2022 年臺灣國際科學展覽會
優勝作品專輯

作品編號　**010023**

參展科別　數學

作品名稱　等差指標的探討

得獎獎項

就讀學校　臺北市立成功高級中學

指導教師　許為明、沈俊嚴

作者姓名　林子惟、陳冠彣、詹詠翔

關鍵詞　等差指標、等差數列、史丹利數列

作者簡介

我是林子惟，就讀成功高中數理資優班。自國中開始自學數學和專題研究至今，喜歡參加長期的數學解題活動，曾參加過通訊解題、雙週一題，也會自己找網路上的題目研究，和自學高等微積分。

我是陳冠彣，目前就讀成功高中數理資優班二年級。從小最有興趣也最拿手的科目就是數學了，自從上了高中，我參與了許多大大小小的科展及競賽，希望也能在這次的比賽中得到名次。

我是詹詠翔。目前就讀成功高中數理資優班二年級。今年 17 歲。我對數學和資訊非常有興趣，數學常可以透過電腦程式的窮舉來找出規律，甚至算出正確答案。我發現此次研究與撰寫資訊奧林匹亞的題目的原理類似，因此我有了莫大的興趣。我想透過我所擁有的專長，來將這份苦差事交給電腦處理，加速我們研究的進行。

摘要

在國中及高一的數學課程當中，我們都學過「等差數列」這個單元，而本文的研究著重於在某些限制條件下等差數列的產生時機。在本篇報告中，我們從給定整數範圍(如：從0到n)的條件下出發，去探索在給定的範圍當中至少要加進去多少正整數才能確保其中存在三個數字成等差，進而定義等差指標等概念。我們研究了各個數字n的等差指標，並找出其關係，進而延伸出相關的不等式，然後進行推論。最後，我們藉由發現了一些規律得到了一些可增進最小等差指標的估計，我們大略估計了其上下界，並嘗試往探討不存在三正整數成等差數列的自然數集合密度去做發展，用集合密度的角度去切入討論，以幫助我們的估計和定理推導。

Abstract

Problems about arithmetic progressions have played a significant role in arithmetic combinatorics, which are our major research topic. We have been trying to discover the specific situation when a 3-term arithmetic progression exist in a set of integers without using difficult and complicated analysis skills.

We started from giving it a range, for example, from 0 to n, n has to be an integer, and we try to figure out how many integers should we add in the given range to make sure it must contain a 3-term arithmetic progression, and then we defined the concept of our major researching topic – arithmetic progression index.

For small n, We've found some arithmetic progression indexes to seek relation between n and the index and devise some inequalities. In the end of our research, we did some estimates of Arithmetic Progression Indexes for n large enough. Now we're doing research in the set of integer which doesn't contain any 3-term arithmetic progression, which would help us with the better estimates and more conjectures.

壹、 研究動機

我們國中時都學過「等差數列」，不過以前的所學通常是給定既有的數列去判斷是否為等差或給定一個等差數列讓我們去求其他的問題，反而很少讓我們去考慮一些開放性的問題。所以，本文想試著去討論一些不一樣的東西，如：反過來探討在某些限制條件或範圍下，是否至少能找到三個整數會成等差數列。由於**等差數列平移之後仍然是等差數列**，所以我們可以將討論範圍從**一般的整數範圍**限縮至區間**[0, $\boldsymbol{n}$]** (n為任意正整數)。最後，我們終於在所有想得到的問題當中，找到了一個研究目標：定義**等差指標**，並致力於尋求其**公式**或相關的**最佳估計**。

貳、 研究目的

<u>**研究問題**</u>：對於任意正整數n，我們希望就其等差指標$A_1(n)$之值作探討並給予估計。

針對我們的問題，我們有以下的幾個目標:

1.先從解決簡單的問題著手，整數範圍小的情形開始討論，如：0~2、0~3，0~4 等。從中找尋數據，並逐漸加大範圍，以找出規律。

2.試圖去證明或發現規律或性質。

3.若無法直接找出結論，至少找出接近的估計值或可供參考的不等式，再深入探討並尋找問題的應用性及尋求問題的推廣。

4.尋找其他類似的研究問題。

5.藉由觀察等差集合差集的加法結構

參、 研究設備及器材

筆電、紙、筆

肆、 研究過程或方法

首先，我們先給一些定義如下：

<u>**定義1**</u> 我們定義**等差指標**$A_1(n)$、$A_2(n)$、$A_3(n)$、$A_4(n)$如下:

- $\boldsymbol{A_1(n)}$為最小的 s 使得在0與n之間任意插入 s 個正整數a_1 、a_2 ……、a_s後，0、a_1 、a_2 ……、a_s、 n中必定存在三項成等差數列 。定義$A_1(1)=0$。

- $\boldsymbol{A_2(n)}$為最大的 s 使得在0與n之間可以插入 s 個正整數a_1 、a_2 ……、a_s，可以滿足0、 a_1 、a_2 ……、a_s、 n中不存在三項成等差數列 。顯而易見，$A_1(n)=A_2(n)+1$。

- $\boldsymbol{A_3(n)}$為最大的 s 使得在1與n之間可以插入 s 個正整數a_1 、a_2 ……、a_s，可以滿足1、 a_1 、a_2 ……、a_s、 n中不存在三項成等差數列 。

- $\boldsymbol{A_4(n)}$為最大的 s 使得在1與n之間可以插入 s 個正整數a_1 、a_2 ……、a_s，可以滿足 a_1 、a_2 ……、a_s中不存在三項成等差數列 。

  [註0-1] **等差指標**為判斷是否會形成等差數列的重要依據。

  [註0-2] 當n很大時，四個等差指標所得出的值的差距可忽略，故他們在n很大時可視為相同。

<u>**定義2**</u>:定義集合 $A+B:=\{a_i+b_j|a_i\in A,b_j\in B\}$，$A-B:=\{a_i-b_j|a_i\in A,b_j\in B\}$。

<u>**定義3**</u>:令S_n為收集1~n之間的整數集合，且滿足:

(1) S_n當中任三元素不會形成長度為3之等差數列的集合。

(2) $\boldsymbol{S_n}$**具飽和性**，即若在S_n中再放入任意一個正整數必至少產生一個長度為3等差數列。

(一)、數據與預測：

在尋找$A_1(n)$的過程中，首先考慮讓0與n當中插入「**最多**」個數字使得這些數字(也包含0~n)當中**任三個數字皆不成等差**，事實上所找到的「**最多**」個數字即為「$A_2(n)$」個數字。**我們將此取法取得的數列記為$\{T_k^{(n)}\}$，其中 k 表項數**。

1. 以最小的數字2為例

較小的數字優先考慮，不能插1進去，否則0、1、2即會成為一等差數列，故$A_2(2)$=0，即 $A_1(2)$=1。

2. 以一個稍微大一點的數字16為例

和找$A_2(2)$的方法相同，取0、1、3、4、9、10、11、12、16，扣掉首末項之後是7個數字，故$A_1(16)$=8。做法如下，取1，不取2 (0、1、2成等差)，接著3、4，不取5，不取6 (0、3、6成等差)，不取7 (1、4、7成等差)，不取8 (0、4、8成等差或0、8、和末項16成等差)，取9、10，不取11 (9、10、11成等差)，取12，不取13 (10、13、末項16成等差)，不取14 (10、12、14成等差)，不取15(9、12、15成等差)。這樣的觀察法雖然有點費時，卻可較為容易地找出$A_2(n)$的數值。

3. 在討論過程中找到的規律

在尋找$A_2(n)$的過程中，我們將$n=2$到$n=50$分別所取的數列$\{T_k^{(n)}\}$列出如第7頁中的**表1**，其中選取這些數字的目的在於讓**任三個數字不成等差**且**個數最多**，故其個數即為$A_2(n)$。結果，我們發現所取的n值越來越大時，其等差指標數列$\{T_k^{(n)}\}$與其他小於n的數值所對應的等差指標數列之前幾項會不間斷出現越來越多重複的數字。

綜上所述，我們猜測重複的數字和n的大小有關，於是我們開始考慮不受末項的

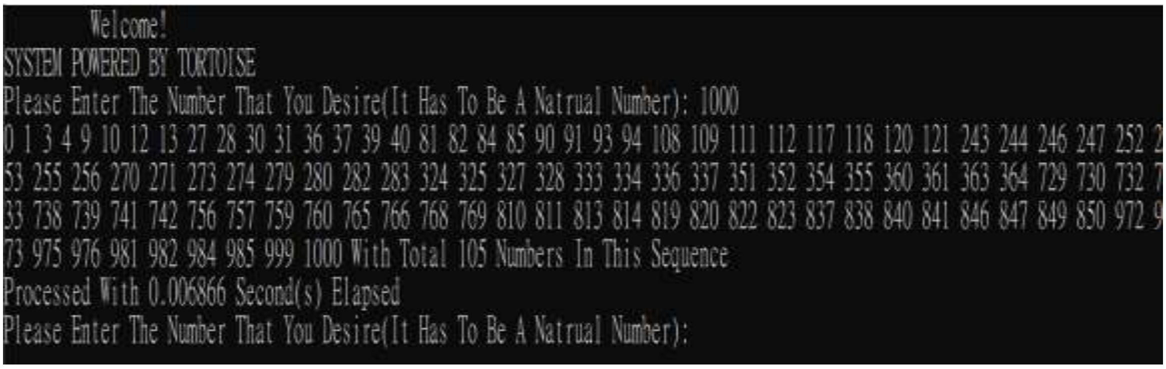
限制下(即假設n無窮大)去建構一個數列同樣滿足插入「最多」個數字使得這些數字(包含0)當中任三個數字皆不成等差，我們暫且將這樣的數列用$\{T_k^{(n)}\}$表示。接著，我們嘗試找出$\{T_k^{(n)}\}$的前幾項為1、3、4、9、10、12、13、27、28、30、31……，剛好這些數字皆為當n越來越大時，其等差指標數列$\{T_k^{(n)}\}$與其他小於n的數值所對應的等差指標數列之前幾項可能出現的不間斷重複數字。

接下來，我們觀察當$\{T_k^{(\infty)}\}$的某個數字 **k 開始不間斷出現在$\{T_k^{(n)}\}$時，該n必大於 2k+1，且標配之間符合0、1(2*0+1)、3(2*1+1)、4、9(2*4+1)、10、12、13、27（2*13+1）、28、30、31、36、37、39、40（2*13+1+13)、81（2*40+1）、82、…，依此規律應可推得$\{T_k^{(\infty)}\}$所有的項。**

此外，我們可以發現$\{T_k^{(\infty)}\}$中有許多連續兩項的數字為n與$2n$+1的關係，我們將其稱為節點，其中n稱為節點前項，$2n$+1稱為節點後項。

為了方便觀察數據，我們用培基(BASIC)語言寫了一個程式，只要輸入末項n，系統就會自動排出由$\{T_k^{(n)}\}$當中與$\{T_k^{(\infty)}\}$相同的數字，並算出其個數。例如： 如**圖1**所示，欲知0到1000內的狀況，僅需輸入1000，電腦即會列出$\{T_k^{(1000)}\}$當中與$\{T_k^{(\infty)}\}$相同的數字，並算出其個數。

圖1：培基(BASIC)語言計算$\{T_k^{(1000)}\}$當中與$\{T_k^{(\infty)}\}$相同的數字及其個數


```
        Welcome!
SYSTEM POWERED BY TORTOISE
Please Enter The Number That You Desire(It Has To Be A Natrual Number): 1000
0 1 3 4 9 10 12 13 27 28 30 31 36 37 39 40 81 82 84 85 90 91 93 94 108 109 111 112 117 118 120 121 243 244 246 247 252 2
53 255 256 270 271 273 274 279 280 282 283 324 325 327 328 333 334 336 337 351 352 354 355 360 361 363 364 729 730 732 7
33 738 739 741 742 756 757 759 760 765 766 768 769 810 811 813 814 819 820 822 823 837 838 840 841 846 847 849 850 972 9
73 975 976 981 982 984 985 999 1000 With Total 105 Numbers In This Sequence
Processed With 0.006866 Second(s) Elapsed
Please Enter The Number That You Desire(It Has To Be A Natrual Number):
```

然而，我們發現在該程式中，將n輸入27時，程式跑出的數字(即$\{T_k^{(27)}\}$當中與$\{T_k^{(\infty)}\}$相同的數字個數)比n輸入26及28所得到的數字(即$\{T_k^{(26)}\}$當中與$\{T_k^{(\infty)}\}$相同的數字個數以及$\{T_k^{(28)}\}$當中與$\{T_k^{(\infty)}\}$相同的數字個數)還要小，明顯不合常理，不符合$A_1(n)$為遞增函數的敘述(後面會證明)，陸續輸入43、47、48、49和50皆發生相同情況。

經過討論之後，我們認為這個取法是錯誤的，這個程式的問題就是一決定數字時，便永遠不會更動，導致末項可以影響中間的數字，無法考慮其塞入數字遞增的合理性。然而數字足夠大或不考慮末項一直取下去，前面部分即可避免這個問題，故$\{T_k^{(\infty)}\}$可以保證正確。

我們發現原本的取法要從最小的數字開始考慮，於是我們的取法中必定取1，必不取2。在我們無法確定這個取法的完全正確性之前，我們姑且只能稱他是一個正確率非常高的方法，可以當作參考卻不能完全相信。

表1：$n=2$到$n=50$所取的數列$\{T_k^{(n)}\}$

註:黃色螢光部分為此取法之不符合者，若有刪除記號者表示該列數字取錯，若無則代表已找到其符合之$\{T_k^{(n)}\}$了。本表主要是為了觀察各$\{T_k^{(n)}\}$的結構趨勢

n	$\{T_k^{(n)}\}$
2	0、2
3	0、1、3
4	0、1、3、4
5	0、1、4、5
6	0、1、4、6

7	0、1、3、7
8	0、1、3、7、8
9	0、1、3、4、9
10	0、1、3、4、9、10
11	0、1、3、4、9、11
12	0、1、3、4、9、10、12
13	0、1、3、4、9、10、12、13
14	0、1、3、4、9、10、13、14
15	0、1、3、4、10、11、14、15
16	0、1、3、4、9、11、12、16
17	0、1、3、4、11、12、15、17
18	0、1、3、4、10、12、13、18
19	0、1、3、4、9、12、13、19
20	0、1、3、4、9、11、16、20
21	0、1、3、4、9、10、13、21
22	0、1、3、4、9、10、12、22
23	0、1、3、4、9、10、21、23
24	0、1、3、4、9、10、13、21、24
25	0、1、3、4、9、10、12、22、25
26	0、1、3、4、9、10、12、22、25、26
27	~~0、1、3、7、9、12、13、16、22、27~~
28	0、1、3、4、9、10、12、13、27、28
29	0、1、3、4、9、10、12、13、27、29
30	0、1、3、4、9、10、12、13、27、28、30
31	0、1、3、4、9、10、12、13、27、28、30、31
32	0、1、3、4、9、10、12、13、27、28、31、32

33	0、1、3、4、9、10、12、13、27、28、31、33
34	0、1、3、4、9、10、12、13、27、28、30、34
35	0、1、3、4、9、10、12、13、27、28、30、34、35
36	0、1、3、4、9、10、12、13、27、28、30、31、36
37	0、1、3、4、9、10、12、13、27、28、30、31、36、37
38	0、1、3、4、9、10、12、13、27、28、30、31、36、38
39	0、1、3、4、9、10、12、13、27、28、30、31、36、37、39
40	0、1、3、4、9、10、12、13、27、28、30、31、36、37、39、40
41	0、1、3、4、9、10、12、13、28、29、31、32、37、38、40、41
42	0、1、3、4、9、10、12、13、28、29、31、32、38、39、41、42
43	~~0、1、3、4、9、10、12、13、27、29、30、32、38、39、43~~
44	0、1、3、4、9、10、12、13、29、30、32、33、38、39、42、44
45	0、1、3、4、9、10、12、13、28、30、31、33、37、40、42、45
46	0、1、3、4、9、10、12、13、27、30、31、34、36、39、43、46
47	~~0、1、3、4、9、10、12、13、27、32、33、36、38、46、47~~
48	~~0、1、3、4、9、10、12、13、27、28、31、32、39、43、48~~
49	~~0、1、3、4、9、10、12、13、27、28、30、34、35、37、49(?)~~
50	~~0、1、3、4、9、10、12、13、28、29、32、33、40、42、50(?)~~
∞	0、1、3、4、9、10、12、13、27、28、30、31、36、37、39、40、81…

表2：n ＝2到n ＝50所對應的$A_1(n)$

$A_1(n)$的值						
$A_1(2)=1$	$A_1(9)=4$	$A_1(16)=7$	$A_1(23)=7$	$A_1(30)=10$	$A_1(37)=13$	$A_1(44)=15$
$A_1(3)=2$	$A_1(10)=5$	$A_1(17)=7$	$A_1(24)=7$	$A_1(31)=11$	$A_1(38)=13$	$A_1(45)=15$

$A_1(4)=3$	$A_1(11)=5$	$A_1(18)=7$	$A_1(25)=7$	$A_1(32)=11$	$A_1(39)=14$	$A_1(46)=15$
$A_1(5)=3$	$A_1(12)=6$	$A_1(19)=7$	$A_1(26)=7$	$A_1(33)=11$	$A_1(40)=14$	$A_1(47)=15$
$A_1(6)=3$	$A_1(13)=7$	$A_1(20)=7$	$A_1(27)=8$	$A_1(34)=11$	$A_1(41)=15$	$A_1(48)=15$
$A_1(7)=3$	$A_1(14)=7$	$A_1(21)=7$	$A_1(28)=9$	$A_1(35)=12$	$A_1(42)=15$	$A_1(49)=15$
$A_1(8)=4$	$A_1(15)=7$	$A_1(22)=7$	$A_1(29)=9$	$A_1(36)=12$	$A_1(43)=15$	$A_1(50)=15$

此外，觀察上述表格可發現一些性質或估計，我們將在接下來的單元中作討論。

(二)、關於等差指標的不等式

在這裡，我們將對已發現的不等式作敘述並證明。

定理一

對任意的正整數n，以下的不等式成立

$$A_4(n) \leq A_1(n+1)$$

【**證明**】假設$A_4(n)=s$，則在1與n之間可以找到 s 個正整數$a_1, a_2, \dots, a_s$，使得其中不存在三項成等差級數。因為這些數也在1與n+1之間，所以根據定義就有$A_4(n+1) \geq s$，遂有$A_4(n) \leq A_4(n+1)$而得證。

定理二

對任意正整數n，令$A_1(n)=d$，給定任意整數 m。若在 m 到$m+n$之中插入任意 d 個由小到大的正整數，則這些數字之中(包含 m 與$m+n$)必存在三數成等差。若在 m 到$m+n$之中插入任意 d−1個由小到大的整數，則這些數字之中(包含 m 與$m+n$)必不存在三數成等差。

【**證明**】任取在 m 到$m+n$之間任意插入 d 個正整數$s_1, a_2, \dots, a_d$，並依序排列如下：

$$m, a_1, a_2, \dots, a_d, m+n。$$

我們將上述數列中的每一項同時減去 m，平移為

$$0, a_1-m, a_2-m, \dots, a_d-m, n。$$

由於等差數列平移後仍為等差數列，再由等差指標的定義可得證。

<u>定理三</u>

對任意正整數n，以下的不等式成立

$$\mathbf{A_1(2n) \le 2A_1(n) + 1}$$

【證明】令$A_1(n)$=d，在0到$2n$間插入2d+1個正整數$a_1, a_2, \dots, a_{2d+1}$並依序排列如下：

$$0, a_1, a_2, \dots, a_{2d+1}, 2n$$

<u>Case 1</u>：數列$a_1, a_2, \dots, a_{2d+1}$中包含$n$

若$a_i = n$, 其中$1 \le i \le d+1$，則在數列的n到$2n$之間，存在大於或等於 d 個整數，而其必大於或等於n的等差指標，故在此條件下，不等式成立。

若$a_j = n$, 其中$d+2 \le j \le 2d+1$，則在數列的0到n之間，存在大於或等於 d 個整數，而其必大於或等於n的等差指標，故在此條件下，不等式成立。

<u>Case 2</u>：數列$a_1, a_2, \dots, a_{2d+1}$中不包含$n$

若$a_{d+1} > n$，取$2n - a_{d+1} = p$，則 p< n，可推得A_1(p)$\le A_1(n)$= d，故 d 必大於或等於 p 的等差指標，在此條件下，不等式成立。

若$a_{d+1} < n$，取$n - a_{d+1} = q$，則 q< n，可推得A_1(q)$\le A_1(n)$= d，故 d 必大於或等於 q 的等差指標，在此條件下，不等式成立。

<u>定理四</u>

對任意的正整數 m，n (m $\neq n$)，以下的不等式成立:

$$\mathbf{A_1(m) + A_1(\boldsymbol{n}) + 1 \geq A_1(\boldsymbol{m}+\boldsymbol{n})}$$

【證明】 假設A_1(m +n) = s，則在1與 m +n之間可以找到 s 個正整數$a_1, a_2, \ldots, a_s$使得其中不存在三項成等差級數。不妨假設 s = u+v，而$a_1, a_2, \ldots, a_u$在1與 m 之間，同時$a_{u+1}, a_{u+2}, \ldots, a_{u+v}$在 m+1與 m+$n$之間。此時，因為$a_1, a_2, \ldots, a_u$在1與 m 之間而且任三項不成等差數列，所以就有$A_1$(m) ≥ u。同時，$a_{u+1} - m, a_{u+2} - m, \ldots, a_{u+v} - m$在1與$n$之間而且任三項不成等差數列，所以就有$A_1$($n$) ≥ v。綜合得到$A_1$(m) +$A_1$($n$) ≥ u + v = s = A_1(n+1) 。

事實上，**<u>定理四</u>**中的不等式不一定只能置入兩個元素，在多個元素的條件下亦成立，由原先二元的不等式推得以下多元的不等式。

<u>推論一</u>

給定正整數n及相異正整數$a_1, a_2, \ldots, a_k$，以下的不等式成立:

$$\sum_{k=1}^{n} \boldsymbol{A_1(a_k)} + (\boldsymbol{n}-\mathbf{1}) \geq \boldsymbol{A_1}\left(\sum_{k=1}^{n} \boldsymbol{a_k}\right)$$

【證明】

$$\boldsymbol{A_1}\left(\sum_{k=1}^{n} \boldsymbol{a_k}\right)$$

$= A_1(a_1 + a_2 + \cdots + a_k)$

$= A_1[a_1 + (a_2 + \cdots + a_k)]$

$\leq A_1(a_1) + A_1(a_2 + \cdots + a_k) + 1$

$\leq A_1(a_1) + [A_1(a_2) + A_1(a_3 + \cdots + a_k) + 1] + 1$

$\leq \ldots$

$$\leq \sum_{k=1}^{n} \boldsymbol{A_1(a_k)} + (\boldsymbol{n}-\mathbf{1})$$

事實上，由**<u>推論一</u>**可以將**<u>定理三</u>**中的不等式推廣如下：

<u>推論二</u>

對任意大於1的正整數n及對任意的正整數 k，以下的不等式成立:

$$\mathbf{A_1(k\boldsymbol{n}) \leq kA_1(\boldsymbol{n}) + k\text{-}1}。$$

【證明】 對任意大於1的正整數n及對任意的正整數 k，

$A_1(\mathrm{k}n) = A_1((\mathrm{k}\text{-}1)n{+}n)$

$\leq A_1((\mathrm{k}\text{-}1)n) + A_1(n) + 1$

$\leq A_1((\mathrm{k}\text{-}2)n) + (A_1(n){+}\ 1) + A_1(n) + 1$

$= A_1((\mathrm{k}\text{-}2)n) + 2A_1(n) + 2$

$\leq A_1((\mathrm{k}\text{-}3)n) + 3A_1(n) + 3$

$\leq \ldots$

$\leq A_1(n) + (\mathrm{k}\text{-}1)A_1(n) + (\mathrm{k}\text{-}1)$

，得證。

<u>推論三</u>

對任意的正整數$\boldsymbol{n}$，以下的不等式成立

$$\mathbf{A_1(\boldsymbol{n}){+}1 \geq A_1(\boldsymbol{n}{+}1) \geq A_1(\boldsymbol{n})}$$

【證明】 根據**<u>定理四</u>**，對於任意正整數n，$A_1(1) + A_1(n) + 1 \geq A_1(n{+}1)$

先前已定義$A_1(1)$=0，故可和**<u>定理一</u>**合併，變成$A_1(n) + 1 \geq A_1(n{+}1) \geq A_1(n)$。

由**<u>推論三</u>**易知等差指標$A_1(n)$之間的值前後項最多差1，而此不等式也在前

面**表1**的數據中顯現出來。此外，配合**表1**的數據可以很清楚知道**推論三**所提供的不等式無法再更進一步精簡，所以這已是**最佳型式**。

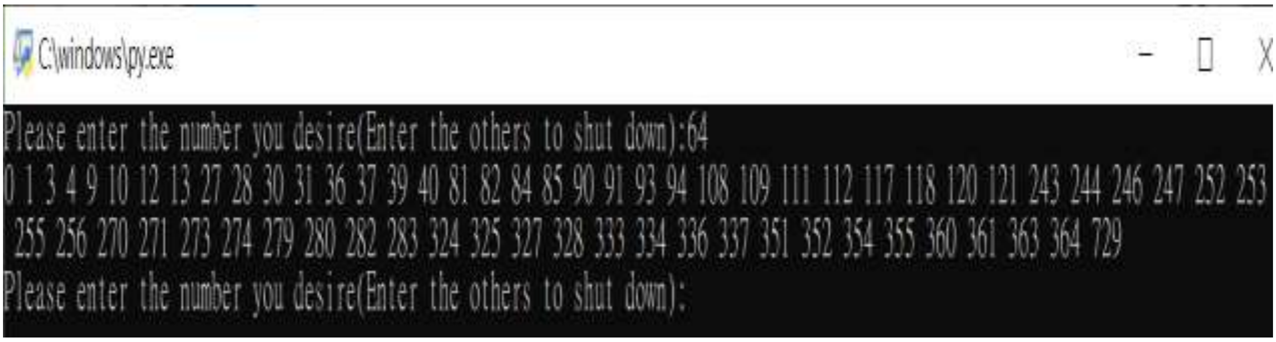
(三)、等差指標的估計

1. $\{T_k^{(\infty)}\}$的相關性質

利用先前的程式去跑$\{T_k^{(\infty)}\}$的前幾項，如下面的**圖2**所示，$\{T_k^{(\infty)}\}$的第0項到第64項為:

0 1 3 4 9 10 12 13 27 28 30 31 36 37 39 40 81 82 84 85 90 91 93 94 108 109 111 112 117 118 120 121 243 244 246 247 252 253 255 256 270 271 273 274 279 280 282 283 324 325 327 328 333 334 336 337 351 352 354 355 360 361 363 364 729

圖2：培基(BASIC)語言計算數列$\{T_k^{(\infty)}\}$中的前64項



經觀察後發現以下特性：

(1).節點前項為必發生於$T_{2k-1}^{(\infty)}$的形式，其中 k 為任意非負整數

(2).節點後項為3^{k}的形式

(3).相鄰兩項的差為1,2或數個相異的3的 k 次方和再加上2，依序排列後1和其他數交錯排。

接下來，我們好奇這樣的數列是否為我們已知的數列，故我們將$\{T_k^{(\infty)}\}$的前幾項輸入整數數列線上大全(OEIS)，發現$\{T_k^{(\infty)}\}$事實上為<u>**史坦利**</u>**(Stanley)數列**，根據上面所述，這樣的數列可被一般表示為**在三進位之中不含有2的非負整數數列**，換

句話說，將$\{T_k^{(\infty)}\}$轉換為三進位時會得到一串和二進位數列一樣的數列，故以二進位看的話，看到的是各項的順序(自0開始)，和我們的觀察相符，舉例說明如下：

*以數列$\{T_k^{(\infty)}\}$中的第8項27為例:

將27轉為3進位後為$(1000)_3$，將1000看成2進位後，再將$(1000)_2$轉回10進位得

$$1\cdot 2^3+0\cdot 2^2+0\cdot 2^1+0\cdot 2^0 = 8，$$

故27為數列$\{T_k^{(\infty)}\}$中的第8項。

*以數列$\{T_k^{(\infty)}\}$中的第31項355為例:

將355轉為3進位後為$(11111)_3$，11111看成2進位後，再將$(11111)_2$轉回十進位得

$$1\cdot 2^4+1\cdot 2^3+1\cdot 2^2+1\cdot 2^1+1\cdot 2^0 = 31，$$

故355為數列$\{T_k^{(\infty)}\}$中的第31項。

2. $A_1(n)$的上界估計與下界估計

藉由$\{T_k^{(\infty)}\}$即為史坦利數列的事實，我們接下來開始尋找$A_1(n)$的上下界。

定理五(等差指標的上界與下界估計)

正整數 k 使得$2\cdot 3^{\mathrm{k}-1}+1 < 2n+1 \leq 3^{\mathrm{k}}$時，則$2^{\mathrm{k}-1} < A_1(n) \leq 2^{\mathrm{k}}-1$

【證明】

以每個$3^{\mathrm{k}-1}$到3^{k}當作一個區間，當$3^{\mathrm{k}-1} < n$ 時，$A_1(n)$會大於$3^{\mathrm{k}-1}$的項數，也就是 $2^{\mathrm{k}-1}$ ∵$\{T_k^{(\infty)}\}$不存在等差數列的性質和$A_1(n)$必須要達成「任取」

上界和下界有些不同，3^{k}的前一項是$\frac{3^{\mathrm{k}}-1}{2}$，故當 $2n+1 \leq 3^{\mathrm{k}}$時，則$A_1(n) \leq 2^{\mathrm{k}}-1$ ∵條件為大於等於並且$A_1(n)$必須要達成「任取」

綜合以上，當存在正整數 k 可使得$2\cdot 3^{k-1}+1 < 2n+1 \le 3^{k}$時，則 $2^{k-1} < A_1(n) \le 2^{k}-1$成立。但是當 k 太大時，上下界的差也會跟著變得很大，導致範圍大到難以尋找，於是我們嘗試找出較為精確的上下界:

範例：取n=34，依照定理五試求出$A_1(34)$的範圍。

$$2\cdot 3^{3} \le 2\cdot 34+1 < 3^{4}$$

$$\rightarrow 2^{3} \le A_1(34) \le 2^{4}-1$$

(四)、精確下界:

找到n在$A_1(n)$的精確下界的步驟非常繁複：

(1).將n轉為三進位

(2).如果n轉換後數字包含2，將其第一個2以及其以後之數字全部轉為1，此數為n在$\{T_k^{(\infty)}\}$中，小於n且最接近的數，假設其項數為 k，故$A_1(n)$必定大於 k

∵$\{T_k^{(\infty)}\}$本身任意三項不會成等差數列，且$A_1(n)$為遞增函數。

(3).如果n轉換後數字沒有包含2，表示n是$\{T_k^{(\infty)}\}$的其中一項，而因為$A_1(n)$計算不會包含n，故$A_1(n)$會大於等於n在$\{T_k^{(\infty)}\}$的項數。

(4).例：

$34_{10} = 1021_3$

將第一個2以後數字全部轉為1 $1011_3 < 1021_3$

將轉換後數字視為二進位 $1011_2 = 11_{10}$

故$11 < A_1(34)$

以上步驟可針對$A_1(n)$粗略上下界範圍太大的情況使下界更靠近精確的$A_1(n)$。而我們推論標準上界不存在，因為無法確定粗略上界不等式的等號成立於何種情況，導致存在節點之間的n有可能突破所推論出的上界，所以我們推測標準的上界會有矛盾。

(五)、A_1和n的關係

根據史丹利數列中n對於 k 的上下界

$3^k \le n < 3^{k+1}$，且$2^{\mathrm{k}} \le A_1(n)$

藉由其上界$n < 3^{\mathrm{k}+1}$

$\Rightarrow \dfrac{\ln n}{\ln 3} < k+1 \quad \Rightarrow \dfrac{\ln n}{\ln 3} - 1 < k$

又當在n非常大的情況下

$\Rightarrow \frac{\ln n}{\ln 3} \le k$ ，再和 $2^{\mathrm{k}} \le A_1(n)$ 結合

$\Longrightarrow 2^{\frac{\ln n}{\ln 3}} \le A_1(n)$

(六)、利用集合方法來做估計

我們另一個研究方向為利用集合來去做加法結構之研究

我們先來做幾個基本的定義:

- 定義集合S_n為一個 1~n之整數集合且其任三元素不會形成等差數列，另外S_n必須具飽和性，該集合內元素數量需在條件下達到最大量，使得若在其集合中再插入任意一正整數必會使其產生一長度為 3 等差數列。

- 定義符號$S_n - S_n$為S_n之內各元素相減所形成之差集

  而$|S_n - S_n|$、$|S_n|$皆用來表示其個數

  其中$|S_n|$=$A_1(n)$+1

現在，定義清楚了過後，我們要來推估$|S_n - S_n|$的上下界

首先，利用一個集合學常用的定理:

$$|A+B| \ge |A| + |B| - 1$$

(其中，A、B 為正整數集合，等號成立的條件為 A 集合和 B 集合的元素本身會形成等差數列，在此不多做證明。)

將$\mathrm{A} = -\mathrm{B} = S_n$帶入上式，即可得下列式子

$$|S_n - S_n| \ge 2|S_n| - 1$$

當n很大的時候

$$|S_n - S_n| \geq 2|S_n|$$

又因曾在**利用集合來做估計**章節得出$|S_n|$，也就是$A_1(n)$的範圍，我們將其改寫

$$2^{\frac{ln\, n}{ln\, 3}} = n^{\frac{ln\, 2}{ln\, 3}} \leq A_1(n) = |S_n|$$

於是可和上式子結合

$$|S_n - S_n| \geq 2|S_n| \geq n^{\frac{ln\, 2}{ln\, 3}} \times 2 \approx 2 \times n^{0.65}$$

數學家 Bourgain 曾經研究過相關的定理，當初他是使用和集，而我們認為差集會更為簡單，相對於和集，差集多了一個對稱性，且可以觀察的出來$S_n - S_n$集合中必定存在長度為3等差數列(必存在n、0、-n)。我們的最終目標在於證明$S_n - S_n$含有任意長度為 k 的等差數列，對於所有 k 屬於正整數。

且根據我們的觀察，我們發現說：對於任意正整數n，必可找到在$\{T_k^{(\infty)}\}$的兩數 a, b 使得 a-b=n，不失一般性的假設n是正整數，考慮 a=n+b，將n轉換成3進位，各位有0, 1, 2三種。

b：在2的對應位置補上1使其和n相加時進位，而在因2和1相加進位導致下一位會變成2的位也補上1，其餘位置補上0，相加後即可得 a。

<u>**推論：**</u>設S為一集合包含所有史丹利數列的項，可知$S - S$包含所有的整數，顯然有五個以上元素可構成等差數列，而只要有集合元素構成和S相似(平移、倍數)，奇差集會有五項以上形成等差數列。

伍、 研究結果

定理:

$$A_1(n) \leq A_1(n+1)$$

$$A_1(2n) \leq 2A_1(n) + 1$$

$$A_1(m) + A_1(n) + 1 \geq A_1(m+n)$$

推論:

$$\sum_{k=1}^{n} A(a_k) + (n-1) \geq A\left(\sum_{k=1}^{n} a_k\right)$$

$$A_1(kn) \leq kA_1(n) + k - 1 \text{。}$$

$$A_1(n) + 1 \geq A_1(n+1) \geq A_1(n)$$

若存在正整數 k 使得$2 \cdot 3^{k-1} + 1 < 2n + 1 \leq 3^k$時，則$2^{k-1} < A_1(n) \leq 2^k - 1$

在n非常大時，$2^{\frac{\ln n}{\ln 3}} \leq A_1(n)$

定義一集合S_n，使其飽和，其元素數量為在不形成等差數列的前提下的最大數量，則其差集和自己本身有下列的關係:

$$|S_n - S_n| \geq 2|S_n| \geq \approx 2 \times n^{0.65}$$

陸、 討論

隨著數字增加，$A_1(n)$的同時增加，其估計所得的範圍變得更大，相對地誤差也會變大，愈來愈不易表示，討論到更後面，必須想辦法找到$A_1(n)$的通式，讓n越大時誤差率盡量降低。

我們的$A_1(n)$取法並沒有證明出來取法絕對正確，且$A_1(n)$的取法是非常隨機，且具有多種長相的。當數字大的時候可能會有非常多的錯誤，所以找尋$\{T_k^{(n)}\}$將不會成為我們未來的研究目標，只會拿來當成做為估計的工具。盡可能的利用規律或者是不等式的持續推導來幫助我們找到$A_1(n)$的最佳估計，才會是這個問題的核心做法。

於是我們找尋網路上許多跟此類問題相關的一些研究文獻，找到了類似像 Brehend construction 等等在研究等差數列的問題。但是實際代入數字後會發現，他們的研究皆是利用漸進的方式去找到其趨勢，並沒有辦法幫助我們在數字較小的時候達成較好的估計。於是我們正在想辦法由這個問題的本身，觀察出規律，並適時地帶入一些已研究過後的成果來完成估計。

我們目前著手於在研究在 *A tribute to Paul Erdös* 一書中所提到的 Bourgain 定理，其內容會對我們的研究非常有幫助，其利用和集的方式將不存在等差數列的集合帶入進一些集合學的定理，憑藉著等差數列其完整的加法結構來找到特性，進而解決了找出最佳估計之前必要的一個先決定理。我們學習他這種精神，利用較好做觀察和具有對稱性的差集來去做探討會相對容易得出相同的結果，於是未來將專注於研究這項定理。

柒、 結論

本文剛開始試圖找到$A_1(n)$對於所有n的絕對數值，發現其困難度過高，轉而研究其最佳估計，進而引入許多離散數學、數論、不等式、甚至組合來幫助我們找到最佳估計。我們對其做了初步的估計，並持續延伸此問題，且嘗試多方面的利用不同面向來解決這個問題。等差數列這個主題十分有趣，他具有非常完美的加法結構，常常會形成許多數學上定理之中的特例，例如著名的 Cauchy – Davenport inequality，其等號成立的條件恰好就為兩集合本身

皆為等差數列。有許多的數學家也對此很感興趣，而創造出許多不可思議的成果。我們接下來會盡可能的完成我們原題目的最佳估計，並且將問題嘗試做出一些改變，例如改為研究等差數列的長度為4出現的情況，如果可行我們也有試圖將這個一維的問題拉高一個維度，將這個問題結合平面來探討，抑或是利用我們找到的結果去嘗試利用不同的方法去解釋許多相關的定理，例如 van der Waerden 定理。過去的大數學家 Paul Erdös 也有對等差數列相關的問題進行過研究，且留下了許多的研究成果，我們會參考目前已知的定理，嘗試在不使用許多高等數學如微積分、傅立葉分析等的情況下，也能夠獲得相同的結果，然後針對已知定理做出延伸和優化，相信我們做出來的結果會十分豐富且創新，也會對世界上其他研究等差數列相關的問題做出貢獻。

捌、 參考資料資料

1. Erdos, Paul. *A Tribute to Paul Erdos*. Cambridge University Press, 1990.

2. Erdős, Paul*;* Turán, Paul *(1936).* "On some sequences of integers" . Journal of the London Mathematical Society*. 11(4): 261–264.*

3. Szemerédi, Endre (1975). "On sets of integers containing no kelements in arithmetic progression" . Acta Arithmetica. 27: 199–245.

4. F. Behrend, On sets of integers which contain no three terms in arithmetical progression, Proc. Nat. Acad. Sci. USA 32 (1946) 331-332.

5. Croot, Ernie, Imre Ruzsa, and Tomasz Schoen. "Arithmetic progressions in sparse sumsets." *Combinatorial number theory*. De Gruyter, 2011. 157-164.

6. Gasarch, William, James Glenn, and Clyde P. Kruskal. "Finding large 3-free sets I: The small n case." *Journal of Computer and System Sciences* 74.4 (2008): 628-655.

【評語】010023

作品著手的問題懸宕多年，迭經 Erdős、Szemerédi 等人攻堅。作者勇於挑戰難題，難能可貴。作品條理明暢，言之有物，但畢竟攻堅不易，難有實質突破。電腦演算效能亦差強人意。建議多方探索，厚植實力，並著手改善電腦演算法。