2020科展作品介紹

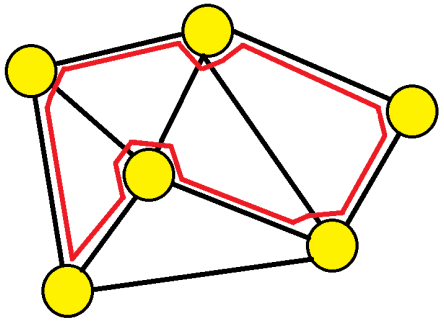
第五組:陳怡傑、黃擎天、
蔡孟勳、陳仕銘、李柏勛

目錄

研究主題：

改良式廣度優先網路爬蟲演算法之組合分析

研究動機



- 高一數學課程上到排列組合單元，研究一個排列錯排個數的問題，叫做錯排問題。這跟著名的帽子問題或拿名片的問題一樣
- 在資訊課程學到有關網路爬蟲的概念。網路爬蟲是一種自動流覽全球資訊網的網路機器人，為搜尋引擎的重要組成部分，其目的為搜索網站，盡量不要搜索重覆的網站。這樣情形如同一個給定的圖，假設圖中兩點u, v間有一個點串列，在點串列中任意相鄰之點都有邊相連，即表示兩點間存在一條路徑。在一個圖中，判斷是否存在 著一條恰好通過所有的點，並且沒有重複經過的路徑。從這個觀點來看是一筆畫問題，也就是哈密頓路徑(Hamiltonian path—在圖中存在一條路徑通過且僅通過每一個頂點一次。 而閉合的哈密頓路徑稱作哈密頓迴路(Hamiltonian cycle)。

研究目的

- 我們期望以組合的觀點進行分析。並應用至網路爬蟲的最佳化上。
- （一）探討k-錯排相關性質。
  1. 推導$|D_{k,n}|$的遞迴關係式。
  2. 推導於 k-錯排下循環組數的遞迴關係式。
  3. 提出 k-錯排演算法。
  4. k-錯排數值分析。
  5.開放問題$|D_{k,n}| \equiv$ (mod k)
- （二）分析網路爬蟲最佳化問題。
  1. 分析網路爬蟲的遍歷行為。
  2. 提出網路爬蟲最佳爬行疆域。
  3. 如何改良分散式網路爬蟲？

名詞介紹

- m cycle －（循環組）

如圖1，上列代表著一數列，而下列表該數列每項所對應的數，用代數中群論的方法可表示為(135)(24)。

其中(135)(1→3→5→1)為一個3-cycle ,(24)(2→4→2)為一個2-cycle（如圖2）

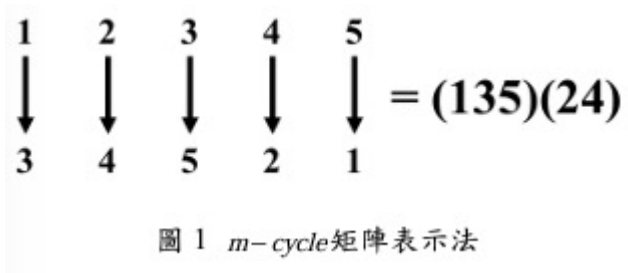


圖 1 *m－cycle*矩陣表示法

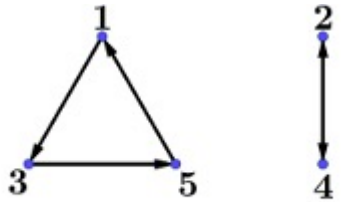


圖 2 *m－cycle*圖形表示法

名詞介紹

- k-錯排

設s_n表n個相異元素(假設為1, 2, …, n連續整數)所有排列所成集合。若σ∈s_n且對所有的X⊆{1,2,.....,n}且|X|=k,滿足σ(X) ≠X,稱σ為k-錯排。所有k-錯排所形成的集合以$D_{k,n}$表示。假如|X|=1 ，也就是n個相異元素進行簡單排列，排列後每個元素都不在原來的位置上，此時這樣的排列稱為一般的錯排。錯排又稱不定點排列，即不產生1- cycle，也就是1-錯排。

例如： $D_{1,4}$為4個相異元素所有排列中，1-錯排所成集合，即 $D_{1,4}$ = {(1234), (1243), (1324), (1342), (1432), (1423), (12)(34), (13)(24), (14)(23)}。

k-錯排循環組數

- 當 k =1,2,3 時，我們可以統整出 fk(n,j)的遞迴關係式，如下所示
- F_1(n,j)=(n-1) F_1(n-2,j-1)+(n-1) F_1(n-1,j)
- F_2(n,j)=C_1^n×F(n-1,3,j-1)+f(n,3,j)
- F_3(n,j)= C_1^n×F(n-1,4,j-1)+C_2^n×F(n-2,4,j-2)+G(n,3,j)

得知與k互質且小於k的m-cycle 情況與j呈現正相關，因此當 k=4時，3-cycle 的情況會隨著j增加而增加，但目前我們無法採用G(n,4,j)來解決， 因為G(n,4,j)會包含產生兩個或兩個以上的2-cycle 將不符合4-錯排。因此當k≧時，目前只能分別求出其遞迴關係式。

k-錯排演算法

- 除了遞迴關係式外，也嘗試使用 C 語言透過演算法，進行數值模擬。
- $|D_{k,n}|$的演算法　輸入：$|D_{k,n}|$的k，n 輸出：$|D_{k,n}|$
- 步驟 1　sum ← 0；num ← 1
- 步驟 2　l← 1
- 步驟 3　l_l={1, 2, 3, ..., l}
- 步驟 4　$\sum_{i\in l} ni$=n
- 步驟 5　選出$\sum_i ni \neq k$所有 i 的組合，如果沒有，則跳至步驟 12，否則往下執行
- 步驟 6　若選出的n_i組合中的元素個數 ≠j ，則回到步驟 5。
- 步驟 7　記錄此分割有多少相同分區；x ← n。

k-錯排演算法

- 步驟 8　　l← 計算n_i的相同數
- 步驟 9　　num=num(C_{n1}^{x} ×(n_1-1)!)
- 步驟 10　 x=x- n_1
- 步驟 11　 若 x = 0 往下執行，否則回到步驟 9。
- 步驟 12　 運算num=num/j! ,sum=sum+num ,num =1 ,回到步驟 5。
- 步驟 13　 輸出 sum

數值分析

- $|D_{k,n}|$的值:我們將透過遞迴關係式以及演算法 1 求得的 k-錯排的方法數整理成下表 1。在表 1 中發 現其值最大的是 $|D_{5,10}|$ 。除此之外，我們可以觀察到對稱性，也就是$|D_{k,n}|$= $|D_{k-n,n}|$ ，這是由於 當 n 分割成 k 及 n－k 兩個整數時，若任意 k 個元素所形成的子集，經過簡單排列後，沒有對 應到自己本身所成集合，意味著另外 n－k 個元素所形成的子集，經過簡單排列後，也無法對應到自己本身所成集合。而根據表 2 歸納，我們觀察到當 n 越大，且 k 越接近n/2時， $|D_{k,n}|$的值越大。

表一、表二

表 1：$|D_{k,n}|$之規律性

k / n	1	2	3	4	5	6	7	8	9	10
1	0	1	1	1	1	1	1	1	1	1
2	1	0	2	2	2	2	2	2	2	2
3	2	2	0	6	6	6	6	6	6	6
4	9	14	9	0	24	24	24	24	24	24
5	44	54	54	44	0	120	120	120	120	120
6	265	304	459	304	265	0	720	720	720	720
7	1854	2260	2568	2568	2260	1854	0	5040	5040	5040
8	14833	18108	20145	26704	20145	18108	14833	0	40320	40320
9	133496	161756	176076	200240	200240	176076	161756	133496	0	362880
10	1334961	1618496	1833741	1931616	2492225	1931616	1833741	1618496	1334961	0

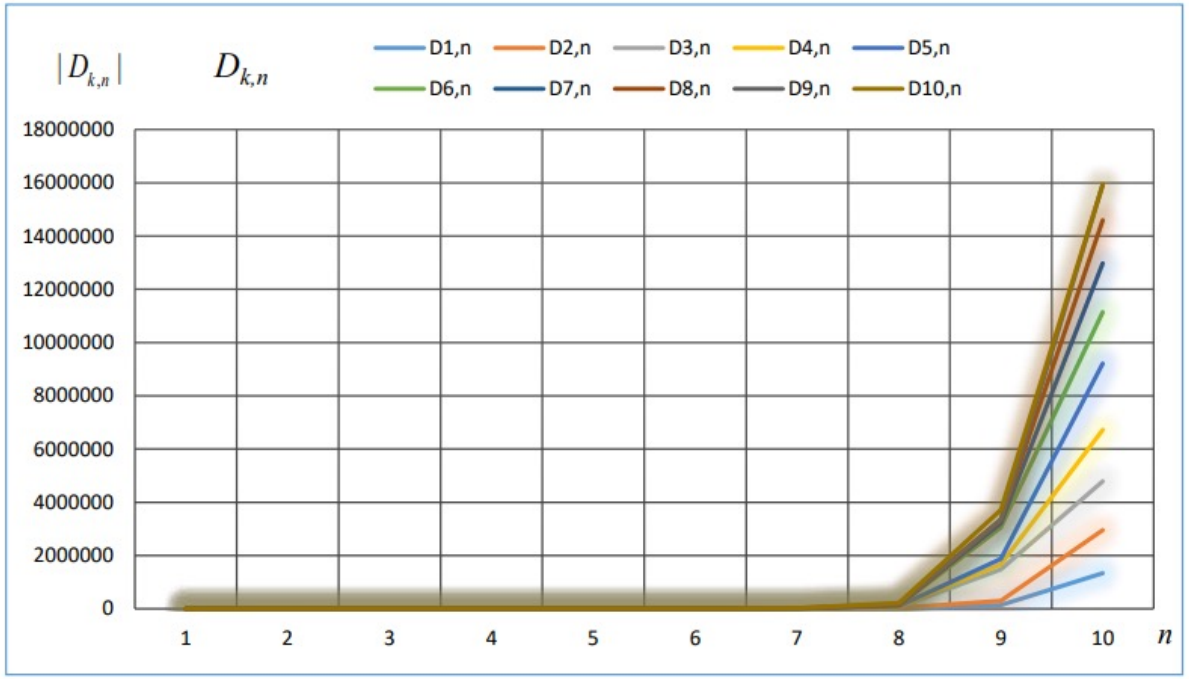


圖 7 $|D_{k,n}|$趨勢圖

網路爬蟲

一種用來自動瀏覽全球資訊網的網路機器人，其目的一般為編纂網路索引及網路搜尋引擎等。透過爬蟲更新自身的網站內容或其對其他網站的索引，它可以將自己所存取的頁面儲存下來，以便搜尋引擎事後生成索引供用戶搜尋。分散網路爬蟲主要為透過多個網路爬蟲遍歷全球資訊，如圖3所示。

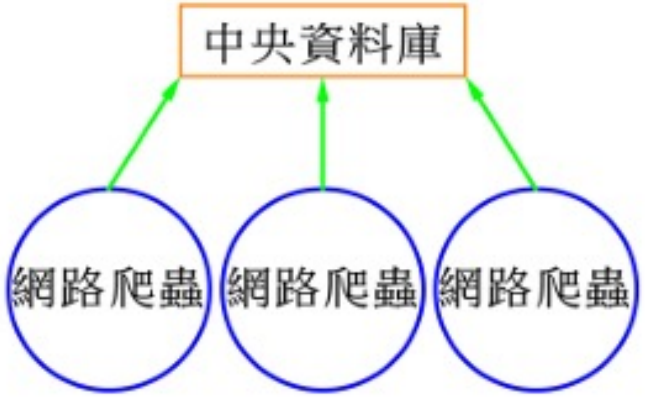


圖 3 分散式網路爬蟲

網路爬蟲

- 若n個相異物進行排列所對應的圖為G =(V,E) ，其中 =|V(G)|=n且 E(G)為此n個相異物進行排列所對應之關係。則$D_{k,n}$可以表示為$D_{k,Kn}$、F_k(n,j)可以表示F_k(K_n,j)。

而$D_{k,Kn}$就可以表示為當有 n 個網站連接情形為K_n時，符合 k-錯排的爬行疆域所形成的集合。$F_k(K_n,j)$即可表示為當有 n 個網站連接情形為K_n ，共有 j 隻網路爬蟲進行搜索時， 符合 k-錯排的爬行疆域所形成的集合。

而網路爬蟲進行遞迴時，須要參數作為指引，往哪裡進行遞迴搜索，效率會比較好或情況會比較多。當網站間連結情形為K_6時，如下表 3 所示，網路爬蟲往 k=3 的方向遞迴，其情 況數會比 k=1,2,4,5 時來得多。

表三

表 3：網站間連結情形為 K_6

k	1	2	3	4	5
K_6	265	304	459	304	265

爬行疆域

- 網路爬蟲進行搜索時，會先編纂出頁面上所有的超連結，並將它們寫入一張「待訪列表」，即所謂爬行疆域，並依照此表得順序進行搜索。在研究中我們透過圖論方法來探討。圖是由一些頂點和邊所組成的結構。設圖G所有頂點所成的集合為V (G)所有邊所成集合為E(G) ，若圖G中有n個頂點且任兩點都有邊相連，我們就稱圖G為n-完全圖，以K_n表示。若圖H及圖G滿足如果V(H)⊆V(G)且E(H)⊆E(G),我們稱H為G的子圖[3]，如圖5就是圖4的一個子圖。

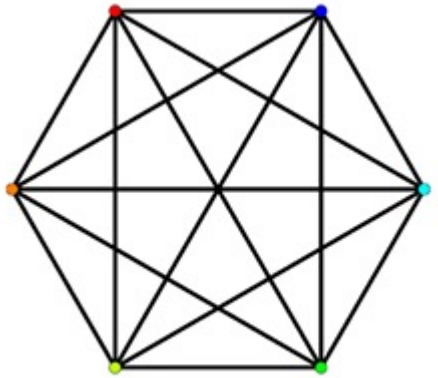


圖4 完全圖 K_6 (網站間連結情形)

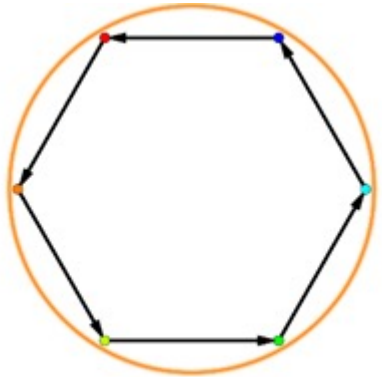


圖5 K_6 的子圖(其中一種爬行疆域)

最佳爬行疆域

- 假設網路爬蟲處理量沒有限制時，當 n 個相異物進行直線排列後，任取 k(k<n)個元素不完全相同時，同時也只產生一個 cycle ，即表示僅由一隻爬蟲進行運作即可，即為單一網路爬蟲的最佳利用也就是解決哈密頓路徑問題(Hamiltonian path problem)，據此提出定理20， 以優化解決廣度優先的網路爬蟲問題。
- 定理20：當 n 個相異物進行直線排列後，只產生 1 個 cycle 的情況，所對應的爬行疆域即達到單一網路爬蟲最佳利用 。
- 證明:

排列情況中，對於所有正整數k(k<n)，符合所有 k-錯排的排列情況，在整數分割分類中，只有 n 的情況，因為此分類只會產生一個 n-cycle，而在只有 n-cycle 的情況下，必不符合 n-錯排，且符合 1-錯排~ n −1-錯排，由此得證。

網路爬蟲改良說明

分散式網路爬蟲演算法改良

Step 1：將整個網路的所有 URL 檢視一次判斷是否為所要的特定集合 A={1,2,3} or B={4,5} or C={ } or …。若判斷為非特定的集合，則不做處置。

Step 2：判斷 Step 1 的特定集合 A or B or C or …是否為空集合：

(a) 若為非空集合(如 STEP1 中的 A，B)，則對此集合內的元素進行**定理 20 及 21 的錯排運算**，以加速滿足運算，找出是否為網路爬蟲所要進行的特定活動。

(b) 若為空集合(如 STEP1 中的 C)，則不做處置。

我們以圖9為例：

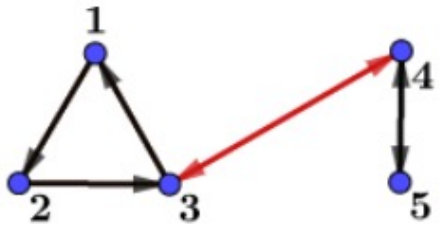


圖9 廣度優先分散式網路爬蟲改良演算法遍歷模擬

結論與應用

- 由於網站常是分層進行設計的，由上而下分別是最上層的是高級功能變數名稱，之後是子功能變數名稱，子功能變數名稱下又有子子功能變數名稱等等。因此，每個子功能變數名稱可能還會擁有多個同級功能變數名稱，而且 URL 之間有相互連結，由此構成一個複雜的網路。由於網路爬蟲有所謂的廣度優先演算法。

- 廣度優先搜尋法，本質是一種圖形(graph)搜索演算法。從圖的某一節點(vertex 或 node)開始走訪，接著走訪此一節點所有相鄰且未拜訪過的節點，由走訪過的節點繼續進行先廣後深的搜尋，它更是圖遍歷問題的一種 NP-hard。以樹(tree)來說即把同一深度(level)的節點走訪完，再繼續向下一個深度搜尋，直到找到目的節點或遍尋全部節點。

結論與應用

- 利用佇列(Queue)來處理，通常以迴圈的方式進行暴力法呈現。而此分 析方式恰好是，做網路爬蟲過程中，可以推測深度優先演算法本質上是具遞迴性的方式的

- 本研究提出了定理，可有效改善深度優先演算法網路爬蟲瀏覽時間及提升資料覆蓋率之最佳化結果，並透過$D_{k,kn}$ ，$F_k(K_n,j)$參數及遞迴關係來達到廣度優先分散式網路爬蟲改良演算法的網路爬蟲搜索方向。由於網路爬蟲是多面向的工具方法組成的資訊學科議題，因此若只關注某一方式的分析話題或利用關聯分析法演算，可能無法更有效地提升原有方法的效能。由於網路中每一個網站皆可視為一個大集合中的若干子集，每個網站中的資料又可視為子集合內的元素。因此有必要利用各種組合數學的方式來對其進行演算改善，避免網路爬蟲陷入NP-hard 問題的局部最佳化的分析盲點。

結論

- 這篇文章在討論分散網路爬蟲搜尋網址問題，它的本質是遍歷完所有的頂點且沒有重複經過，即所謂哈密頓路徑問題，是NP-hard 問題。這篇論文由 k-錯排遞迴之性質來探討分散式網路爬蟲最佳化問題，最後透過電腦模擬及組合數學分析推導。
- 本研究結果相當豐富，作者在各式各樣的條件下寫下遞迴式，研究錯排個數及循環組數的特殊情形。然而其計算效率仍須進一步電腦實證。

研究主題：

超立方體最小控制集建構方式的探討

研究動機

- 過去對超立方體最小控制集的研究，1990年文獻證明：
  - $\forall$ k$\in\mathbb{N}$，n = 2^k-1，有$\gamma(Q_n)= 2^{n-k}$
- 1998年文獻證明：
  - $\gamma(Q_5)=7$，$\gamma(Q_6)=12$
- 2008年文獻則提出$\gamma(Q_n)$的猜想：
  - $\forall$ $2^{p-1}-1 < n < 2^p-1, \gamma(Q_n) = 2^{n-p+1} - [2^{2n-p-2^p+1}$

研究動機

- 起初希望透過研究控制點的排列方式，觀察是否對證明此猜想有所幫助，而在撰寫程式得出超立方體Q_5中的每一種最小控制集排列時，發現每一種解之間似乎均同構(isomorphic)，且Q_6的最小控制集中似乎也有相同規律。因此我們希望能在知道$\gamma(Q_n)$的各個超立方體中找出其一般化的建構模式，並探討是否存在著同構的現象。

研究目的

- 探討Q_1、Q_2、Q_3、Q_4最小控制集的建構模式與同構現象。
- 提出$\gamma(Q_5)$=7更簡單的證明，並探討Q_5最小控制集的建構模式與同構現象。
- 提出$\gamma(Q_6)$=12更簡單的證明，並探討Q_6最小控制集的建構模式與同構現象。
- 證明$\gamma(Q_8)$=32，並探討Q_7、Q_8 最小控制集的建構模式與同構現象。

研究主题：

正三角形的最小拼接

研究動機

- 「給邊長分別 為 7 、5、3 的三種正三角形，如何使用這三種正三角形拼出另一正三角形，使其邊長達到最 小？」對此問題甚感興趣，除了找出此問題的解外，也嘗試把「 7 、5、3 」轉換成「 a 、b 、c 三數兩兩互質」作為研究題目。

研究目的

- 1.使用兩種不同邊長的正三角形拼出另一正三角形，使其邊長達到最小。
- 2.使用三種邊長兩兩互質的正三角形，利用特定拼法拼出另一正三角形，使其邊長達到 最小。
- 3.使用三種邊長互質的正三角形，利用特定拼法拼出另一正三角形，使其邊長達到最 小。
- 4.使用三種任意正整數邊長的正三角形，利用特定拼法拼出另一正三角形，使其邊長達 到最小。

研究主題：

渾「圓」有「定」— 從七圓定理到雙心六圓的性質探討與推廣

研究動機

- 在閱讀有關反演變換的文獻時，意外發現一個有趣、形似湯圓的圖形：在一個大碗中放入六顆湯圓。兩兩均內切的六個小圓，若其兩相鄰均外切，則其對應內切點連線共點，這就是「七圓定理」；雖名為「七圓」，關鍵卻在於其中的「六圓」，如下圖。

研究動機

- 文獻上除了以不同樣貌出現外，如何尺規作圖？除了諸線共點的性質外，我們亦發現諸點共線、諸點共圓等特性。如果不是六個小圓而是更少圓，甚至更多圓呢？又或同時內外切於兩個內離圓，甚至同時外切於兩個外離圓的情形呢？於是展開本研究。

研究目的

本研究試圖從七圓定理的性質探討與推廣，進而研究雙心六圓，以至多圓時的作圖關係式及共點、共線、共圓、共錐的不變性與對偶性探討，研究問題如下：

- 探討七圓定理的性質與多圓的推廣。
- 探討雙心六圓的作圖關係式與性質。
- 以雙心六圓的結果探討雙心多圓的性質。
- 試研究雙心多圓在兩外離圓及圓與直線的性質。
- 七圓定理及雙心多圓在球面及球體上的推廣。

研究主題：

正 n 邊形內接正四邊形之探討

研究動機

- 在閱讀第54屆全國中小學科學展覽歷屆作品時，看到在一個正n邊形的三個不同邊上可以內接無限多個正三角形，因此好奇：

是否在正n邊形內也都能接出正四邊形？

是否也有無限多個內接正四邊形？因此開始進行研究和探討。

研究目的

- 首先利用電腦繪圖觀察是否所有的正n邊形都存在內接正四邊形，再嘗試以數學式證明之。
- 內接正四邊形有無限多個嗎？
- 內接正四邊形是否和正n邊形有關聯性?
- 找一個尺規作圖法畫出所有正n邊形的內接正四邊形。

研究主題:

二元3平衡n字串之排列數探討

研究動機

「由 0, 1 排成長 度 n 的字串，稱為二元 n 字串。若一個二元 n 字串中出現字串 00 和字串 11 的個數一樣多，則稱為長度 n 的二元平衡字串。若以 an 表示長度 n 的二元平衡字串之個數，已知 $a1 = a2 = a3 = 2$, $a4 = 4$, $a5 = 6$，試求 an 的一般公式。」解法是特例解，只有在 00- 子字串, 11-子字串平衡的狀況下才適用，因此便想要改變作法，試圖用一個一般化的解法來處理這個問題。此外，也想將此問題拓展，討論在 000-子字串,111-子字串平衡時， n 位數列排列之符合個數是否也有一般解，也就是二元 3 平衡 n 字串個數是否有一般式。

研究目的

- 1.原題目之非特例解。
- 2.二元 3 平衡 n 字串之關係式探討。
- 3.二元 r 平衡 n 字串在滿足字串中無連續 r 個 0 或連續 r 個 1 時之個數遞迴式探討。
- 4.二元 3 非平衡 n 字串之關係式探討。
- 5.觀察二元 3 非平衡 n 字串在改變 000-子字串及 111-子字串之差值或字串之長度時，符合個數彼此間有何性質存在？

研究目的

- 6. 階差數列中各階階差首項值之求解過程與性質。
- 7. 將第五點推廣至二元 r 非平衡 n 字串。
- 8. 二元平衡 n 字串、二元 3 平衡 n 字串 $\lim_{n\to\infty}\frac{S_{(n,i,0)}}{S_{(n-1,i,0)}}(i=2,3)$之探討，其中$S(n,i,0)$代表 長度為$n$的字串中滿足$i$-0-子字串(連續$i$個 0 所形成的子字串)與$i$-1-子字串(連續$i$個 1 所形成的子字串)數目相同的個數。

分工

改良式廣度優先網路爬蟲演算法之組合分析:蔡孟勳
超立方體最小控制集建構方式的探討、渾「圓」有「定」—從七圓定理到雙心六圓的性質探討與推廣、正n邊形內接正四邊形之探討 :陳怡傑
正三角形的最小拼接、二元3平衡n字串之排列數探討、簡報整理:黃擎天
報告:陳仕銘、李柏勛

謝謝