        } else if(pos[0]==3){
          ans[pos[1]].c='1';
          ans[pos[2]].c='5';
          ans[pos[3]].c='4';
        } else if(pos[0]==4){
          ans[pos[1]].c='1';
          ans[pos[2]].c='2';
          ans[pos[3]].c='3';
          ans[pos[4]].c='4';
        }
        bottom_id++;
      }
    }
  }
}
void step2(character ans[],int len){
  int i,j,k,a,b,c,n,m;
  bool three_dimensional_pair[101];
  character temp;
st:
  for(i=len-1;i>=0;i--){
    if(ans[i].original_count==4&&ans[i].c=='4'&&!ans[i].used){
      for(j=0;j<i;j++){
        if(ans[i].bottom==ans[j].bottom&&ans[j].c=='3'&&!ans[j].used){
          i^=j^=i^=j;
```

```
          for(k=0;k<len;k++){
            if(ans[k].c=='2'&&ans[j].bottom==ans[k].bottom){
              break;
            }
          }
          for(a=0;a<len;a++){
            three_dimensional_pair[a]=0;
          }
          three_dimensional_pair[k]=three_dimensional_pair[i]=1;
          goto La;
        }
      }
    }
  }
  return;
La:
  ans[i].used=ans[j].used=1;
  for(k=i+1;k<j;k++){
    if(ans[k].c=='4'&&ans[k].original_char!='d'){
      goto Lb;
    }
  }
  goto Least;
Lb:
  n=m=0;
  for(a=0;a<i;a++){
    if(ans[a].original_count!=4){
      if(ans[a].c=='1'){
        n++;
      } else if(ans[a].c=='4'){
        m++;
      }
    }
  }
  if(n>m){
    for(a=i+1;a<len;a++){
      if(ans[a].c=='4'&&ans[a].original_char!='d'){
        break;
      }
    }
    temp=ans[a];
    ans[a]=ans[i];
    ans[i]=temp;
    if(three_dimensional_pair[a]!=three_dimensional_pair[i]){
      three_dimensional_pair[a]=!three_dimensional_pair[a];
      three_dimensional_pair[i]=!three_dimensional_pair[i];
    }
    i=a;
    goto La;
  } else if(n==m){
    for(a=i+1;a<j;a++){
      if(ans[a].c=='1'&&ans[a].original_count!=4){
        break;
      }
    }
    for(b=i+1;b<j;b++){
      if(ans[b].c=='4'&&ans[b].original_count!=4){
        break;
      }
    }
    if(a==j||b==j)goto Least;
    ans[a].c='2';
    ans[b].c='3';
    three_dimensional_pair[a]=three_dimensional_pair[b]=1;
    i=b;
    goto La;
  }
Least:
  for(a=len-1;a>=0;a--){
    if(ans[a].c=='3'&&three_dimensional_pair[a]){
      three_dimensional_pair[a]=0;
      for(b=a-1;b>=0;b--){
        if(ans[b].c=='2'&&three_dimensional_pair[b]){
          three_dimensional_pair[b]=0;
        upStep:
          for(k=b;k<=a;k++){
            if(ans[k].c=='5'){
              if(ans[k].original_char=='a'){
                temp=ans[k];
                ans[k]=ans[b];
```

```
              ans[b]=temp;
              if(three_dimensional_pair[b]!=three_dimensional_pair[k]){
                three_dimensional_pair[b]=!three_dimensional_pair[b];
                three_dimensional_pair[k]=!three_dimensional_pair[k];
              }
              b=k;
              goto upStep;
            } else {
              break;
            }
          }
        }
        break;
      }
    }
  }
}
  i++;
  goto st;
}
main(int argc){
  int i,j,k;
  int len;
  char s[101];
  char ans[101];
  character step[4][101];
  last['d']='c';
  last['c']='b';
  last['b']='a';
  last['a']='\0';
  while(scanf("%s",s)==1){
    len=strlen(s);
    for(i=0;i<4;i++){
      for(j=0;j<101;j++){
        step[i][j]=(character){0,0,0,-1,0};
      }
    }
    for(i=0;i<len;i++){
      step[0][i].c=s[i];
    }
    copy(step[0],step[1],len);
    step1(step[1],len);
    copy(step[1],step[2],len);
    step2(step[2],len);
    printf("%s ",s);
    for(i=0;i<len;i++){
      putchar(step[2][i].c);
    }
    putchar('\n');
  }
}
```

(四). 由「(1,1,0)、②(1,-1,1)、③(1,1,-1)、④(1,-1,0)、⑤(1,0,0) 向量群」至「一路領先」規則之轉換程式：輸入得票之順序，輸出向量之對應結果。 (如輸入 5141512344，則輸出 aababababcdc。)

程式碼如下：

```
#include<stdio.h>
#include<string.h>
#include<conio.h>
struct Pos{
  int x,y,z;
}pos[101];
void makePOS(char s[],int len){
  int x=0,y=0,z=0;
  for(int i=0;i<len;i++){
    if(s[i]=='1')x++,y++;
    else if(s[i]=='2')x++,y--,z++;
    else if(s[i]=='3')x++,y++,z--;
    else if(s[i]=='4')x++,y--;
    else if(s[i]=='5')x++;
    pos[i]=(Pos){x,y,z};
  }
}
main(){
  int ii,i,j,k,n,m,a,b,c,e,f,g,h,st,ed,len;
  int can_use[101],up_num,din,last_same;
  int count1,count2,count;
  int special_pair[101],special_pair_count;
  char s[101],ans[101];
  char abcd_1,abcd_2,n_1,n_2;
  bool note[101],used[101],special[101];
  while(scanf("%s",s)==1){
    din=0;
    len=strlen(s);
    ans[len]=0;
    makePOS(s,len);
    for(i=0;i<=len;i++){
      ans[i]='\0';
      used[i]=0;
      special[i]=0;
      can_use[i]=2;
    }
    for(i=0;i<len;i++){
      if(s[i]=='1'){
        ans[i]='a';
      } else if(s[i]=='5'&&pos[i].y==0&&pos[i].z==0){
        ans[i]='a';
      } else if(s[i]=='5'||s[i]=='2'||s[i]=='3'||s[i]=='4'){
        if(s[i]=='5'||s[i]=='2'){
          abcd_1='a';
          abcd_2='b';
          n_1='2';
          n_2='5';
        } else {
          abcd_1='b';
          abcd_2='c';
          n_1='3';
          n_2='4';
        }
        last_same=i+1;
        while(s[last_same-1]==s[last_same]){
          last_same++;
        }
        int total_same=last_same-1;
        st=-999;
        for(j=i-1;j>=0;j--){
          if(ans[j]==abcd_1&&( (s[j]==n_1&&special[j]==0) || s[j]==n_2)){
            st=j+1;
            break;
          }
          if(s[j]=='1'&&(j==0||pos[j-1].y==0&&pos[j-1].z==0)){
            st=j;
            break;
          }
        }
        if(s[i]=='2'||s[i]=='3'){
          special_pair_count=0;
          for(j=0;j<len;j++){
            used[j]=0;
          }
          for(ii=0;ii<last_same;ii++){
```

```
  if(s[ii]=='1'&&!used[ii]){
    for(j=ii+1;j<last_same;j++){
      if(s[j]=='2'&&!used[j]){
        for(k=j+1;k<last_same;k++){
          if(s[k]=='3'&&!used[k]){
            for(m=k+1;m<last_same;m++){
              if(s[m]=='4'&&!used[m]&&ans[m]=='d'){
                used[ii]=used[j]=used[k]=used[m]=1;
                goto st;
              }
            }
            if(m==last_same){
              used[ii]=used[j]=used[k]=1;
              goto st;
            }
          }
        }
        if(k==last_same){
          used[ii]=used[j]=1;
          goto st;
        }
      } else if(s[j]=='5'&&!used[j]){
        for(k=j+1;k<last_same;k++){
          if(s[k]=='2'&&!used[k]){
            for(m=k+1;m<last_same;m++){
              if(s[m]=='3'&&!used[m]){
                for(n=m+1;n<last_same;n++){
                  if(s[n]=='4'&&!used[n]&&ans[n]=='d'){
                    used[ii]=used[j]=used[k]=used[m]=used[n]=1;
                    goto st;
                  }
                }
                if(n==last_same){
                  used[ii]=used[j]=used[k]=used[m]=1;
                  goto st;
                }
              }
            }
            if(m==last_same){
              used[ii]=used[j]=used[k]=1;
              goto st;
            }
          }
          if(s[k]=='4'&&!used[k]){
            used[ii]=used[j]=used[k]=1;
            goto st;
          }
        }
        if(k==last_same){
          used[ii]=used[j]=1;
          goto st;
        }
      } else if(s[j]=='4'&&!used[j]){
        used[ii]=used[j]=1;
        goto st;
      }
    }
  }
  st:;
}
for(ii=0;ii<last_same;ii++){
  if(s[ii]=='2'&&!used[ii]&&special[ii]==1){
    for(j=ii+1;j<last_same;j++){
      if(s[j]=='3'&&!used[j]){
        used[ii]=used[j]=1;
        special_pair[special_pair_count++]=j;
        goto st;
      }
    }
  }
}
if(s[i]=='2'){
  for(last_same--;last_same>=i&&!used[last_same];last_same--){
    ans[last_same]='a';
    special[last_same]=1;
  }
} else {
  m=0;
  for(j=0;j<special_pair_count;j++){
    if(special_pair[j]>=i&&special_pair[j]<last_same){
      m++;
    }
  }
  for(last_same--;last_same>=i&&m;last_same--){
    ans[last_same]='b';
```

```
      special[last_same]=1;
      m--;
    }
  }
  last_same++;
}
if(s[i]=='4'){
  for(m=0;m<i;m++){
    if(can_use[m]==1){
      can_use[m]=2;
    }
  }
  din=0;
s:
  if(din>=last_same-i)goto s_end;
  for(a=0;a<i;a++){
    if(s[a]=='1'&&can_use[a]==2){
      for(b=a+1;b<i;b++){
        if(s[b]=='2'&&can_use[b]==2&&special[b]==0){
          for(c=b+1;c<i;c++){
            if(s[c]=='3'&&can_use[c]==2&&special[c]==0){
              int flag=2;
              k=c;
              while(++k<i){
                if(flag==2&&s[k]=='2'&&special[k]==1&&can_use[k]==2){
                  flag=3;
                  can_use[k]=4;
                } else if(flag==3&&s[k]=='3'&&special[k]==1&&can_use[k]==2){
                  flag=2;
                  can_use[k]=4;
                }
              }
              if(flag==2){
                din++;
                can_use[a]=can_use[b]=can_use[c]=0;
                for(m=0;m<i;m++){
                  if(can_use[m]==4){
                    can_use[m]=0;
                  }
                }
              } else {
                can_use[a]=can_use[b]=can_use[c]=1;
                for(m=0;m<i;m++){
                  if(can_use[m]==4){
                    can_use[m]=1;
                  }
                }
              }
              goto s;
            }
          }
        }
      }
    }
  }
s_end:
  if(din){
    ans[i]='d';
    din--;
    for(j=i+1;j<len&&s[j]==s[j-1];j++){
      if(!din)break;
      din--;
      ans[j]='d';
    }

    if(s[j]!=s[j-1]){
      i=j-1;
      continue;
    } else {
      i=j;
    }
  }
}
count1=count2=0;
for(j=st;j<i;j++){
  if(ans[j]==abcd_1&&special[j]!=1){
    count1++;
  } else if(ans[j]==abcd_2&&special[j]!=1){
    count2++;
  }
}
count=count1-count2;
for(j=i;j<last_same;j++){
  if(count>0){
    ans[j]=abcd_2;
```

```
          count--;
        } else {
          ans[j]=abcd_1;
        }
      }
      i=total_same;
    }
  }
  printf("%s %s\n",s,ans);
 }
}
```

2010 ISEF / Mathematics

Research Paper

Ballot Problem Approached from *n*-dimensional Paths

Justin Tony Hou

and

Te-Wei Hsu

Kaohsiung Municipal

Kaohsiung Senior High School

Taiwan

Abstract

Suppose n candidates A(1),…,A(n) are running for office and a fixed number of votes are cast. In how many ways can the ballots be counted (called the ballot sequences) such that during the counting candidate A(j) is never ahead of the candidate A(i) if $i < j$?

The closed formulae for this very hard and famous enumerative problem (called the Generalized Ballot Problem) are only known for n<6 and remain open for others. Recently a bijection between Motzkin paths and the ballot sequences with three candidates is established. Based on this bijection, we propose in this project the notion of "n-dimensional Motzkin path". This notion will eventually establish the bijection between the higher-dimensional Motzkin paths and the ballot sequences with n candidates for any n. We are happy to announce that in this project the complete proof of the bijective relations has been given for n = 4, 5, 6.

At this stage of investigation, we have strong evidence that the method for attacking the cases n = 4 and 6 can be extended to solving all even cases. A slight modification of the above method will also lead to solving all odd cases.

Introduction

1. Motivation

Suppose two candidates, A and B, are running for office and a hundred ballots are cast. Question: How many ways can the ballots be counted one by one so that candidate B is never ahead of candidate A? Such problem is the famous ballot problem with two candidates [3]. Recently, a bijection between Motzkin paths (a kind of two dimensional paths) and ballot sequences with three candidates was established [1]. As a result, we considered whether higher dimensional paths relating to ballot sequences with more candidates exist.

2. Jargons

2.1. Motzkin path: A Motzkin path is a route from coordinate (0, 0) to coordinate (n, 0) on n steps allowed to move only to the right (up, down or straight) at each step but forbidden from dipping below the x-axis.

2.2. Ballot problem: Suppose candidates A_k are running for office and a fixed number of ballots are cast. Question: How many ways can the ballots be counted one by one so that candidate A_{i+1} is never ahead of candidate A_i?

2.3. Ballot sequence: Let C be the set of candidates. A ballot sequence is a sequence of elements of C following the order of counting. In this project, all ballot sequences are assumed to satisfy the condition of the ballot problem.

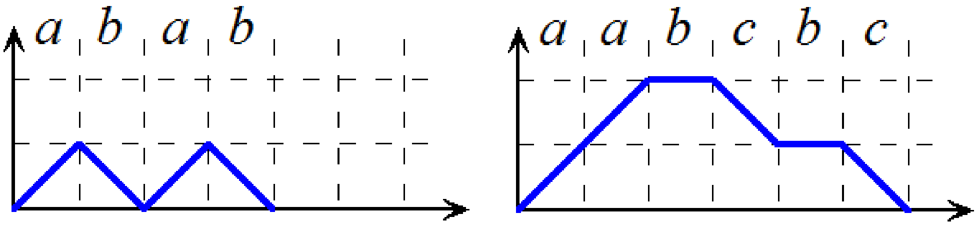


Fig. 1 and Fig. 2 Motzkin paths and ballot sequences

3. Definition and Notations

3.1. Steps: Let $\{\mathbf{e}_1, \cdots, \mathbf{e}_n\}$ denote the standard basis of $\square^n$. Define $v_0 = \mathbf{e}_1$,

$v_1 = \mathbf{e}_1 + \mathbf{e}_2$, $v_2 = \mathbf{e}_1 - \mathbf{e}_2$, $v_{2i-3} = \mathbf{e}_1 - \mathbf{e}_i + \mathbf{e}_{i+1}$, and $v_{2i-2} = \mathbf{e}_1 + \mathbf{e}_i - \mathbf{e}_{i+1}$

for $3 \le i \le n, i \in N$.

3.2. Ballots: We define the votes for Candidate A (A_1), Candidate B (A_2), Candidate C (A_3), …etc. as *a*, *b*, *c*,…etc..

3.3. *n*-dimensional Motzkin Paths:

Let $S = \{v_i \mid 0 \le i \le 2n-2\}$ denote the set of all allowed steps. A path $p_0 p_1 \cdots p_r$ in $\square^n$ is a sequence of $(r+1)$ elements of $\square^n$ with non-negative integral coordinates such that:

(1). p_0 is the origin

(2). $p_r = (r, 0, \cdots, 0)$, $r > 0$

(3). $p_k - p_{k-1} \in S$

In case *n*=2, this reduces to the paths originally studied in 1977 [4].

4. Goals

4.1. Formulate the notion of higher dimensional paths to deal with more candidates.

4.2. Establish the bijections between higher dimensional paths and ballot sequences.

4.3. Explore the mathematical structure of the set of all bijections defined above.

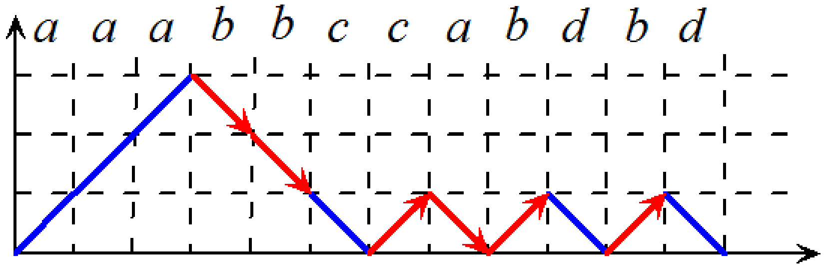


Fig. 3 A higher dimensional path and a ballot sequence

Materials

1. Equipments

1.1. PC, papers and pens

1.2. Models

1.3. Poker cards

2. Softwares

2.1. Microsoft Word 2003

2.2. Cabri 3D

2.3. Dev C++ 4.9.9.2

2.4. Maple 12

Methods and Results

1. Exploration of higher dimensional Motzkin paths and conjectures

1.1. According to [1], the following results are known:

Lemma 1.1. For two candidates: Ballot sequences with two candidates have a one to one correspondence with Motzkin paths with the restriction that the level steps (v_0) appear only on axis y=0.

Lemma 1.2. For three candidates: Ballot sequences with three candidates have a one to one correspondence with Motzkin paths.

1.2. Based on Motzkin paths, we add two extra steps $v_3 = (1,-1,1)$, $v_4 = (1,1,-1)$ to construct 3-dimensional Motzkin paths. Through the computer programs, we have the following sequences:

Seq1. Ballot sequences with four candidates (number sequence with length *k*):

1, 2, 4, 10, 25, 70, 196, 588, 1764, 5544, 17424, 56628, 184041, 613470…

Seq2. 3-dim. Motzkin paths with the restriction that the level steps (v_0) appear only on plane *z*=0 (number sequence with length *k*):

1, 2, 4, 10, 25, 70, 196, 588, 1764, 5544, 17424, 56628, 184041, 613470…

Seq3. Ballot sequences with five candidates (number sequence with length *k*):

1, 2, 4, 10, 26, 75, 225, 715, 2347, 7990, 27908, 99991, 365587, 1362310…

Seq4. 3-dim. Motzkin paths (number sequence with length *k*)

1, 2, 4, 10, 26, 75, 225, 715, 2347, 7990, 27908, 99991, 365587, 1362310…

Surprisingly, we notice that Seq1. is equal to Seq2., and Seq3. is equal to Seq4.. This leads to our conjectures:

Conjecture 1.1. Ballot sequences with four candidates have a one to one correspondence with 3-dim. Motzkin paths with the restriction that the level steps (v_0) appear only on plane z=0.

Conjecture 1.2. Ballot sequences with five candidates have a one to one correspondence with 3-dim. Motzkin paths.

1.3. Based on 3-dim. Motzkin paths, we add two extra steps $v_5 = (1, 0, -1, 1)$, $v_6 = (1, 0, 1, -1)$ to construct 4-dim. Motzkin paths. Through the computer programs, we have the following sequences:

Seq5. Ballot sequences with six candidates (number sequence with length *k*):

1, 2, 4, 10, 26, 76, 231, 756, 2556, 9096, 33231, 126060, 488488…

Seq6. 4-dim. Motzkin paths with the restriction that the level steps (v_0) appear only on hyperplane *u*=0 (number sequence with length *k*):

1, 2, 4, 10, 26, 76, 231, 756, 2556, 9096, 33231, 126060, 488488…

Seq7. Ballot sequences with seven candidates (number sequence with length *k*):

1, 2, 4, 10, 26, 76, 232, 763, 2611, 9415, 35135, 136335, 544623…

Seq8. 4-dim. Motzkin paths (number sequence with length *k*)

1, 2, 4, 10, 26, 76, 232, 763, 2611, 9415, 35135, 136335, 544623…

Surprisingly, we notice that Seq5. is equal to Seq6. and Seq7. is equal to Seq8.. This leads to our conjectures:

Conjecture 1.3. Ballot sequences with six candidates have a one to one correspondence with 4-dim. Motzkin paths with the restriction that the level steps (v_0) appear only on hyperplane u=0.

Conjecture 1.4. Ballot sequences with seven candidates have a one to one correspondence with 4-dim. Motzkin paths.

1.4. Through the computer programs and the former conjectures, we give out the following conjectures for general cases:

Conjecture 1.5. Let $\{\mathbf{e}_1, \cdots, \mathbf{e}_n\}$ denote the standard basis of $\mathbb{R}^n$. Ballot sequences with 2n-2 candidates have a one to one correspondence with n-dim. Motzkin paths with the restriction that the level steps (v_0) appear only on the hyperplane spanned by $\{\mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_{n-1}\}$.

Conjecture 1.6. Ballot sequences with 2n-1 candidates have a one to one correspondence with n-dim. Motzkin paths.

2. Alteration of the algorithms introduced in [1]

2.1. Examples: the bijective relationship between ballot sequences with three candidates and Motzkin paths.

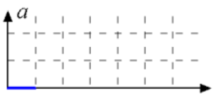


Fig. 4.1

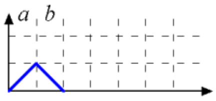


Fig. 4.2

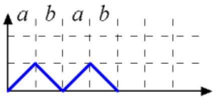


Fig. 4.3

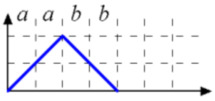


Fig. 4.4

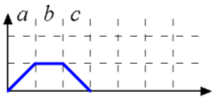


Fig. 4.5

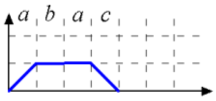


Fig. 4.6

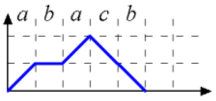


Fig. 4.7

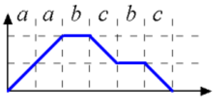


Fig. 4.8

2.2. The bijection between the set of all ballot sequences with three candidates and the set of all Motzkin paths:

ϕ: Ballot sequences to Motzkin paths

Step1: From right to left, map each *a* to v_0.

Step2: From right to left, map each *b* to v_2 and change the nearest v_0 on its left to v_1.

Step3: From right to left, map each *c* to v_2 and change the nearest v_2 on its left to v_0.

ϕ^{-1} :Motzkin paths to ballot sequences

Step1: From left to right, change each v_0 whose starting point is not on the *x*-axis to v_2, and map the nearest v_2 on its right to *c*.

Step2: From left to right, change each v_1 to v_0, and map the nearest v_2 on its right to *b*.

Step3: Map each v_0 to *a*.

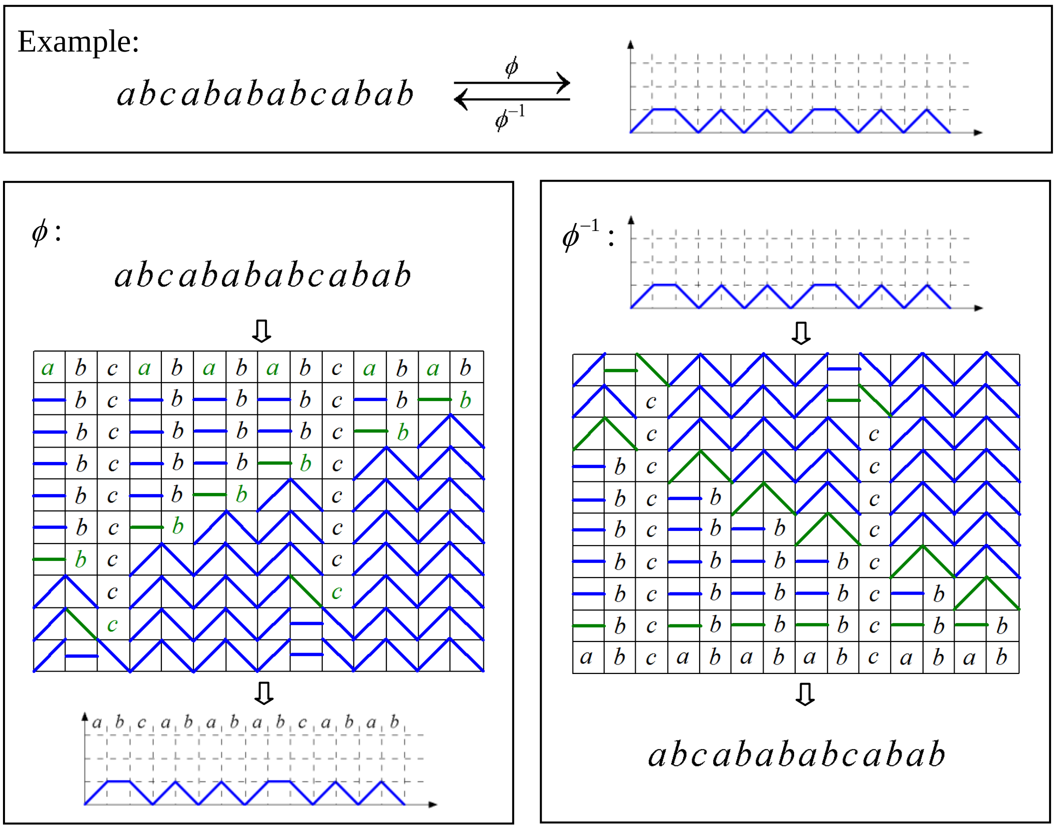


Fig. 5 An example of the mapping (three candidates)

2.3. Through our algorithms, we prove:

Lemma 1.1. Ballot sequences with two candidates have a one to one correspondence with Motzkin paths with the restriction that the level steps (v_0) appear only on axis y=0.

Lemma 1.2. Ballot sequences with three candidates have a one to one correspondence with Motzkin paths.

2.4. By our algorithms, we have the important relationship:

$$\text{Motzkin paths (3 candidates)} \xrightarrow[\textbf{only lie on axis } y=0]{\textbf{retain those paths whose level steps}} \text{Motzkin paths (2 candidates)}$$

3. Bijective relationship between 3-dim. Motzkin paths and ballot sequences with four and five candidates

3.1. Examples: The bijective relationship between ballot sequences with four candidates and 3-dim. Motzkin paths (We project the steps $v_3 = (1,-1,1)$, $v_4 = (1,1,-1)$ on *x*-*y* plane in red color)

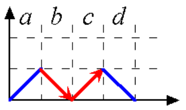


Fig. 6.1

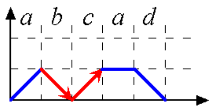


Fig. 6.2

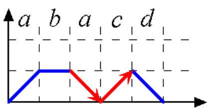


Fig. 6.3

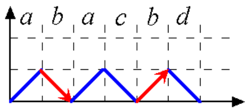


Fig. 6.4

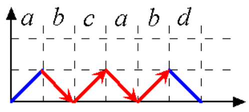


Fig. 6.5

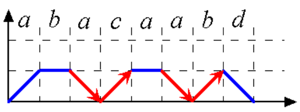


Fig. 6.6

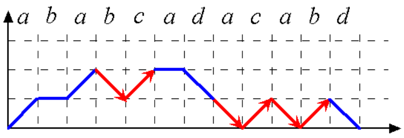


Fig. 6.7

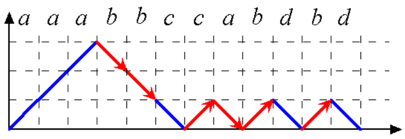
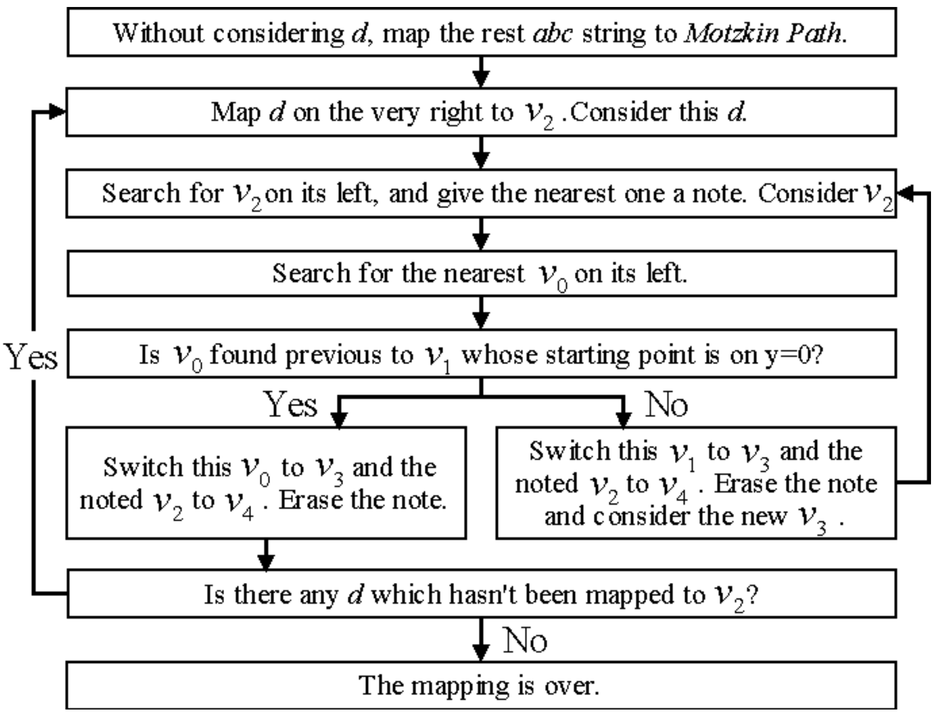


Fig. 6.8

3.2. The bijection between the set of all ballot sequences with four candidates and 3-dim. Motzkin paths with the restriction that the level steps (v_0) appear only on plane *z*=0:

ϕ :Ballot sequences to 3-dim. Motzkin paths



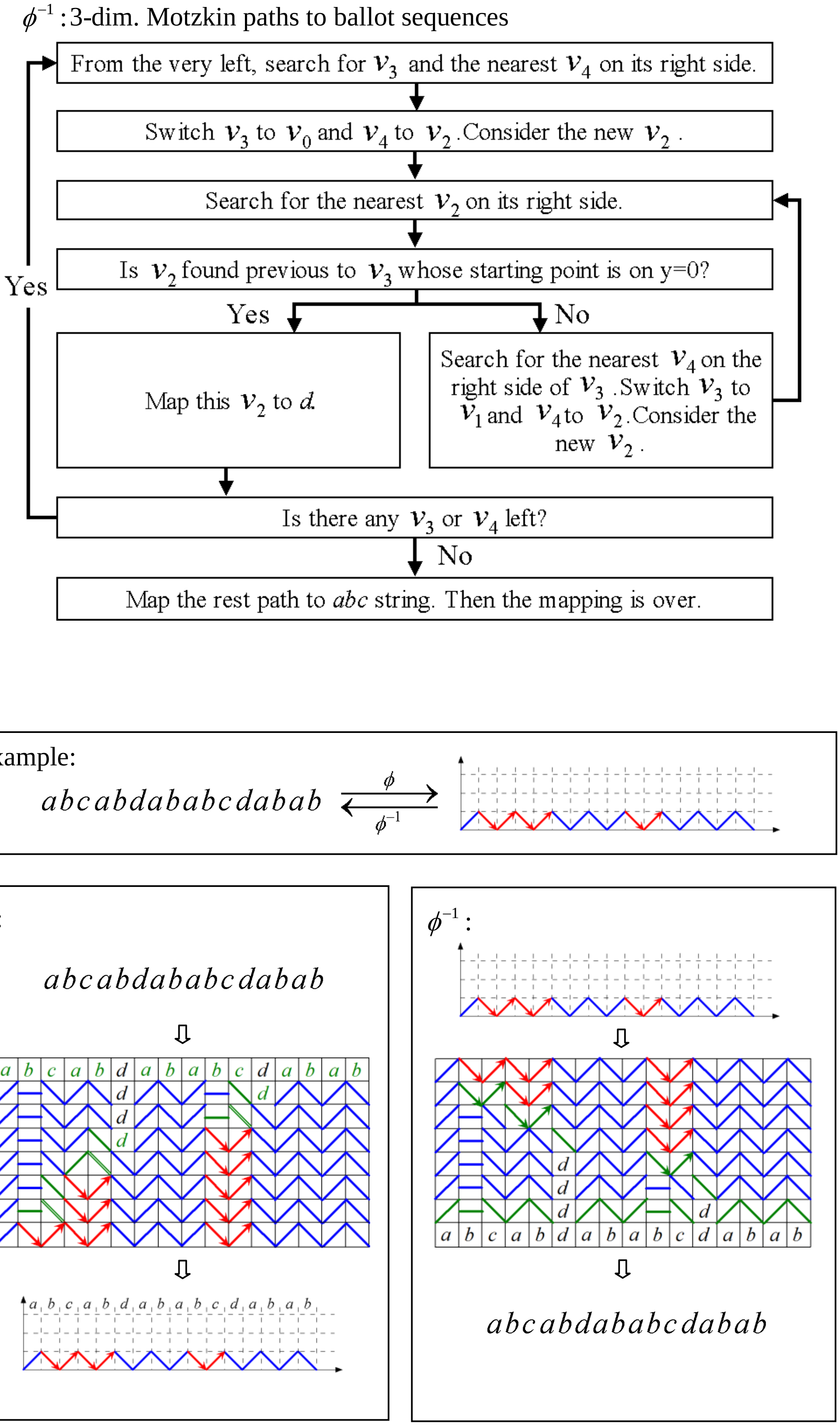


Fig. 7 An example of the mapping (four candidates)

3.3. Examples: the bijective relationship between ballot sequences with five candidates and 3-dim. Motzkin paths (We project the steps $v_3 = (1,-1,1)$, $v_4 = (1,1,-1)$ on *x*-*y* plane in red color)

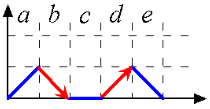


Fig. 8.1

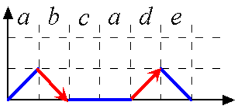


Fig. 8.2

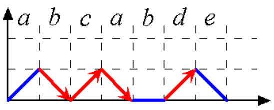


Fig. 8.3

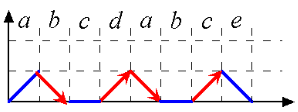


Fig. 8.4

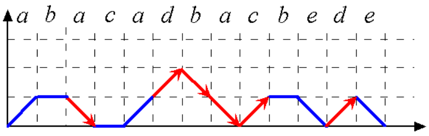


Fig. 8.5

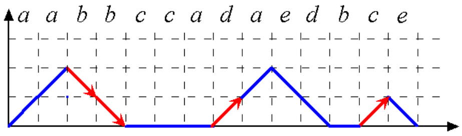
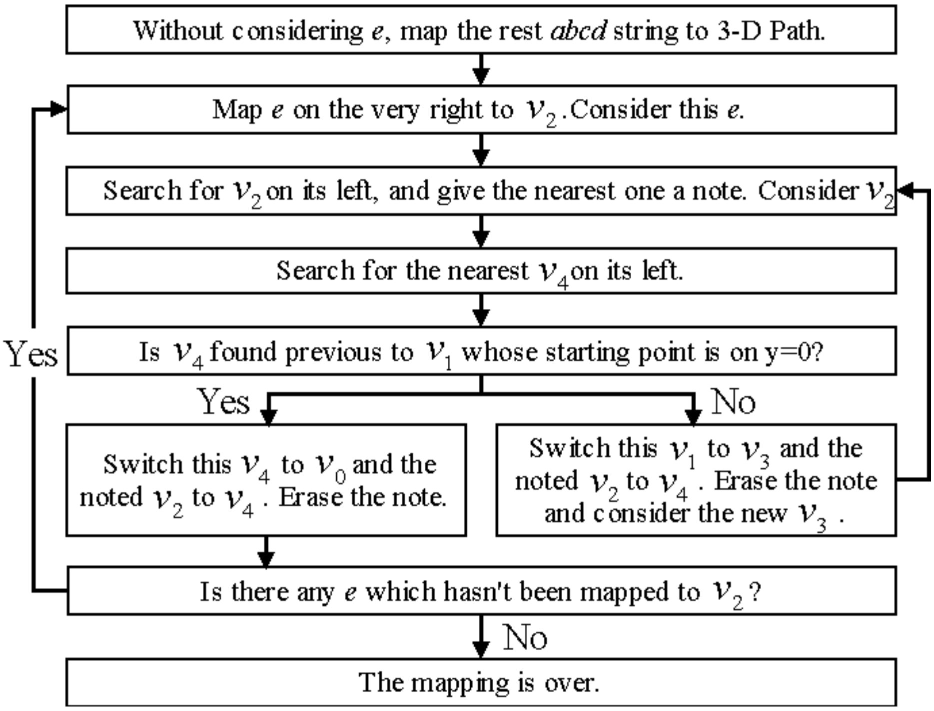


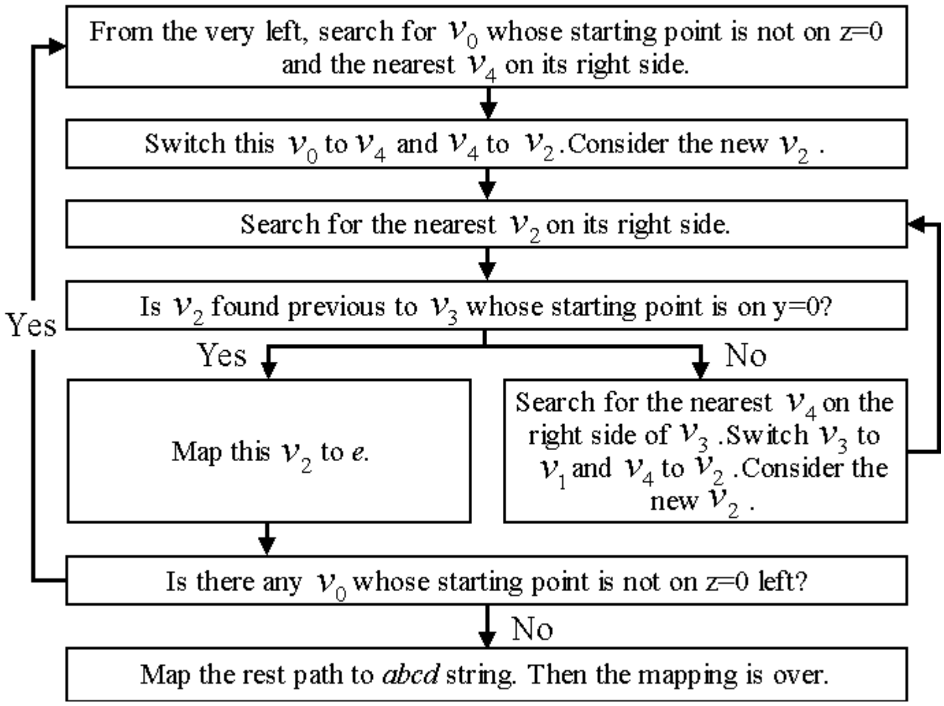
Fig. 8.6

3.4. The bijection between the set of all ballot sequences with five candidates and 3-dim. Motzkin paths:

ϕ :Ballot sequences to 3-dim. Motzkin paths



ϕ^{-1} : 3-dim. Motzkin paths to ballot sequences



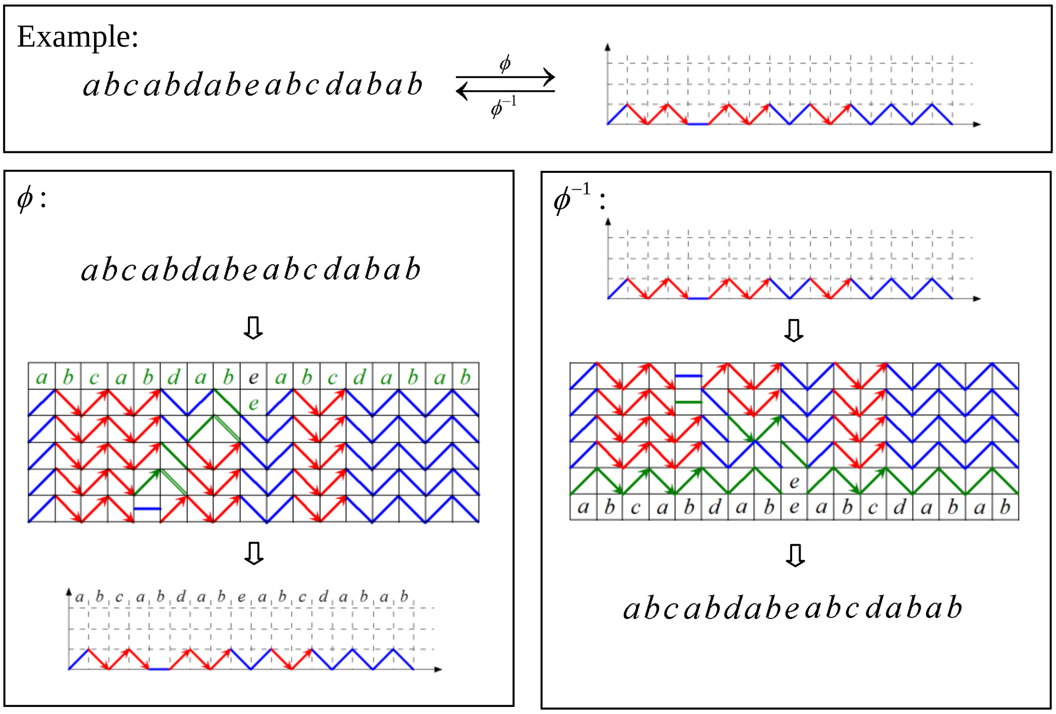


Fig. 9 An example of the mapping (five candidates)

3.5. By our algorithms, we have the important relationship:

3-dim. Motzkin paths **(5 candidates)** $\xrightarrow[\text{only lie on plane } z=0]{\text{retain those paths whose level steps}}$ 3-dim. Motzkin paths **(4 candidates)**

4. Bijective relationship between 4-dim. Motzkin paths and ballot sequences with six and seven candidates

4.1. Examples: The bijective relationship between ballot sequences with six candidates and 4-dim. Motzkin paths (We project the steps $v_5 = (1, 0, -1, 1)$, $v_6 = (1, 0, 1, -1)$ on *x*-*y* plane in purple color. Arrowhead—right: v_5, left: v_6)

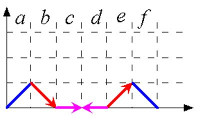


Fig. 10.1

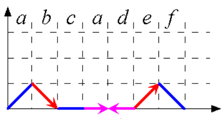


Fig. 10.2

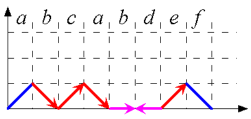


Fig. 10.3

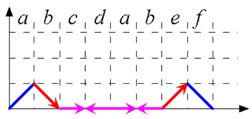


Fig. 10.4

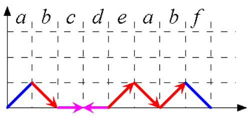


Fig. 10.5

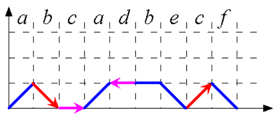


Fig. 10.6

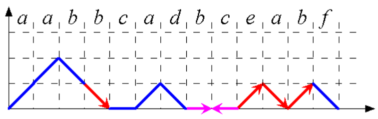


Fig. 10.7

4.2. The bijection between the set of all ballot sequences with six candidates and 4-dim. Motzkin paths with the restriction that the level steps (v_0) appear only on hyperplane *u*=0. Bijections are omitted.

4.3. Examples: The bijective relationship between ballot sequences with seven candidates and 4-dim. Motzkin paths (We project the steps $v_5 = (1, 0, -1, 1)$, $v_6 = (1, 0, 1, -1)$ on *x*-*y* plane in purple color. Arrowhead—right: v_5, left: v_6)

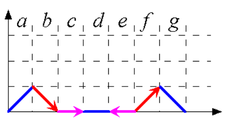


Fig. 11.1

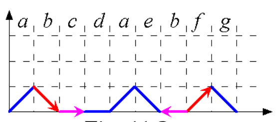


Fig. 11.2

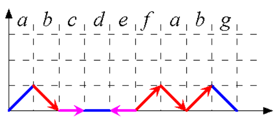


Fig. 11.3

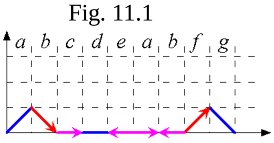


Fig. 11.4

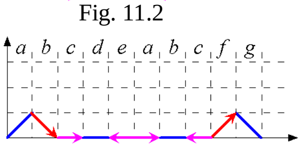


Fig. 11.5

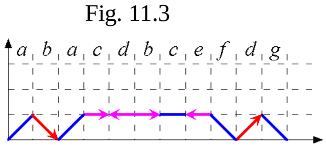


Fig. 11.6

4.4. The bijection between the set of all ballot sequences with seven candidates and 4-dim. Motzkin paths. Bijections are omitted.

4.5. By our algorithms, we have the important relationship:

4-dim. Motzkin paths **(7 candidates)** $\xrightarrow[\textbf{only lie hyperplane } u\textbf{=0}]{\textbf{retain those paths whose level steps}}$ 4-dim. Motzkin paths **(6 candidates)**

5. Bijective relationship between *n*-dim. Motzkin paths and ballot sequences with 2*n*-2 and 2*n*-1 candidates

5.1. By the above bijections, we can categorize the cases into two main situations: the cases with even number of candidates and the cases with odd number of candidates.

5.2. Let $(x_1, x_2, \cdots, x_n) \in \mathbb{R}^n$ and let $\{\mathbf{e}_1, \cdots, \mathbf{e}_n\}$ denote the standard basis of $\mathbb{R}^n$ .We establish the bijection between the set of all ballot sequences with 2*n*-2 candidates ($a_1 a_2 \cdots a_{2n-2}$) and the set of *n*-dim. Motzkin paths with the restriction that the level steps (v_0) appear only on the hyperplane spanned by $\{\mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_{n-1}\}$. Bijections are omitted.

5.3. Let $(x_1, x_2, \cdots, x_n) \in \mathbb{R}^n$ .We establish the bijection between the set of all ballot sequences with 2*n*-1 candidates ($a_1 a_2 \cdots a_{2n-1}$) and the set of all *n*-dim. Motzkin paths.

5.4. By our algorithms, we have the important relationship:

$$\begin{array}{c} n\text{-dim. Motzkin paths} \\ \textbf{(2}\boldsymbol{n}\textbf{-1 candidates)} \end{array} \xrightarrow[\textbf{the hyperplane which the last coordinate=0}]{\textbf{retain those paths whose level steps only lie on}} \begin{array}{c} n\text{-dim. Motzkin paths} \\ \textbf{(2}\boldsymbol{n}\textbf{-2 candidates)} \end{array}$$

6. The mathematical structure of the set of all bijections

Let M_k denote the set of all *k*-dimensional paths and let T_k be the bijective image of k -candidate ballot sequences. To explore the structure of the bijections, we fix a nested embeddings of the vector spaces $\mathbb{R}_1 \subset \mathbb{R}_2 \subset \cdots \subset \mathbb{R}_n \subset \mathbb{R}_{n+1} \subset \cdots$ by identifying the vector *v* in $\mathbb{R}^n$ with the vector (*v*,0) in $\mathbb{R}^{n+1}$. In this way, M_i can be regarded as a subset of M_{i+1}. In addition, T_2 is in fact a proper subset of all M_2 and this fact can be extended to higher dimensions: T_{2n-2} is a proper subset of M_n which is exactly T_{2n-1}. Therefore, we have $T_1 \subset T_2 \subset \cdots \subset T_n \subset T_{n+1} \subset \cdots$. In this sense, we have established a nested embedding of the set of all ballot sequences into the set of all generalized Motzkin paths.

Applications

1. By using the paths, we can apply it to counting the winning probability that candidate A_{i+1} is never ahead of A_i during the rest votes counting process.

   e.g. Regev's Conjecture and its generalized problem for more candidates.

   Regev's Conjecture: Given $\mu=(\mu_1,\mu_2,\mu_3)$ a partition of at most three parts, let $|\mu|:=\mu_1+\mu_2+\mu_3$ and $T_3(\mu;n-|\mu|)$ be the set of SYTs with $n-|\mu|$ entries in the "skew strip" T_3/μ. Regev conjectured that for $\mu=(2,1,0)$,

   $$|T_3((2,1,0);n-3)|=m_{n-1}-m_{n-3},$$

   a difference of two Motzkin numbers.

2. By the concept of higher dimensional paths and our algorithms, we can apply them to solving the problem for ballot-lattice paths and other related questions. The following question is a typical one among them:

   Count the number of underdiagonal lattice paths in the first quadrant, going from (0,0) to a point on the *x*-axis and consisting of *n* steps from {E=(1,0), W=(-1,0), N=(0,1), S=(0,-1)}.

   It's related to the cases with four candidates, and the bijection between paths and ballot sequences is omitted.

3. Playing cards can be used as our analog computer, being able to demonstrate the process of the algorithms.

   (1) Use Spade, Heart, Diamond, Club, to indicate Candidate A, B, C, D.

   (2) By flipping the playing cards, we use the backside of the playing cards to illustrate the paths, and use the other side to illustrate ballot sequences.

   We can use our "analog computer" to demonstrate our algorithms, and we notice that the process of mapping from ballot sequences to paths and the process of mapping from paths to ballot sequences are inverse through our analog computer. The processes of the mappings are given in the paper previously.

Future works

1. Use the notion of nested embeddings and try to figure out the close formulae for the cases with more than five candidates.
2. Observe the relations between each model which counts the number of ballot sequences with a fixed number of candidates.

Acknowledgement

With warm thanks to Prof. J.-C. Chuan of NTHU for his great help and guidance in presentation and research paper; to Prof. S.-P. Eu of NUK for his instruction in research development and structure; to Prof. J.-Y. Lin of Academia Sinica for his suggestion in posters and checking in references.

Warm thanks also go to our high school homeroom teacher Y.-T. Chung, who is our math teacher as well, for his three-year teaching and guidance in math, and his valuable instruction of our project; to our classmates K.-H. Tseng and S.-T. He, who give us advice and inspiration of our research; also to our junior high school math teachers J.-M. Lu of YMJH and Y.-F. Hsieh of MHJH for their enlightening teaching.

We would also like to express our appreciation to our school, Kaohsiung Junior High School, Taiwan; and classmates and friends in school who have supported us to fulfill our dream finishing this project since the day we began our research, and those who have helped us improving our presentation and report. Without them, this research will be incomplete for sure.

We also want to express our special thanks to T.-Y. Cheng, who has helped and given us advice for the computer programs which have supported our findings and let us implement our research.

Finally we would like to say thanks to our parents: T.-H. Hou and M.-H. Chiu, T.-M. Hsu and W.-C. Liao. They have encouraged us and showed their best support while we were doing the research. And, most of all, their invaluable love has accompanied us throughout all the time.

References

[1] S.-P. Eu, Skew-standard tableaux with three rows, accepted for publication in *Advances in Applied Mathematics*.

[2] R. Stanley, Enumerative Combinatorics, Volume 2, *Cambridge Studies in Advanced Mathematics* **62**, Cambridge University Press, Cambridge, 1999.

[3] D. André, Solution directe du problème résolu par M. Bertrand, *Comptes Rendus de l' Académie des Sciences*, Paris **105**, 1887, 436 – 437.

[4] R. Donaghey and L.W. Shapiro, Motzkin Numbers, *Journal of Combinatorial Theory, Series A* **23**, 291-301, 1977.