中華民國第 60 屆中小學科學展覽會
作品說明書

國中組　數學科

030407

分堆問題之收斂性的探討

學校名稱：桃園市立龍岡國民中學

作者：	指導老師：
國二 蘇皓暘	許珮禎
國二 張成緯	林樂鄺
國二 吳家宏	

關鍵詞：收斂、迴圈、歸納

摘　　要

此作品研究「在 n 堆彈珠數目的分堆問題，當每堆個數不同並進行分配，每次都從最多個數那堆拿最少個數那堆的數目給最少個數的那堆，直到收斂或出現迴圈時，則停止分配彈珠。」首先，探究分配過程中，奇偶數變化與總和(∑)的關聯性，進而觀察出可達成穩定狀態(收斂)之條件的規律；而當無法達成穩定狀態時，起始彈珠數之總和固定，在甚麼條件下會形成最大迴圈或最小迴圈，其迴圈數是多少，而迴圈的個數又會是多少；並探討起始值在甚麼條件下，其分配過程中，每個組合都能經過且不重複；最後，將分配過程所找出的規律，運用數學邏輯並加入條件因素，寫成演算法，利用程式 GDScript 動態語言執行，並呈現其分配過程及驗證我們的方法。

壹、研究動機

本題目的主要動機是來自於科展作品(第 58 屆 國小組: 公平分配遊戲)的狀況是：「現有兩堆彈珠，一堆是 23 個彈珠，另一堆是 9 個彈珠，如何分配讓這兩堆數目不一樣的彈珠進行平分？同時在每次進行分配時，都從數量較多的那一堆拿出較少數量那堆的個數，將其放到數量較少的那堆，反覆進行這樣的動作，直到因兩堆數量相等而無法分配。」

更進一步，我們也想到如果最開始時並不是兩堆呢？因而衍生從兩堆彈珠變成三堆彈珠，然而兩堆彈珠的分配方式，總是由多的一堆分配到少的一堆，但是從分為三堆彈珠時，分配的方式已不只一種，該分配哪兩堆是一個需考量的因素，倘若變成四堆、五堆及六堆彈珠時，分配方法也非常多種，於是我們上網參考其他相關的科展作品(第 55 屆 國中組: 從平分問題到動態穩定)(第 56 屆 國中組: 再探均分問題的動態穩定)之分配方法，而在確立方法後，我們就開始探究分堆彈珠遊戲之規律並將其延伸至 n 堆的情形，例如:分配過程中的奇偶數性質有何規律的變化性？穩定狀態與分配情形的關聯性又如何？而當為 n 堆達到穩定狀態時的一般式(56 屆已有歸納、探討出，但尚未證明)，所以我們轉而探討從無法達到穩定狀態並形成迴圈的狀態下，起始彈珠數之總和固定時，在甚麼條件下會形成最大迴圈或最小迴圈？其迴圈數是多少？迴圈的個數又有幾個？最後，再探討總和為定值時，起始值在甚麼條件下，其分配過程中所有組合都能經過且不重複。以下是我們的研究過程。

貳、研究目的

有數堆彈珠，其個數不一樣(a+b+c+……=∑)進行分配，每次都從最多個數那堆拿最少個數那堆的數目給最少個數的那堆，直到收斂或出現迴圈時，則停止分配彈珠。

一、分為兩堆：

1.起始組合要有什麼條件?此分堆問題會收斂，而當起始值為何時，能每一個狀態都走過?

2.當分堆問題會收斂時，其過程是不是會有均分的情況?當有均分的情形發生時，其過程會不會收斂?

3.若不收斂，起始值為何時，會形成最大迴圈或最小迴圈? 其迴圈數會是多少？迴圈的個數又會是多少？

二、分為三堆、四堆到 n 堆：

1.起始組合要有什麼條件?此分堆問題會收斂，而當起始值為何時，能每一個狀態都走過?

2.若收斂且過程中有均分的情形時，起始值要符合甚麼條件?當有均分的情形發生時，其過程會不會收斂?

3.若不收斂，起始值為何時，會形成最大迴圈或最小迴圈? 其迴圈數會是多少？迴圈的個數又會是多少？

三、將分配過程所找出的規律，運用數學邏輯並加入條件因素，以程式 GDScript 動態語言寫出執行，並呈現其分配過程。

參、研究設備及器材

一、筆記本、筆(記錄用)　二、電腦、自備隨身碟　三、Microsoft Excel 2010

四、Godot Engine 合法免費授權軟體

肆、文獻探討

做本研究之前，我們先參考了一些別人的文獻資料：

屆數	組別	作品名稱	研究方法	得獎情形
55	國中組	從平分問題到動態穩定	輾轉相除法及二進位制	佳作
56	國中組	再探均分問題的動態穩定	二進位制	佳作
58	國小組	公平分配遊戲	公因數	

表一：文獻資料

從全國中小學科展網站查詢到相關系列議題的資料：第 55 屆全國科展國中組數學科佳作「從平分問題到動態穩定」其研究是：將兩堆小石子利用輾轉相除法記錄移動過程，使平分問題轉化成二進位制的移位問題！第 56 屆全國科展國中組數學科佳作「再探均分問題的動態穩定」其研究是：將三堆石頭移動成數量皆相等的狀態，利用二進位制歸納三堆時的移動次數型態，找出可形成穩定狀態的條件及移動方法與次數。第 58 屆全國科展國小組數學科「公平分配遊戲」其研究是：將兩堆糖果公平分配成功的情形，觀察出 A+B=4n 至少有一組成功解及 $A+B=2^n$ 一律成功，但都缺少了一般化的證明。

因此，在第 60 屆我們的作品從分堆問題所形成的狀態作探究，其中有許多令人好奇的點，透過不同思維想法與方式，我們將分堆數加以擴充，再研究第 55 、56 屆的作品，並定義出分配多堆彈珠的方法，而本作品研究方法分為三個方向:1. 先探究分配過程中，奇偶數變化與總和(∑)的關聯性，進而觀察出可達成穩定狀態(收斂)之條件，並探討穩定狀態與均分的關聯性。2. 探討總和為定值時，起始值在甚麼條件下，其分配過程中所有組合都能經過且不重複。3. 在無法達到穩定狀態並形成迴圈的情形下，起始彈珠數的總和固定時，在甚麼條件下會形成最大迴圈或最小迴圈？其迴圈數會是多少？而迴圈的個數又是多少？以上三點與上述三件參考文獻的**研究方法上，有著極大不同的探討方向，也是歷屆作品所沒有做過的**，期盼研究性價值在題材創新度及作品內容能有提升的空間，讓結果更加廣義且完備，故我們以此作為主題，做以下一系列的探討及研究。

伍、研究過程及方法

研究期間，分成了不少階段，也有不同的突破，下表是我們研究進度概要：

時間	進展
108.09	發現將兩堆糖果公平分配成功的情形，並開始研究「公平分配遊戲」(二)
108.10	發現將兩堆小石子平分問題轉成移位問題，並著手研究「從平分問題到動態穩定」(四)
108.11	發現將三堆石頭移動成數量皆相等的狀態，並著手研究「再探均分問題的動態穩定」(三)
108.12 109.01	嘗試推廣到四堆、五堆及六堆彈珠甚至 n 堆彈珠時，進行分類探討並找出各種情形在達成穩定狀態的規律，並嘗試證明第 56 屆推導出的一般式。
109.02 109.03	探討在推廣到 n 堆彈珠時，當達成穩定狀態的規律下，同時討論奇偶數變化的規律性與總和(∑)之關聯性；嘗試利用分組拆法來進行分配，並觀察其中規律。
109.04	將分配過程所導出的規律，運用數學邏輯並加入條件因素，以程式 GDScript 動態語言寫出執行，並予以呈現。

109.05	探討總和固定下，在何種情況會收斂；研究達成穩定狀態時會不會收斂並加以證明。
109.06	觀察總和固定下，起始值為何時，會形成最大迴圈或最小迴圈？其迴圈數會是多少？而迴圈個數又會是多少？與總和是多少時，每個狀態都能走過且不重複？

表二:研究進度概要

一、名詞定義：

1.分堆與分配的定義：

(1) 所有任意組合，且不論分為幾堆，皆從數對組合之最大的數中拿出最小數的值給最小數，且一次只分配一個最大數和一個最小數，直到收斂或形成迴圈時，則停止分配。

Ex:(2,2,8) →(4,2,6) →(4,4,4) →成功。

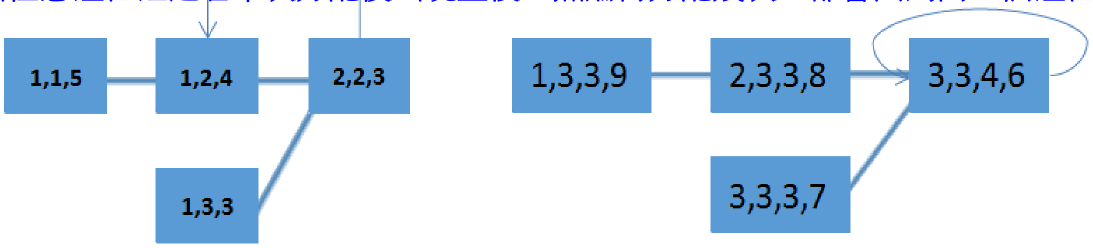
(2) 倘若不依循規則，起始組合一次分配數個最小數時，則最大數可能會變為負數，由此可知，要依循規則。

Ex: (2,2,2,2,7) →(4,4,4,4,-1) →出現負數，無解。

2.穩定狀態(均分)的定義：**將彈珠分為 n 堆時，每堆分別有 a,b,c,d,……個，若每堆都相等，則數對(a,b,c,d,……)稱為「穩定狀態」。**

3.迴圈的定義：

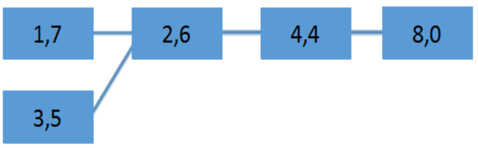
指任意組合經過若干次分配後出現重複，無論再分配幾次，都會回到同一個組合。



4.迴圈數： 迴圈內第一次出現某組合到第二次出現重複之間經過的分配次數。

5.迴圈個數：**在同總和下，出現的所有不同迴圈之個數。**

6.收斂的定義：**將彈珠分為 n 堆時，每堆分別有 a,b,c,d,……個，當組合中有一堆為 0 的狀態，也就是無法再分配的情況。**



二、可均分的數字總和(∑)或不可均分的數字總和(∑)，其分別會出現的規律與產生相對應的狀況是什麼？當推廣到多堆數目時，在達穩定狀態下，若將∑分為數堆不同數目相加時，會有相同或相異的點嗎？部分可均勻分配的數，遇到什麼狀況會無法均勻分配？

1.分為兩堆彈珠：數對分配過程

列出分為兩堆，∑為 4、8、12、16、20、24 時，能否形成穩定狀態的情形:

4	8	12	16	20	24
所有組合皆可成功 所有數對倒數第二步的比例皆為 1:3 Ex:(1,3)	所有組合皆可成功 所有數對倒數第二步的比例皆為 1:3 Ex:(3,5)	大部分皆失敗，而成功的數對中，倒數第二步的比例皆為 1:3 Ex:(9,3)	所有組合皆可成功 所有數對倒數第二步的比例皆為 1:3 Ex:(11,5)	大部分皆失敗，而成功的數對中，倒數第二步的比例皆為 1:3 Ex:(5,15)	大部分皆失敗，而成功的數對中，倒數第二步的比例皆為 1:3 Ex:(18,6)

以提出公因數的方法，列出關係式，並找出各數的所有分配情形：

4		
(a,b)	關係式	說明
2A+2B	A+B=2	只有一組解
4A+4B	A+B=1	無正整數解
8		
(a,b)	關係式	說明
2A+2B	A+B=4	全部成功
4A+4B	A+B=2	只有一組解
8A+8B	A+B=1	無正整數解
12		
(a,b)	關係式	說明
2A+2B	A+B=6	全部失敗
3A+3B	A+B=4	全部成功
4A+4B	A+B=3	全部失敗
6A+6B	A+B=2	只有一組解
12A+12B	A+B=1	無正整數解
16		
(a,b)	關係式	說明
2A+2B	A+B=8	全部成功
4A+4B	A+B=4	全部成功
8A+8B	A+B=2	只有一組解
16A+16B	A+B=1	無正整數解
20		
(a,b)	關係式	說明
2A+2B	A+B=10	全部失敗
4A+4B	A+B=5	全部失敗

5A+5B	A+B=4	全部成功
10A+10B	A+B=2	只有一組解
20A+20B	A+B=1	無正整數解
24		
(a,b)	關係式	說明
2A+2B	A+B=12	部分成功
3A+3B	A+B=8	全部成功
4A+4B	A+B=6	全部失敗
6A+6B	A+B=4	全部成功
8A+8B	A+B=3	全部失敗
12A+12B	A+B=2	只有一組解
24A+24B	A+B=1	無正整數解

2.分為三堆彈珠：數對分配過程

列出分為三堆，∑為 6、12、18、24 時，能否形成穩定狀態的情形:

<table>
<tr><th>6</th><th>12</th><th>18</th><th colspan="2">24</th></tr>
<tr><td rowspan="3">所有組合皆可成功
所有數對倒數第二步的比例皆為 1:2:3
Ex:(1,1,4)</td><td rowspan="3">大部分皆成功，而失敗的數對倒數第二步中，最大數為最小數的兩倍，故無法形成穩定狀態。
Ex:(3,3,6)</td><td rowspan="3">大部分皆失敗，而成功的數對中，倒數第二步比例皆為 1:2:3
Ex:(3,3,12)</td><td colspan="2">成功與失敗的組合約各占一半。</td></tr>
<tr><td>成功的組合</td><td>失敗的組合</td></tr>
<tr><td>成功的組合中，倒數第二步比例皆為 1:2:3
Ex:(2,4,18)</td><td>有兩種情況:
1.倒數第二步中，最大數為最小數的兩倍。Ex:(3,6,15)
2.某個固定的組合(Ex:7,7,10)，出現一次過後，經若干步後又重複出現。Ex:(5,7,12)</td></tr>
</table>

以提出公因數的方法，列出關係式，並找出各數的所有分配情形：

12		
(a,b,c)	關係式	說明
2A+2B+2C	A+B+C=6	全部成功
3A+3B+3C	A+B+C=4	全部失敗
4A+4B+4C	A+B+C=3	只有一組解
6A+6B+6C	A+B+C=2	無正整數解
12A+12B+12C	A+B+C=1	無正整數解
18		
(a,b,c)	關係式	說明
2A+2B+2C	A+B+C=9	全部失敗
3A+3B+3C	A+B+C=6	全部成功
6A+6B+6C	A+B+C=3	只有一組解
9A+9B+9C	A+B+C=2	無正整數解
18A+18B+18C	A+B+C=1	無正整數解

24		
(a,b,c)	關係式	說明
2A+2B+2C	A+B+C=12	部分成功
3A+3B+3C	A+B+C=8	全部失敗
4A+4B+4C	A+B+C=6	全部成功
6A+6B+6C	A+B+C=4	全部失敗
8A+8B+8C	A+B+C=3	只有一組解
12A+12B+12C	A+B+C=2	無正整數解
24A+24B+24C	A+B+C=1	無正整數解

3.分為四堆彈珠：數對分配過程

列出分為四堆，∑為 8、16、24、32 時，能否形成穩定狀態的情形:

<table>
<tr><th rowspan="3">8</th><th rowspan="3">16</th><th rowspan="3">24</th><th colspan="2">32</th></tr>
<tr><td colspan="2">成功與失敗的組合約各占一半。</td></tr>
<tr><td>成功的組合</td><td>失敗的組合</td></tr>
<tr><td>所有組合皆可成功
所有數對倒數第二步的比例皆為 1:2:2:3
Ex:(1,1,1,5)</td><td>大部分皆成功，而失敗的數對倒數第二步中，最大數為最小數的兩倍，故無法形成穩定狀態。
Ex:(1,3,3,9)</td><td>大部分皆失敗，而成功的數對中，倒數第二步比例皆為 1:2:2:3
Ex:(3,3,3,15)</td><td>成功的組合中，倒數第二步比例皆為 1:2:2:3
Ex:(1,3,4,24)</td><td>有兩種情況:
1.倒數第二步中，最大數為最小數的兩倍
Ex:(1,3,6,22)
2.某個固定的組合，出現一次過後，經若干步後又重複出現。
Ex:(1,3,7,21)</td></tr>
</table>

以提出公因數的方法，列出關係式，並找出各數的所有分配情形:

16		
(a,b,c,d)	關係式	說明
2A+2B+2C+2D	A+B+C+D=8	全部成功
4A+4B+4C+4D	A+B+C+D=4	只有一組解
8A+8B+8C+8D	A+B+C+D=2	無正整數解
16A+16B+16C+16D	A+B+C+D=1	無正整數解
24		
(a,b,c,d)	關係式	說明
2A+2B+2C+2D	A+B+C+D=12	全部失敗
3A+3B+3C+3D	A+B+C+D=8	全部成功
4A+4B+4C+4D	A+B+C+D=6	全部失敗
6A+6B+6C+6D	A+B+C+D=4	只有一組解
8A+8B+8C+8D	A+B+C+D=3	無正整數解
12A+12B+12C+12D	A+B+C+D=2	無正整數解
24A+24B+24C+24D	A+B+C+D=1	無正整數解
32		
(a,b,c,d)	關係式	說明

2A+2B+2C+2D	A+B+C+D=16	部分成功
4A+4B+4C+4D	A+B+C+D=8	全部成功
8A+8B+8C+8D	A+B+C+D=4	只有一組解
16A+16B+16C+16D	A+B+C+D=2	無正整數解
32A+32B+32C+32D	A+B+C+D=1	無正整數解

4.分為五堆彈珠：數對分配過程

列出分為五堆，∑為 10、20、30、40 時，能否形成穩定狀態的情形:

<table>
<tr><th>10</th><th>20</th><th>30</th><th>40</th></tr>
<tr>
<td>所有組合皆可成功
所有數對倒數第二步的比例皆為 1:2:2:2:3
Ex:(1,1,1,3 ,4)</td>
<td>大部分皆成功，而失敗的數對倒數第二步中，最大數為最小數的兩倍，故無法形成穩定狀態。
Ex:(2,2,3,3,10)</td>
<td>大部分皆失敗，而成功的數對中，倒數第二步比例皆為 1:2:2:2:3
Ex:(3,3,6,9,9)</td>
<td>成功與失敗的組合約各占一半。
<table>
<tr><th>成功的組合</th><th>失敗的組合</th></tr>
<tr><td>成功的組合中，倒數第二步比例皆為 1:2:2:2:3
Ex:(1,1,2,2,34)</td><td>有兩種情況:
1.倒數第二步中，最大數為最小數的兩倍
Ex:(1,1,3,6,29)
2.某個固定的組合，出現一次過後，經若干步後又重複出現。
Ex:(1,2,7,10,20)</td></tr>
</table>
</td>
</tr>
</table>

以提出公因數的方法，列出關係式，並找出各數的所有分配情形:

20		
(a,b,c,d,e)	關係式	說明
2A+2B+2C+2D+2E	A+B+C+D+E=10	全部成功
4A+4B+4C+4D+4E	A+B+C+D+E=5	只有一組解
5A+5B+5C+5D+5E	A+B+C+D+E=4	無正整數解
10A+10B+10C+10D+10E	A+B+C+D+E=2	無正整數解
20A+20B+20C+20D+20E	A+B+C+D+E=1	無正整數解
30		
(a,b,c,d,e)	關係式	說明
2A+2B+2C+2D+2E	A+B+C+D+E=15	全部失敗
3A+3B+3C+3D+3E	A+B+C+D+E=10	全部成功
5A+5B+5C+5D+5E	A+B+C+D+E=6	全部失敗
6A+6B+6C+6D+6E	A+B+C+D+E=5	只有一組解
10A+10B+10C+10D+10E	A+B+C+D+E=3	無正整數解
15A+15B+15C+15D+15E	A+B+C+D+E=2	無正整數解
30A+30B+30C+30D+30E	A+B+C+D+E=1	無正整數解
40		
(a,b,c,d,e)	關係式	說明
2A+2B+2C+2D+2E	A+B+C+D+E=20	部分成功
4A+4B+4C+4D+4E	A+B+C+D+E=10	全部成功
5A+5B+5C+5D+5E	A+B+C+D+E=8	全部失敗

8A+8B+8C+8D+8E	A+B+C+D+E=5	只有一組解
10A+10B+10C+10D+10E	A+B+C+D+E=4	無正整數解
20A+20B+20C+20D+20E	A+B+C+D+E=2	無正整數解
40A+40B+40C+40D+40E	A+B+C+D+E =1	無正整數解

5.分為六堆彈珠：數對分配過程

列出分為六堆，∑為 12、24、36、48 時，能否形成穩定狀態的情形:

<table>
<tr><th>12</th><th>24</th><th>36</th><th colspan="2">48</th></tr>
<tr><td rowspan="3">所有組合皆可成功
所有數對倒數第二步的比例皆為 1:2:2:2:2:3
Ex:(1,1,1,1,1,7)</td><td rowspan="3">大部分皆成功，而失敗的數對倒數第二步中，最大數為最小數的兩倍，故無法形成穩定狀態。
Ex:
(1,1,3,3,6,10)</td><td rowspan="3">大部分皆失敗，而成功的數對中，倒數第二步比例皆為
1:2:2:2:2:3
Ex:(3,3,3,3,3,21)</td><td colspan="2">成功與失敗的組合約各占一半。</td></tr>
<tr><td>成功的組合</td><td>失敗的組合</td></tr>
<tr><td>成功的組合中，倒數第二步比例皆為 1:2:2:2:2:3
Ex:(1,1,1,2,16,27)</td><td>有兩種情況:
1.倒數第二步中，最大數為最小數的兩倍
Ex:(1,1,1,3,3,39)
2.某個固定的組合，出現一次過後，經若干步後又重複出現。
Ex:(1,1,1,6,11,28)</td></tr>
</table>

以提出公因數的方法，列出關係式，並找出各數的所有分配情形:

24		
(a,b,c,d,e,f)	關係式	說明
2A+2B+2C+2D+2E+2F	A+B+C+D+E+F=12	全部成功
3A+3B+3C+3D+3E+3F	A+B+C+D+E+F=8	全部失敗
4A+4B+4C+4D+4E+4F	A+B+C+D+E+F=6	只有一組解
6A+6B+6C+6D+6E+6F	A+B+C+D+E+F=4	無正整數解
8A+8B+8C+8D+8E+8F	A+B+C+D+E+F=3	無正整數解
12A+12B+12C+12D+12E+12F	A+B+C+D+E+F=2	無正整數解
24A+24B+24C+24D+24E+24F	A+B+C+D+E+F=1	無正整數解
36		
(a,b,c,d,e,f)	關係式	說明
2A+2B+2C+2D+2E+2F	A+B+C+D+E+F=18	全部失敗
3A+3B+3C+3D+3E+3F	A+B+C+D+E+F=12	全部成功
4A+4B+4C+4D+4E+4F	A+B+C+D+E+F=9	全部失敗
6A+6B+6C+6D+6E+6F	A+B+C+D+E+F=6	只有一組解
9A+9B+9C+9D+9E+9F	A+B+C+D+E+F=4	無正整數解
12A+12B+12C+12D+12E+12F	A+B+C+D+E+F=3	無正整數解
18A+18B+18C+18D+18E+18F	A+B+C+D+E+F=2	無正整數解
36A+36B+36C+36D+36E+36F	A+B+C+D+E+F=1	無正整數解
48		
(a,b,c,d,e,f)	關係式	說明
2A+2B+2C+2D+2E+2F	A+B+C+D+E+F=24	部分成功

3A+3B+3C+3D+3E+3F	A+B+C+D+E+F=16	全部失敗
4A+4B+4C+4D+4E+4F	A+B+C+D+E+F=12	全部成功
6A+6B+6C+6D+6E+6F	A+B+C+D+E+F=8	全部失敗
8A+8B+8C+8D+8E+8F	A+B+C+D+E+F=6	只有一組解
12A+12B+12C+12D+12E+12F	A+B+C+D+E+F=4	無正整數解
16A+16B+16C+16D+16E+16F	A+B+C+D+E+F=3	無正整數解
24A+24B+24C+24D+24E+24F	A+B+C+D+E+F=2	無正整數解
48A+48B+48C+48D+48E+48F	A+B+C+D+E+F=1	無正整數解

三、分組拆法：

定義:在分為多堆時，將堆數分成為若干個小組後，再進行分配。

例:分為四堆→分為兩堆、兩堆

1. 分為四堆以上時，才可使用分組拆法，可將分配步驟減短。
2. 若分為 x 個小組，其步驟會變為分組前的 1/x 。
3. 可使用分組拆法的起始組合之情形:
   (1) 最大數與最小數之數量相同。
   (2) 所有數成等差數列時(並非所有等差數列都可使用，只限於公差為 2)。

Ex：分為四堆時，左圖為未分組時的分配情形，右圖為分組後的分配情形

1	1	3	3
1	2	2	3
2	2	2	2

1	3
2	2

1	3
2	2

Ex：分為五堆時，左圖為未分組時的分配情形，右圖為分組後的分配情形

1	1	2	3	3
1	2	2	2	3
2	2	2	2	2

1	2	3
2	2	2

1	3
2	2

Ex：分為六堆時，左圖為未分組時的分配情形，右圖為分組後的分配情形

3	5	7	9	11	13
6	5	7	9	11	10
6	10	7	9	6	10
12	4	7	9	6	10
8	8	7	9	6	10
8	8	7	9	12	4
8	8	7	9	8	8
8	8	14	2	8	8
8	8	12	4	8	8
8	8	8	8	8	8

3	13
6	10
12	4
8	8

7	9
14	2
12	4
8	8

5	11
10	6
4	12
8	8

四、通式: $\sum=n\times p\times 2^k$ (此通式為第 56 屆所列出的，但其缺少證明)

設組合中的數堆彈珠數量總和為∑，且分為 n 堆相加，其中 p 為正奇數($1\le p$)，而 $1\le k$

(一) 若某總和能分解成此通式的形式，那麼此總和至少會有一組以上的數對會收斂。

1.全部組合皆成功的數，分解後 p 一定為 1；

2.部分組合成功的數，分解後 p 不一定為 1；

3.當∑為 2n 乘上一個奇質數的倍數時，P 必為此奇質數的倍數；

Ex: $\sum=a+b+c+d+e+f+g+h=48(16\times3)$

依通式分解為 $48=8\times3\times2^1$，因 48 為 16 的 3 倍，所以 p 為 3 的倍數。

4.當 2n 乘上 2 的次方倍時，p 等於 1(依照通式分解後，p 一定是 1)。

(1) 比例規則：成功倒數第二步需為 1:2:⋯⋯:2:3

【證明】:

倒數第二步 $m:\underbrace{2m:2m:\cdots\cdots:2m}_{(n-2)}:3m$

$\downarrow$

經由一步分配，$\because m$ 最小，$3m$ 最大。$\therefore (m+m):2m:2m:\cdots\cdots:2m:(3m-m)$

$\Rightarrow 2m:2m:2m\ldots\ldots 2m:2m$

(2) 通式($n\times p\times 2^k$)與比例規則(成功倒數第二步需為 1:2:⋯⋯:2:3)相符合。

【證明】:

有 n 堆彈珠的總和(a+b+c+d+⋯⋯=∑)

若將 a+b+c+d+⋯⋯分解成 $1:\underbrace{2:2:\cdots\cdots:2}_{(n-2)}:3$ 的比例時，必成功。

令 $a=m$、$b=2m$、$c=2m$⋯⋯、最後一堆$=3m$

$a+b+c+d+\cdots\cdots=m+2m\cdot(n-2)+3m=2m\cdot n=\sum$ ($1\le m$，$m\in\mathbb{N}$)有兩種情況：

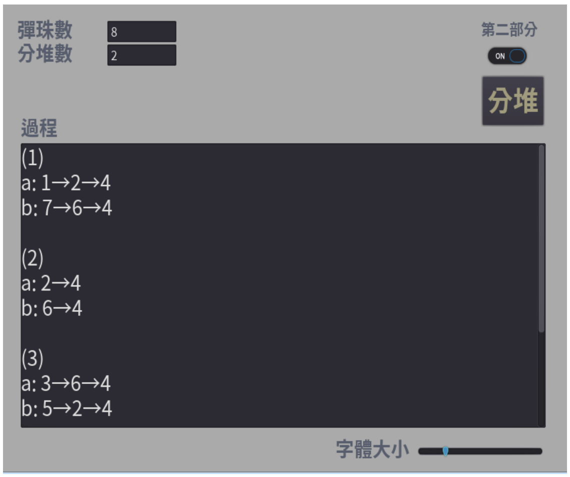
情況①: m 為奇數，$2\times m\times n$

情況②: m 為偶數，所以可將 m 分解為 $2^k\times p$(奇數)

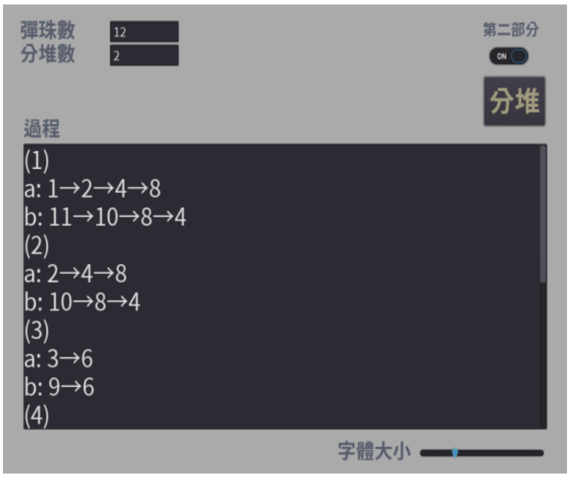
由情況①、②可推得；通式$\sum=n\times p\times 2^k$，且 $1\le k$，k 為自然數。

(3)先以通式分解可收斂的總和，分為兩堆且總和為 8(所有組合皆可收斂)與總和為 12(部分組合可收斂)，再以程式呈現其分配結果。

①.總和為 8，依通式分解為:$2\times1\times2^2$



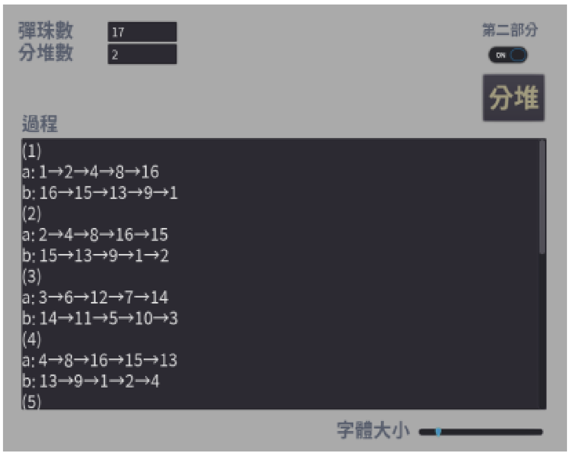
②.總和為 12，依通式分解為:$2\times3\times2^1$



(二) 若某總和不能分解成此通式的形式，那麼此總和的所有數對都不會收斂。

1.利用程式呈現無法分解成此通式之總和的分配結果。(總和為 9、總和為 10)

2.利用程式呈現無法分解成此通式之總和(奇質數)的分配結果。(總和為 7、總和為 17)



五、總和為多少時，所有狀態都能走過且不重複？

1.以下各總和所有狀態皆能一次經過：

a+b=3　(1) a: 1→2，b: 2→1。

a+b=5　(1) a: 1→2→4，b: 4→3→1；(2) a: 2→4→3，b: 3→1→2。

a+b=7　(1) a: 1→2→4→1，b: 6→5→3→6；(2) a: 2→4→1→2，b: 5→3→6→5；
(3) a: 3→6→5→3，b: 4→1→2→4。

a+b=11　(1) a: 1→2→4→8→5→10，b: 10→9→7→3→6→1；
(2) a: 2→4→8→5→10→9，b: 9→7→3→6→1→2；
(3) a: 3→6→1→2→4→8，b: 8→5→10→9→7→3；
(4) a: 4→8→5→10→9→7，b: 7→3→6→1→2→4；
(5) a: 5→10→9→7→3→6，b: 6→1→2→4→8→5。

a+b=13　(1) a:1→2→4→8→3→6→12，b: 12→11→9→5→10→7→1；
(2) a: 2→4→8→3→6→12→11，b: 11→9→5→10→7→1→2；
(3) a: 3→6→12→11→9→5→10，b: 10→7→1→2→4→8→3；
(4) a: 4→8→3→6→12→11→9，b: 9→5→10→7→1→2→4；
(5) a: 5→10→7→1→2→4→8，b: 8→3→6→12→11→9→5；
(6) a: 6→12→11→9→5→10→7，b: 7→1→2→4→8→3→6。

a+b=19　(1) a: 1→2→4→8→16→13→7→14→9→18，b: 18→17→15→11→3→6→12→5→10→1；
(2) a: 2→4→8→16→13→7→14→9→18→17，b: 17→15→11→3→6→12→5→10→1→2；
(3) a: 3→6→12→5→10→1→2→4→8→16，b: 16→13→7→14→9→18→17→15→11→3；
(4) a: 4→8→16→13→7→14→9→18→17→15，b: 15→11→3→6→12→5→10→1→2→4；
(5) a: 5→10→1→2→4→8→16→13→7→14，b: 14→9→18→17→15→11→3→6→12→5；
(6) a: 6→12→5→10→1→2→4→8→16→13，b: 13→7→14→9→18→17→15→11→3→6；
(7) a: 7→14→9→18→17→15→11→3→6→12，b: 12→5→10→1→2→4→8→16→13→7；
(8) a: 8→16→13→7→14→9→18→17→15→11，b: 11→3→6→12→5→10→1→2→4→8；
(9) a: 9→18→17→15→11→3→6→12→5→10，b: 10→1→2→4→8→16→13→7→14→9。

a+b=23　(1) a: 1→2→4→8→16→9→18→13→3→6→12→1，
b: 22→21→19→15→7→14→5→10→20→17→11→22；
(2) a: 2→4→8→16→9→18→13→3→6→12→1→2，
b: 21→19→15→7→14→5→10→20→17→11→22→21；
(3) a: 3→6→12→1→2→4→8→16→9→18→13→3，
b: 20→17→11→22→21→19→15→7→14→5→10→20；
(4) a: 4→8→16→9→18→13→3→6→12→1→2→4，
b: 19→15→7→14→5→10→20→17→11→22→21→19；
(5) a: 5→10→20→17→11→22→21→19→15→7→14→5，
b: 18→13→3→6→12→1→2→4→8→16→9→18；
(6) a: 6→12→1→2→4→8→16→9→18→13→3→6，
b: 17→11→22→21→19→15→7→14→5→10→20→17；
(7) a: 7→14→5→10→20→17→11→22→21→19→15→7，
b: 16→9→18→13→3→6→12→1→2→4→8→16；
(8) a: 8→16→9→18→13→3→6→12→1→2→4→8，
b: 15→7→14→5→10→20→17→11→22→21→19→15；
(9) a:9→18→13→3→6→12→1→2→4→8→16→9，
b: 14→5→10→20→17→11→22→21→19→15→7→14；

(10) a: 10→20→17→11→22→21→19→15→7→14→5→10，
b: 13→3→6→12→1→2→4→8→16→9→18→13；
(11) a: 11→22→21→19→15→7→14→5→10→20→17→11，
b: 12→1→2→4→8→16→9→18→13→3→6→12，

a+b=29 (1) a: 1→2→4→8→16→3→6→12→24→19→9→18→7→14→28，
b: 28→27→25→21→13→26→23→17→5→10→20→11→22→15→1；
(2) a: 2→4→8→16→3→6→12→24→19→9→18→7→14→28→27，
b: 27→25→21→13→26→23→17→5→10→20→11→22→15→1→2；
(3) a: 3→6→12→24→19→9→18→7→14→28→27→25→21→13→26，
b: 26→23→17→5→10→20→11→22→15→1→2→4→8→16→3；
(4) a: 4→8→16→3→6→12→24→19→9→18→7→14→28→27→25，
b: 25→21→13→26→23→17→5→10→20→11→22→15→1→2→4；
(5) a: 5→10→20→11→22→15→1→2→4→8→16→3→6→12→24，
b: 24→19→9→18→7→14→28→27→25→21→13→26→23→17→5；
(6) a : 6→12→24→19→9→18→7→14→28→27→25→21→13→26→23，
b: 23→17→5→10→20→11→22→15→1→2→4→8→16→3→6；
(7) a: 7→14→28→27→25→21→13→26→23→17→5→10→20→11→22，
b: 22→15→1→2→4→8→16→3→6→12→24→19→9→18→7；
(8) a: 8→16→3→6→12→24→19→9→18→7→14→28→27→25→21，
b: 21→13→26→23→17→5→10→20→11→22→15→1→2→4→8；
(9) a: 9→18→7→14→28→27→25→21→13→26→23→17→5→10→20，
b: 20→11→22→15→1→2→4→8→16→3→6→12→24→19→9；
(10) a: 10→20→11→22→15→1→2→4→8→16→3→6→12→24→19，
b: 19→9→18→7→14→28→27→25→21→13→26→23→17→5→10；
(11) a: 11→22→15→1→2→4→8→16→3→6→12→24→19→9→18，
b: 18→7→14→28→27→25→21→13→26→23→17→5→10→20→11；
(12) a: 12→24→19→9→18→7→14→28→27→25→21→13→26→23→17，
b: 17→5→10→20→11→22→15→1→2→4→8→16→3→6→12；
(13) a: 13→26→23→17→5→10→20→11→22→15→1→2→4→8→16，
b: 16→3→6→12→24→19→9→18→7→14→28→27→25→21→13；
(14) a: 14→28→27→25→21→13→26→23→17→5→10→20→11→22→15，
b: 15→1→2→4→8→16→3→6→12→24→19→9→18→7→14。

a+b=37 (1) a: 1→2→4→8→16→32→27→17→34→31→25→13→26→15→30→23→9→18→36，
b: 36→35→33→29→21→5→10→20→3→6→12→24→11→22→7→14→28→19→1；
(2) a: 2→4→8→16→32→27→17→34→31→25→13→26→15→30→23→9→18→36→35，
b: 35→33→29→21→5→10→20→3→6→12→24→11→22→7→14→28→19→1→2；
(3) a: 3→6→12→24→11→22→7→14→28→19→1→2→4→8→16→32→27→17→34，
b: 34→31→25→13→26→15→30→23→9→18→36→35→33→29→21→5→10→20→3；
(4) a: 4→8→16→32→27→17→34→31→25→13→26→15→30→23→9→18→36→35→33，
b: 33→29→21→5→10→20→3→6→12→24→11→22→7→14→28→19→1→2→4；
(5) a: 5→10→20→3→6→12→24→11→22→7→14→28→19→1→2→4→8→16→32，
b: 32→27→17→34→31→25→13→26→15→30→23→9→18→36→35→33→29→21→5；
(6) a: 6→12→24→11→22→7→14→28→19→1→2→4→8→16→32→27→17→34→31，
b: 31→25→13→26→15→30→23→9→18→36→35→33→29→21→5→10→20→3→6；

(7) a: 7→14→28→19→1→2→4→8→16→32→27→17→34→31→25→13→26→15→30，
b: 30→23→9→18→36→35→33→29→21→5→10→20→3→6→12→24→11→22→7；
(8) a: 8→16→32→27→17→34→31→25→13→26→15→30→23→9→18→36→35→33→29，
b: 29→21→5→10→20→3→6→12→24→11→22→7→14→28→19→1→2→4→8；
(9) a: 9→18→36→35→33→29→21→5→10→20→3→6→12→24→11→22→7→14→28，
b: 8→19→1→2→4→8→16→32→27→17→34→31→25→13→26→15→30→23→9；
(10) a: 10→20→3→6→12→24→11→22→7→14→28→19→1→2→4→8→16→32→27，
b: 27→17→34→31→25→13→26→15→30→23→9→18→36→35→33→29→21→5→10；
(11) a: 11→22→7→14→28→19→1→2→4→8→16→32→27→17→34→31→25→13→26，
b: 26→15→30→23→9→18→36→35→33→29→21→5→10→20→3→6→12→24→11；
(12) a: 12→24→11→22→7→14→28→19→1→2→4→8→16→32→27→17→34→31→25，
b: 25→13→26→15→30→23→9→18→36→35→33→29→21→5→10→20→3→6→12；
(13) a: 13→26→15→30→23→9→18→36→35→33→29→21→5→10→20→3→6→12→24，
b: 24→11→22→7→14→28→19→1→2→4→8→16→32→27→17→34→31→25→13；
(14) a: 14→28→19→1→2→4→8→16→32→27→17→34→31→25→13→26→15→30→23，
b: 23→9→18→36→35→33→29→21→5→10→20→3→6→12→24→11→22→7→14；
(15) a: 15→30→23→9→18→36→35→33→29→21→5→10→20→3→6→12→24→11→22，
b: 22→7→14→28→19→1→2→4→8→16→32→27→17→34→31→25→13→26→15；
(16) a: 16→32→27→17→34→31→25→13→26→15→30→23→9→18→36→35→33→29→21，
b: 21→5→10→20→3→6→12→24→11→22→7→14→28→19→1→2→4→8→16；
(17) a: 17→34→31→25→13→26→15→30→23→9→18→36→35→33→29→21→5→10→20，
b: 20→3→6→12→24→11→22→7→14→28→19→1→2→4→8→16→32→27→17；
(18) a: 18→36→35→33→29→21→5→10→20→3→6→12→24→11→22→7→14→28→19，
b: 19→1→2→4→8→16→32→27→17→34→31→25→13→26→15→30→23→9→18。

2.以下各總和的所有狀態無法一次經過：

a+b=17 (1) a: 1→2→4→8→16，b: 16→15→13→9→1；(2) a: 2→4→8→16→15，b: 15→13→9→1→2；
(3) a: 3→8→12→7→14，b: 14→11→5→10→3；
(4) a: 4→8→16→15→13，b: 13→9→1→2→4；
(5) a: 5→10→3→6→12，b: 12→7→14→11→5；
(6) a: 6→12→7→14→11，b: 11→5→10→3→6；
(7) a: 7→14→11→5→10，b: 10→3→6→12→7；
(8) a: 8→16→15→13→9，b: 9→1→2→4→8。

a+b=31 (1) a: 1→2→4→8→16→1，b: 30→29→27→23→15→30；
(2) a: 2→4→8→16→1→2，b: 29→27→23→15→30→29；
(3) a: 3→6→12→24→17→3，b: 28→25→19→7→14→28；
(4) a: 4→8→16→1→2→4，b: 27→23→15→30→29→27；
(5) a: 5→10→20→9→18→5，b: 26→21→11→22→13→26；
(6) a: 6→12→24→17→3→6，b: 25→19→7→14→28→25；
(7) a: 7→14→28→25→19→7，b:24→17→3→6→12→24；
(8) a: 8→16→1→2→4→8，b: 23→15→30→29→27→23；
(9) a: 9→18→5→10→20→9，b: 22→13→26→21→11→22；
(10) a: 10→20→9→18→5→10，b: 21→11→22→13→26→21；
(11) a: 11→22→13→26→21→11，b:20→9→18→5→10→20；
(12) a: 12→24→17→3→6→12，b: 19→7→14→28→25→19；

(13) a: 13→26→21→11→22→13，b: 18→5→10→20→9→18；
(14) a: 14→28→25→19→7→14，b: 17→3→6→12→24→17；
(15) a: 15→30→29→27→23→15，b: 16→1→2→4→8→16。

a+b=41 (1) a: 1→2→4→8→16→32→23→5→10→20→40，
b: 40→39→37→33→25→9→18→36→31→21→1；
(2) a: 2→4→8→16→32→23→5→10→20→40→39，
b: 39→37→33→25→9→18→36→31→21→1→2；
(3) a: 3→6→12→24→7→14→28→15→30→19→38，
b: 38→35→29→17→34→27→13→26→11→22→3；
(4) a: 4→8→16→32→23→5→10→20→40→39→37，
b: 37→33→25→9→18→36→31→21→1→2→4；
(5) a: 5→10→20→40→39→37→33→25→9→18→36，
b: 36→31→21→1→2→4→8→16→32→23→5；
(6) a: 6→12→24→7→14→28→15→30→19→38→35，
b: 35→29→17→34→27→13→26→11→22→3→6；
(7) a: 7→14→28→15→30→19→38→35→29→17→34，
b: 34→27→13→26→11→22→3→6→12→24→7；
(8) a: 8→16→32→23→5→10→20→40→39→37→33，
b: 33→25→9→18→36→31→21→1→2→4→8；
(9) a: 9→18→36→31→21→1→2→4→8→16→32，
b: 32→23→5→10→20→40→39→37→33→25→9；
(10) a: 10→20→40→39→37→33→25→9→18→36→31，
b: 31→21→1→2→4→8→16→32→23→5→10；
(11) a: 11→22→3→6→12→24→7→14→28→15→30，
b: 30→19→38→35→29→17→34→27→13→26→11；
(12) a: 12→24→7→14→28→15→30→19→38→35→29，
b: 29→17→34→27→13→26→11→22→3→6→12；
(13) a: 13→26→11→22→3→6→12→24→7→14→28，
b: 28→15→30→19→38→35→29→17→34→27→13；
(14) a: 14→28→15→30→19→38→35→29→17→34→27，
b: 27→13→26→11→22→3→6→12→24→7→14；
(15) a: 15→30→19→38→35→29→17→34→27→13→26，
b: 26→11→22→3→6→12→24→7→14→28→15；
(16) a: 16→32→23→5→10→20→40→39→37→33→25，
b: 25→9→18→36→31→21→1→2→4→8→16；
(17) a: 17→34→27→13→26→11→22→3→6→12→24，
b: 24→7→14→28→15→30→19→38→35→29→17；
(18) a: 18→36→31→21→1→2→4→8→16→32→23，
b: 23→5→10→20→40→39→37→33→25→9→18；
(19) a: 19→38→35→29→17→34→27→13→26→11→22，
b: 22→3→6→12→24→7→14→28→15→30→19；
(20) a: 20→40→39→37→33→25→9→18→36→31→21，
b: 21→1→2→4→8→16→32→23→5→10→20。

註:粉紅色字體的數對其最後一步的組合與起始組合相同(a,b)，**黑色字體**則與起始組合相反(b,a)。

陸、研究結果

一、當推廣到多堆數目，在∑分為數堆不同數目相加時，比較其相同或相異之處。再進一步分析，部分會均分的數，在其無法均分時，將其狀況歸納整理出來。

1.當**分為 n 堆且可均分**的數對中，在形成穩定狀態的**前一步會出現(n-2)堆是(∑／n)**：

(1) 部份均分的第一個數，其均分組合倒數第二步的數對組合中，a+b+c(三堆)會有一個 4、a+b+c+d(四堆)會有兩個 4、a+b+c+d+e(五堆)會有三個 4、a+b+c+d+e+f(六堆)會有四個 4，以此類推。Ex:(1,1,1,3,14)。

(2) 部份均分的第二個數，其均分組合倒數第二步的數對組合中，a+b+c(三堆)會有一個 6、a+b+c+d(四堆)會有兩個 6、a+b+c+d+e(五堆)會有三個 6、a+b+c+d+e+f(六堆)會有四個 6，以此類推。Ex:(3,3,6,9,9)。

(3) 部份均分的第三個數，其均分組合倒數第二步的數對組合中，a+b+c(三堆)會有一個 8、a+b+c+d(四堆)會有兩個 8、a+b+c+d+e(五堆)會有三個 8、a+b+c+d+e+f(六堆)會有四個 8，以此類推。Ex: (1,2,3,14,20)。

2.**失敗的可能情形**：

(1) a+b+c(三堆)、a+b+c+d(四堆)、a+b+c+d+e(五堆)、a+b+c+d+e+f(六堆)中，**經過若干步後重複某組合而失敗，例如：當∑為 8n 時，失敗組合的過程中必含有(7,7,10)、(6,7,11)、(5,7,12)、(6,9,9)**。若出現在四堆、五堆、六堆的組合中，除了某組合的三數外的其他數字不會改變(因為是中間數，非最大數或最小數)。Ex: (1,2,7,11,19)。

(2) **當任意組合的倒數第二步最大數為最小數的 2 倍時，必定失敗**(再經一步分配後兩數互換) Ex: (2,2,8,12)。

二、任意數對組合在分配過程中，當推廣到多堆彈珠時，同時探討其奇偶數性質所存在的規律性，及與總和(∑)的關聯性。

1.任意的數對組合中之奇偶數變化有四種可能性，其過程如下:

(1) (偶數,…,偶數) → (偶數,…,偶數)；　(2) (偶數,…,奇數) → (奇數,…,偶數)；

(3) (奇數,…,偶數) → (偶數,…,奇數)；　(4) (奇數,…,奇數) → (偶數,…,偶數) 。

註:指任意的數對組合中之最大、最小數(前者較小，後者較大) 。

2.由第 1 點可觀察到：經過四種可能性可發現，偶數的數量只會變多或不變，卻不會變少。

(1) 變多的情形：(奇數,…,奇數) → (偶數,…,偶數)

(2) 不變的情形：(偶數,…,偶數) → (偶數,…,偶數)

(奇數,…,偶數) → (偶數,…,奇數)

(偶數,…,奇數) → (奇數,…,偶數)

3.由第 1 點亦可知:任意可形成穩定狀態的組合中，

倒數第二步中，最大與最小的數必須為(奇數,奇數) 或 (偶數,偶數)才會收斂。

註:因成功組合一定要全部偶數，故最大數與最小數為 (奇數,偶數) 不可能成功分配。

4.倘若過程中經過若干步後，會出現重複的情形，則從第一次出現重複數對到最後一步(即為重複數對的第二次出現)的過程中，奇、偶數的數量都會相同，不會增加或減少；因為上述情形的最大與最小數皆分別為一奇數及一偶數或都是偶數，所以不管再經過幾步分配，其奇、偶數數量都不會改變。

即(奇數,…,偶數) → (偶數,…,奇數)；(偶數,…,奇數) → (奇數,…,偶數)

5.任意可形成穩定狀態的組合中，只要有出現奇數，奇數的數量一定會有偶數個。

因此，**可成功分配的數必為偶數(因分配完後必為若干堆偶數)，而偶數堆奇數相加，也才可形成偶數。**

三、當分堆問題會收斂時，其過程是不是會有均分的情況?當有均分的情形發生時，其過程會不會收斂?總和為何時，所有狀態都能走過且不重複？

1.當有均分的情形發生時，其過程會不會收斂?

(1) 當總和為 2v 時，且 v 為奇數，除了起始值為(v,v)，其他起始值經分配後，都不會收斂。

【證明】:

第一：設起始值為(2t+1,2i+1)，而 i>t，則下次分配會變為(4t+2,2(i-t))，

因此每堆都是偶數，所以不可能收斂。

第二：如果起始值為(v,v)，下次分配會變為(2v,0)，則(v,v)就會收斂。

第三：若起始值為(2m,2l)時，也不會收斂。

(2) 均分(達穩定狀態時)一定會收斂。(此為均分與收斂之關係式)

【**證明**】:設為 n 堆時，總和為 2nr， 所以(2r,…,2r) →(4r,2r…, 2r,0)

由過程中會先達穩定狀態(2r,…,2r)，再經一次分配後變為(4r,2r…, 2r,0)，

也就是每次分配後都會回到自己時，就會收斂。

(3)若收斂時有一堆為 0，其前一步必為穩定狀態。

2.總和為多少時，所有狀態都能走過且不重複？

經推導觀察後，我們發現只有迴圈個數為 1 的奇質數，才有可能所有狀態皆走過

(因只有一條迴圈時，所有狀態皆會在此唯一的迴圈上)。

下列各數為所有狀態都能走過且不重複的奇質數(100 以內)

總和為：3、5、7、11、13、19、23、29、37、47、53、59、61、67、71、79、83

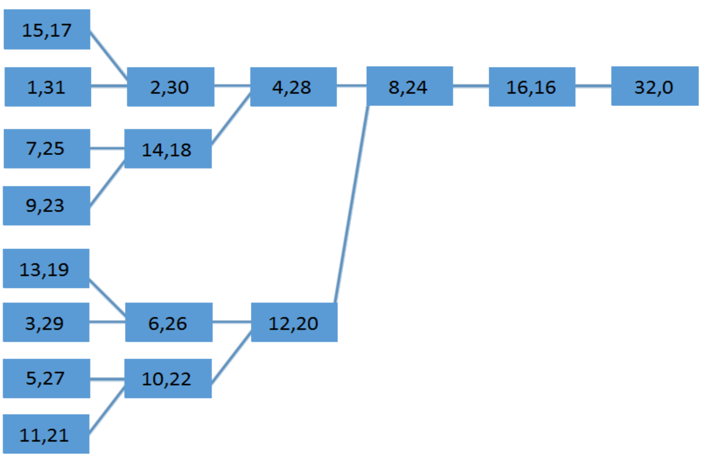
四、若不收斂，起始值為何時，會形成最大迴圈或最小迴圈? 其迴圈數會是多少？而迴圈的個數又會是多少？

以下規則只適用於無法收斂的情形中

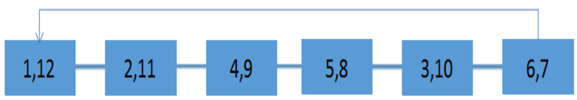
(一) 總和不同時，形成最大迴圈或最小迴圈之情形：

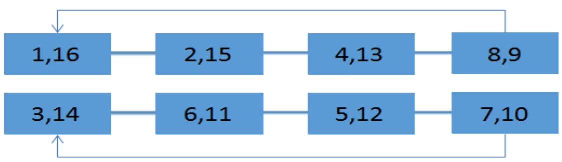
1.分為兩堆：

最大數和最小數的差為最小值的組合和最大值的兩個組合必出現在同一條路線上，若此路線不會收斂，其迴圈數會最大。



Ex:



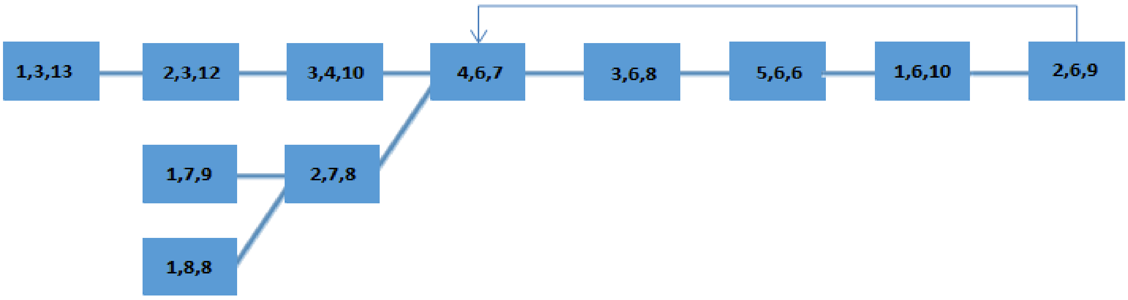


2.分為 n 堆：

(1) 如果 Σ 為奇數且非 n 的倍數時，其最大與最小數的差為最小值和最大值的兩個組合不一

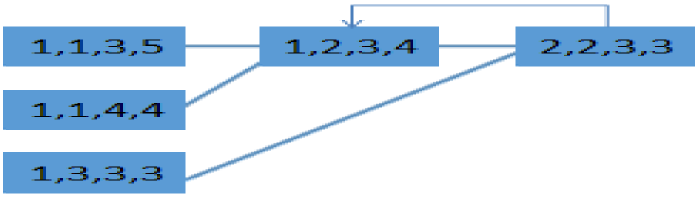
定都會出現在同一條路線。若出現在同一條路線上，其形成的迴圈會最大；若出現在不同條路線上，最大與最小數的差為最小值的組合所形成的迴圈會最大。

Ex:



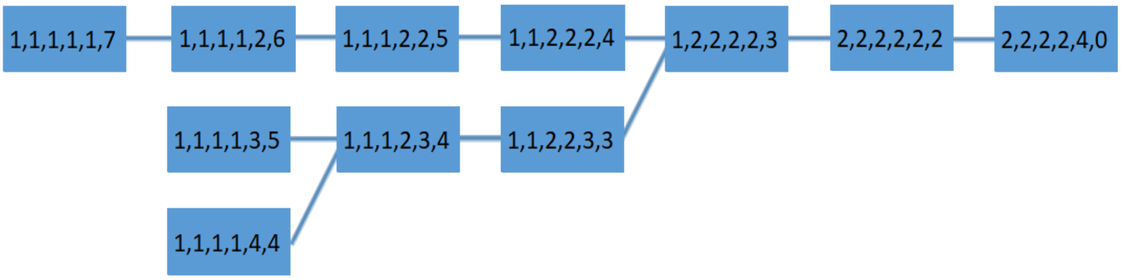
(2) 如果 Σ 為偶數且非 n 的倍數時，其最大與最小數的差為最小值和最大值的兩個組合不會出現在同一條路線，且最大與最小數的差為最小值的組合所形成的迴圈會最大。

Ex:



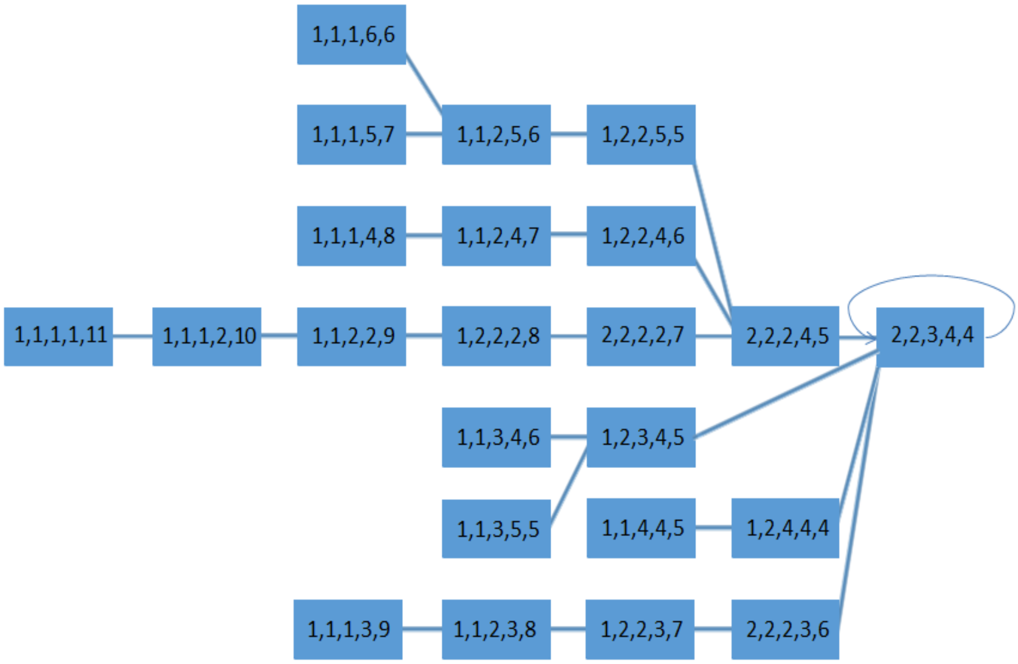
(3) 若 Σ 為 2n 的倍數時，其最大與最小數的差為最小值和最大值的兩個組合會出現在同一條路線上，但不一定是最大迴圈。

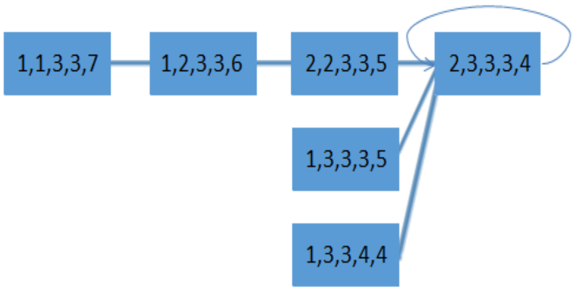
Ex:



(4) 若 Σ 為 n 的奇數倍時，其最大與最小數的差為最小值和最大值的兩個組合會出現在同一條路線，且此路線的迴圈為最大。

Ex:





(二) 若不收斂，其最大迴圈與最小迴圈的迴圈數分別是多少？

1.以下三張圖是分為兩堆且總和不同時的最大與最小迴圈數之列表:

總和	最大迴圈數	最小迴圈數
4		
5	2	
6	1	
7	3	
8		
9	3	1
10	3	2
11	5	
12		
13	6	
14		
15	4	2
16		
17	4	
18	3	1
19	9	

總和	最大迴圈數	最小迴圈數
20	2	
21	6	1
22	5	
23	11	
24	1	
25	10	2
26		
27	9	1
28	3	
29	9	
30		
31	5	
32		
33	5	1
34	4	
35	12	2

總和	最大迴圈數	最小迴圈數
36	3	
37	18	
38	9	
39	12	1
40	2	
41	10	
42	6	1
43	7	
44	6	
45	12	1
46	11	
47	23	
48	1	
49	21	3
50	10	2

2.當分為兩堆時，最大迴圈數不會超過總和的一半。

【證明】:

(1)迴圈數之最小值:當最大數為最小數的兩倍時，迴圈數為 1

(2)迴圈數之最大值:當所有狀態皆能走過且不重複時，共有(∑-1)/2 個組合，

所以迴圈數為(∑-1)/2，不超過∑/2。

(三) 以下兩張圖是總和為奇質數時，迴圈個數之列表：

總和	迴圈個數
3	1
5	1
7	1
11	1
13	1
17	2
19	1
23	1
29	1
31	3
37	1
41	2

總和	迴圈個數
43	3
47	1
53	1
59	1
61	1
67	1
71	1
73	4
79	1
83	1
89	4
97	2

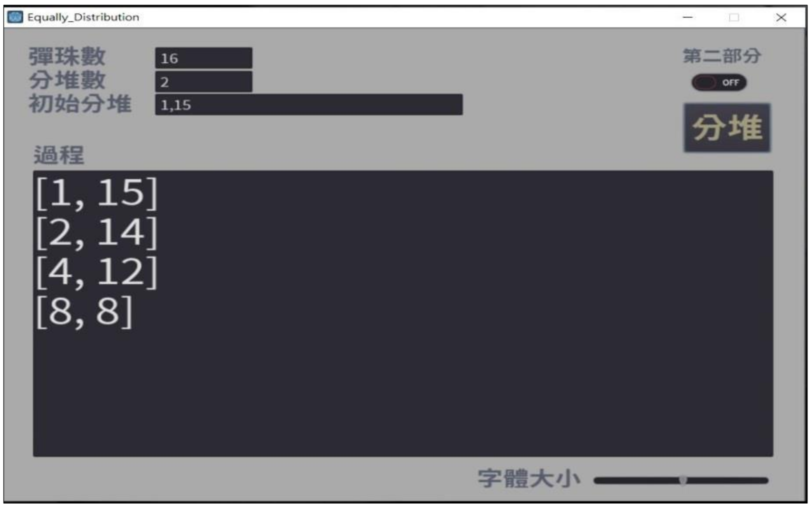
五、先利用 Excel 把數堆彈珠分配過程詳列後，再將分配過程所找出的規律運用數學邏輯並加入條件因素，寫成演算法，以程式 GDScript 動態語言寫出執行，呈現其分配過程。

1.利用程式解決第 58 屆的數學遊戲問題：

「現有兩堆彈珠，一堆是 23 個彈珠，另一堆是 9 個彈珠，如何分配讓這兩堆數目不一樣的彈珠進行平分？

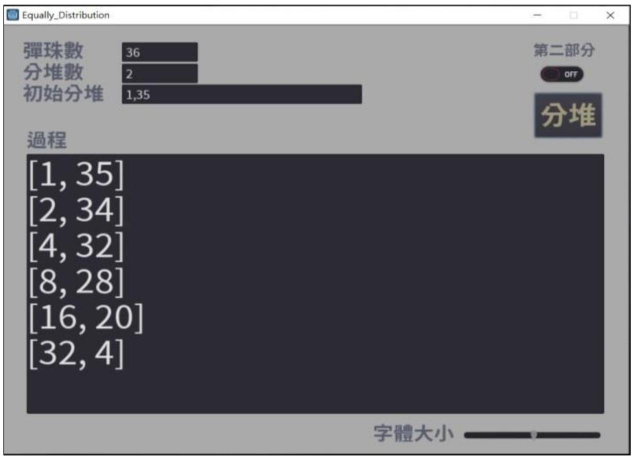
同時在每次進行分配時，都從數量較多的那一堆拿出當下較少數量那堆的個數，將其放到數量較少的那堆，反覆進行這樣的動作，直到因兩堆數量相等而無法分配。」

2.以程式驗證收斂的前一步為穩定狀態。Ex:起始組合:(1,15)



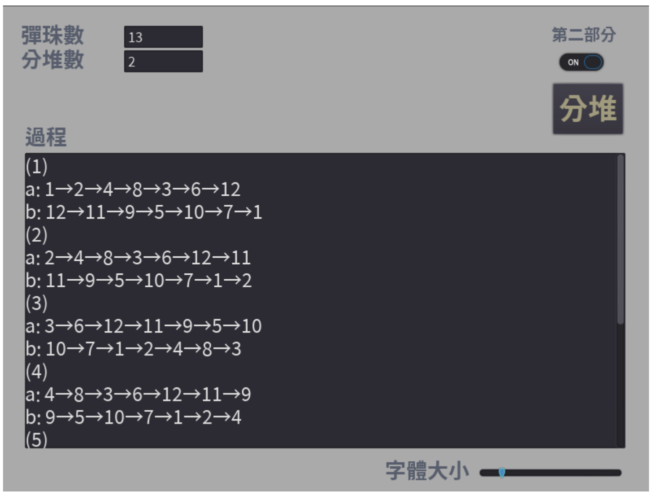
3.以程式呈現無法收斂且總和相同時，迴圈數分別為 1 和 3 時，而迴圈數為 1 時，

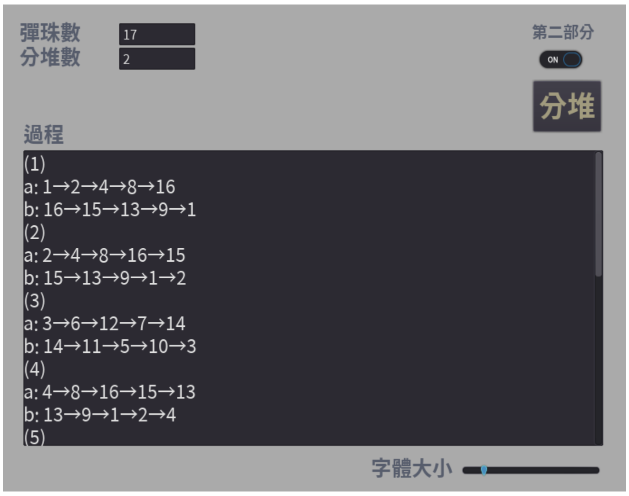
也就是最大數為最小數的兩倍之分配情形。Ex:起始組合:(1,35)和(3,33)



4.當總和為奇質數且分為 2 堆時，以程式呈現所有狀態都能走過且不重複，

及無法都走過的總和之分配情形。Ex:總和為 13、17(程式的第 2 部分)





六、分別推導分堆問題及迴圈問題，我們找到方法寫出其演算法，能快速且無遺漏地列出迴圈，以程式 GDScript 動態語言寫出執行。

【**主程式的演算法部分**】：

#如果最大堆等於最小堆，停止分堆

1.假設最小數為 a，最大數為 b

2.如果最大數等於最小數，則停止分配(當 a=b)。

3.如果最大數不等於最小數，則繼續分配(當 a≠b)。

#分堆進行

#最大堆減去最小堆

#最小堆加上自己的數量

1.假設最小數為 a，最大數為 b

2.因分配規則為最大數減去最小數，所以 b-a

3.從最大數拿最小數的值給最小數時，最小數將加上自己本身。(a+a)

柒、討論

雖然我們更深入地探究、發掘並解決相當部分的問題，而還未解決或未來可繼續研究的問題，我們做出以下的討論：

一、當∑分為數堆不同數目相加時，部分可均勻分配的數，會產生無法均勻分配時，我們歸納統整出的失敗情形有兩種：**(一)最大數為最小數的兩倍；(二)出現某特定組合後，再經過若干步數之分配後，出現重複。**然而，形成這兩種不同情形的條件為何？這還需要更多的時間去進一步觀察研討。

二、我們觀察到如果在迴圈中要所有組合都會經過且不重複，一定要分為兩堆且Σ是奇質數，然而Σ為某些奇質數時，卻不會所有組合都會經過且不重複，舉例：在 100 以內的數有 17、31、41、43、73、89、97 等共 7 個數。依循觀察其結果，我們日後可再進一步探究**「為何只有奇質數才可能會所有組合都會經過且不重複？而其他數字卻一定不可能」**

三、本作品還未找出: **(一)不同總和與其最大迴圈之關聯性；(二)起始組合與收斂之關聯性。**能否找出通式判別最大迴圈數與可否收斂？

四、原先我們設定的定義規則是「一次從最大數拿最小數的值放到最小數上面」，倘若並非這樣呢？比如說「**三個數相加時，從最大數拿[總和平均數]到最小數。**」這個方向值得之後繼續探討。

捌、結論

全國科展	組別	作品名稱	研究範疇與貢獻
第55屆	國中	從平分問題到動態穩定	將兩堆小石子利用輾轉相除法記錄移動過程，使平分問題轉化成二進位制的移位問題！
第56屆	國中	再探均分問題的動態穩定	將三堆石頭移動成數量皆相等的狀態，利用二進位制歸納三堆時的移動次數型態，找出可形成穩定狀態的條件及移動方法與次數。
第58屆	國小	公平分配遊戲	將兩堆糖果公平分配成功的情形，觀察出A+B=4n至少有一組成功解及A+B=2^n 一律成功但都缺少了一般化的證明。
第60屆	國中	分堆問題之收斂性的探討 (本作品是我們所研究的)	1.當起始值可以收斂時： (1)探究分為數堆不同數目相加時，比較其相同或相異之處，並探究奇偶數變化與總和(∑)的關聯性。 (2)部分組合可收斂的數，在無法收斂時，先將其狀況歸納整理出來，再探究穩定狀態與收斂之關聯性。 2.當起始值無法收斂時: (1)在無法收斂並形成迴圈時，歸納在固定總和下，形成最大迴圈或最小迴圈之條件，與不同總和的迴圈數之值。 (2)歸納出當總和不同時，其形成的迴圈個數分別有多少個，並以表格呈現。 (3)觀察出只有分為兩堆且總和為奇質數時，才有可能所有組合皆走過且不重複，而非奇質數一定不可能形成。 3.將分配過程中所找出的規律，運用數學邏輯並加入條件因素，寫成演算法，並以程式GDScript動態語言執行，呈現其分配過程。(第56屆沒有程式輔助)

數學的奧妙之處在於尋求其解決問題的過程中，又引發出此新的問題，讓我們發現可再對討論所列出的進行探究。此過程推導磨練我們耐力與毅力，也遇到許多困難與挫折，還要拚命想出解釋的方法與為什麼會這樣，可是，當我們試驗導出時，也就顯得格外愉悅！在我們的努力和老師們的協助下，我們成功地完成了這個試驗，清楚的解釋了為什麼，並從這次的試驗，我們不斷地假設、推導、觀察、驗證和統整歸納的過程中，我們獲得了更多的知識，從中發掘到了數學樂趣，也更能體會「團結就是力量」！仔細剖析探索，不難發現在日常生活中的每樣東西都有它的規律，相當有趣而奇妙；另外，我們也發現了分堆問題之收斂的規則與其形成最大迴圈或最小迴圈的限制，再發現每個狀態都能走過且不重複的總和都是奇質數，並從最初的如何分配達成穩定狀態，再到由推導過程中發現其奇偶數性質的規律，這三項研究主題中，我們有觀察到規律並推導出通式與判定通則，起初開始的「分配彈珠且試著讓每一堆的彈珠數量相同」這看似簡單且平凡的分堆問題，加以延伸出其他研究方向並花時間去推導，再由推導結果觀察規律，最後統整出通則或是找出通式，其實背後有許許多多有趣的地方。當遇到瓶頸無法突破時，我們試著找尋相關知識並請教老

師們，使問題的研究得以繼續進行，這正是古人所說的:「處處留心皆學問」。有時候一個人做可能非常困難，但如果是幾個同學開始分工，然後再將大家的意見還有分別觀察到的規律結合在一起，就會是更為完整的作品。除了推導出的種種研究結果外，其實「過程」也是非常寶貴的，這些都是我們日後進一步做研究的難得經驗。

玖、參考資料及其他

一、參考資料：

(一)、團康輔導社/主編，有趣的數學遊戲，出版社:文國書局

(二)、徐筠悉、林佳妤、卓品瑜、陳楷翔，第五十八屆全國科展國小組數學科「公平分配遊戲」

(三)、林建銘、高暐芬、廖昇瑋，第五十六屆全國科展國中組數學科佳作「再探均分問題的動態穩定」

(四)、陳奕均，第五十五屆全國科展國中組數學科佳作「從平分問題到動態穩定」

(五)、馬榮喜、陳世易，國中數學第四冊，康軒文教事業股份有限公司，2015

(六)、徐力行 著，沒有數字的數學，天下文化，2003

二、附錄：程式碼【第一部份:分堆問題 及 第二部份:迴圈問題】

```
extends Control
var marbles = 0
var stacks = 0
var n = 0
var distribute = []
var process = []
var temp = []
var str_result = ""
var str_a = ""
var str_b = ""
var is_distributing = false
var is_part2 = false
```

```
#畫面更新迴圈
#根據目前輸入的資料處理輸出結果
func _process(delta):
	if is_distributing and is_part2 and n < marbles/2:
		var pair = [n+1, marbles-(n+1)]
		distribute.clear()
		distribute.resize(2)
		distribute[0] = pair[0]
		distribute[1] = pair[1]
		process.clear()
		var t = [0,0]
		t[0] = pair[0]
		t[1] = pair[1]
		process.push_back(t)
		str_a = "a: " + str(distribute[0])
		str_b = "b: " + str(distribute[1])
		distribution(distribute)
		update_result()
		n = n + 1
		is_distributing = true
	elif is_distributing and not is_part2:
		update_result()
		distribution(distribute)
	else:
		is_distributing = false
		$tags/tag_is_processing.visible = false
#送出分堆資料
func send():
	$tags/tag_is_processing.visible = true
	is_distributing = true
	str_result = ""
```

```
	if not is_part2:
		part1()
	else:
		part2()

#分堆資料處理 第一部分
func part1():
	var str_array = $inputs/distribution.text.split(",", true, 0)
	distribute.resize(str_array.size())
	distribute.clear()
	for i in str_array:
		distribute.append(i.to_int())
	stacks = distribute.size()
	process.clear()
	temp.clear()
	temp.resize(stacks)
	temp = distribute.duplicate()
	temp.sort()
	process.push_back(str(temp))
	$inputs/stacks.text = str(stacks)
	marbles = 0
	for x in distribute:
		marbles = marbles + x
	$inputs/marbles.text = str(marbles)

#分堆資料處理 第二部分
func part2():
	marbles = int($inputs/marbles.text)
	stacks = 2
	$inputs/stacks.text = "2"
	n = 0
```

```
#根據輸入內容進行分堆
func distribution(var array):
        var a = array.min()
        var b = array.max()

#如果最大堆等於最小堆，停止分堆
        if a == b :
                is_distributing = false
                return

#分堆進行
#最大堆減去最小堆
#最小堆加上自己的數量
        array[array.find(b)] = (b - a)
        array[array.find(a)] = (a + a)

#根據分堆紀錄檢查是否有相同的組合
#並記錄當前的分堆
        if is_part2:
                str_a = str_a + "→" + str(array[0])
                str_b = str_b + "→" + str(array[1])
                for s in process:
                        if (s.min() == array.min()) && (s.max() == array.max()):
                                is_distributing = false
                                return
                var t = [0,0]
                t[0] = array[0]
                t[1] = array[1]
                process.push_back(t)
                distribution(array)
```

```
	else:
		temp.clear()
		temp.resize(stacks)
		temp = distribute.duplicate()
		temp.sort()
		for s in process:
			if s == str(temp):
				is_distributing = false
				update_result()
				return
		process.push_back(str(temp))

#變更資料輸入內容
func change_mode():
	is_part2 = not is_part2
	is_distributing = false
	if is_part2:
		$inputs/distribution.visible = false
		$tags/tag_stack_distribute.visible = false
		stacks = 2
		$inputs/stacks.text = "2"
	else:
		$inputs/distribution.visible = true
		$tags/tag_stack_distribute.visible = true
#更新資料輸出結果
func update_result():
	if is_part2:
		str_result = str_result + "(" + str(n+1) + ")\n" +str_a + "\n" + str_b + "\n\n"
	else:
		str_result = str_result + String(distribute) + "\n"
	$display.text = str_result
```

【評語】030407

1. 此作品研究分堆問題的收斂性，考慮兩堆個數不等的彈珠，每次都從較多個數那堆拿較少個數那堆數量的彈珠放至較少個數的那堆，反覆操作直到兩堆數量相等或形成循環迴圈。作者更進一步考慮 n 堆彈珠的分堆問題。已有先前的科展作品(第 56 屆國中組：再探均分問題的動態穩定)探討 n 堆彈珠達到穩定狀態的一般式，本件作品探討起始彈珠數之總和固定時，無法達到穩定狀態，形成最大迴圈或最小迴圈的條件及其迴圈數。此外，在總和為定值時，分配過程中所有組合都能經過且不重複，探討起始值的條件。

2. 此作品的研究成果運用數學邏輯並加入條件因素，寫成演算法，利用程式 GDScript 動態語言執行，並呈現其分配過程及驗方法是一大特色。作者善於利用程式計算出大量資料，觀察數據歸納出許多條件準則，美中不足的是論證的部分。

作品海報

摘 要

此作品研究「在n堆彈珠數目的分堆問題，當每堆個數不同並進行分配，每次都從最多個數那堆拿最少個數那堆的數目給最少個數的那堆，直到收斂或出現迴圈時，則停止分配彈珠。」首先，探究分配過程中，奇偶數變化與總和(∑)的關聯性，進而觀察出可達成穩定狀態(收斂)之條件的規律；而當無法達成穩定狀態時，起始彈珠數之總和固定，在甚麼條件下會形成最大迴圈或最小迴圈，其迴圈數是多少，而迴圈的個數又會是多少；並探討起始值在甚麼條件下，其分配過程中，每個組合都能經過且不重複；最後，將分配過程所找出的規律，運用數學邏輯並加入條件因素，寫成演算法，利用程式GDScript動態語言執行，並呈現其分配過程及驗證我們的方法。

壹、研究動機

1. 參考科展作品(第58屆)的數學問題：「現有兩堆彈珠，一堆是23個彈珠，另一堆是9個彈珠，如何分配讓這兩堆數目不一樣的彈珠進行平分？」
2. 從全國中小學科展網站參考相關系列 (第 55 & 56 屆) 的資料，並使用其中一種分配方法，並延伸到多堆。
3. 藉由閱讀『有趣的數學遊戲』與『沒有數字的數學』，且書籍提及迴圈 (Circle)，相關知識引發了我們高度的興趣。
4. 在歷屆全國科展作品之評審評語建議在作品內容有提升空間，使得這樣的結果仍有很大的研究性價值，進而我們將公平分配遊戲推廣到多堆並做更為深入的探究。

貳、研究目的

有數堆彈珠，其個數不一樣(a+b+c+……=∑)進行分配，每次都從最多個數那堆拿最少個數那堆的數目給最少個數的那堆，直到收斂或出現迴圈時，則停止分配彈珠。

一、分為兩堆：

1.起始組合要有什麼條件?此分堆問題會收斂，而當起始值為何時，能每一個狀態都走過?

2.當分堆問題會收斂時，其過程是不是會有均分的情況?當有均分的情形發生時，其過程會不會收斂?

3.若不收斂，起始值為何時，會形成最大迴圈或最小迴圈? 其迴圈數會是多少？迴圈的個數又會是多少？

二、分為三堆、四堆到n堆:

1.起始組合要有什麼條件?此分堆問題會收斂，而當起始值為何時，能每一個狀態都走過?

2.若收斂且過程中有均分的情形時，起始值要符合甚麼條件?當有均分的情形發生時，其過程會不會收斂?

3.若不收斂，起始值為何時，會形成最大迴圈或最小迴圈? 其迴圈數會是多少？迴圈的個數又會是多少？

三、將分配過程所找出的規律，運用數學邏輯並加入條件因素，以程式GDScript動態語言寫出執行，並呈現其分配過程。

參、研究設備及器材

一、筆記本、筆(記錄用)　二、電腦、自備隨身碟　三、Microsoft Excel 2010　四、Godot Engine合法免費授權軟體

肆、文獻探討

做本研究之前，我們先參考了一些別人的文獻資料：

屆數	組別	作品名稱	研究方法	得獎情形
55	國中組	從平分問題到動態穩定	輾轉相除法及二進位制	佳作
56	國中組	再探均分問題的動態穩定	二進位制	佳作
58	國小組	公平分配遊戲	公因數	

表一：文獻資料

伍、研究過程與方法

研究期間，分成了不少階段及不同的突破，如下:

時間	進展
108.09	發現將兩堆糖果公平分配成功的情形，並開始研究「公平分配遊戲」(二)
108.10	發現將兩堆小石子平分問題轉成移位問題，並著手研究「從平分問題到動態穩定」(四)
108.11	發現將三堆石頭移動成數量皆相等的狀態，並著手研究「再探均分問題的動態穩定」(三)
108.12 109.01	嘗試推廣到四堆、五堆及六堆彈珠甚至 n 堆彈珠時，進行分類探討並找出各種情形在達成穩定狀態的規律，並嘗試證明第 56 屆推導出的一般式。
109.02 109.03	探討在推廣到 n 堆彈珠時，當達成穩定狀態的規律下，同時討論奇偶數變化的規律性與總和(∑)之關聯性；嘗試利用分組拆法來進行分配，並觀察其中規律。
109.04	將分配過程所導出的規律，運用數學邏輯並加入條件因素，以程式 GDScript 動態語言寫出執行，並予以呈現。
109.05	探討總和固定下，在何種情況會收斂；研究達成穩定狀態時會不會收斂並加以證明。
109.06	觀察總和固定下，起始值為何時，會形成最大迴圈或最小迴圈？其迴圈數會是多少？而迴圈個數又會是多少？與總和是多少時，每個狀態都能走過且不重複？

表二:研究進度概要

一、名詞定義:

1.分堆與分配的定義：

(1) 所有任意組合，且不論分為幾堆，皆從數對組合之最大的數中拿出最小數的值給最小數，且一次只分配一個最大數和一個最小數，直到收斂或形成迴圈時，則停止分配。

Ex:(2,2,8) →(4,2,6) →(4,4,4) → (8,4,0) →成功。

(2) 倘若不依循規則，起始組合一次分配數個最小數時，則最大數可能會變為負數，由此可知，要依循規則。

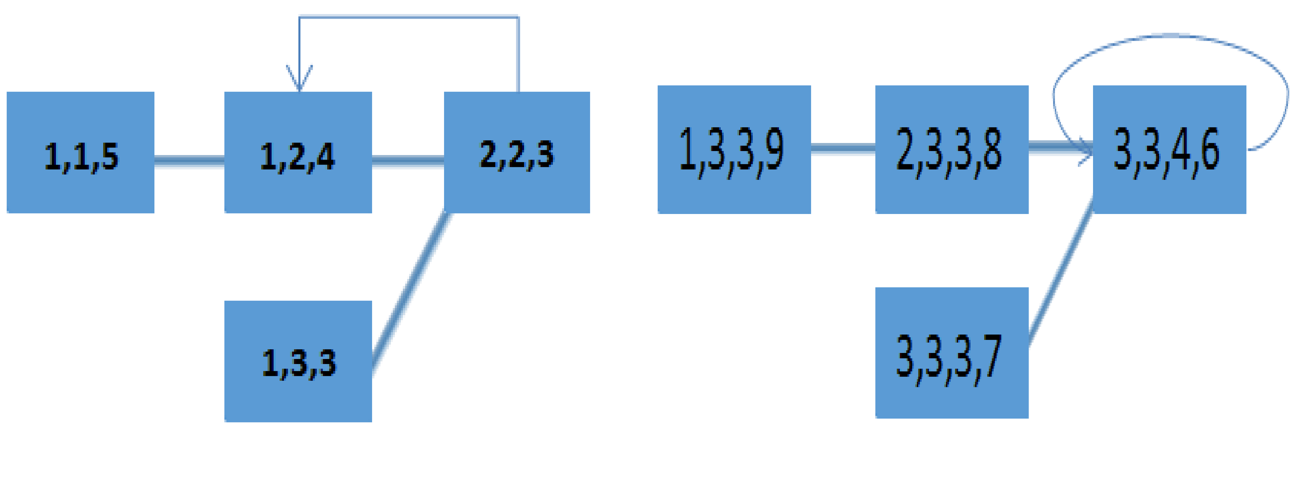
Ex: (2,2,2,2,7) →(4,4,4,4,-1) →出現負數，無解。

2.穩定狀態(均分)的定義：

將彈珠分為n堆時，每堆分別有a,b,c,d,……個，若每堆都相等，則數對(a,b,c,d,……)稱為「穩定狀態」。

3.迴圈的定義：

指任意組合經過若干次分配後出現重複，無論再分配幾次，都會回到同一個組合。



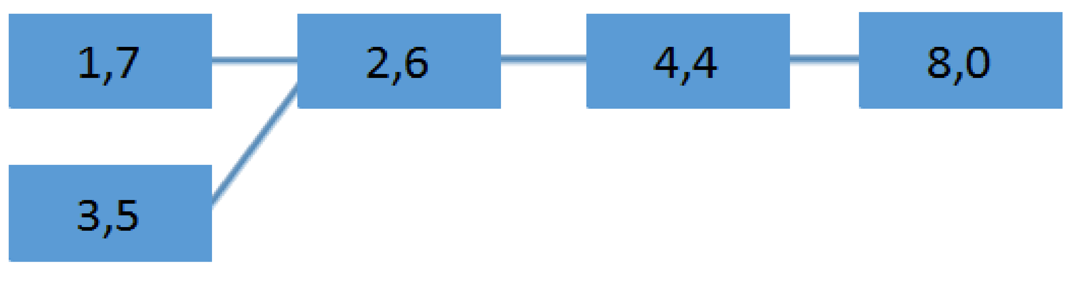
4.迴圈數：

迴圈內第一次出現某組合到第二次出現重複之間經過的分配次數。

5.迴圈個數：在同總和下，出現的所有不同迴圈之個數。

6.收斂的定義：

當組合的迴圈數為1時，也就是每次分配都回到自己本身。(包括一數為0的狀況及最大數為最小數的兩倍)



二、可均分的數字總和(∑)或不可均分的數字總和(∑)，其分別會出現的規律與產生相對應的狀況是什麼？當推廣到多堆數目時，在達穩定狀態下，若將∑分為數堆不同數目相加時，會有相同或相異的點嗎？部分可均勻分配的數，遇到什麼狀況會無法均勻分配？

1.分為四堆，∑為8、16、24、32時，能否形成穩定狀態的情形:

8	16	24	32	
所有組合皆可成功 所有數對倒數第二步的比例皆為 1:2:2:3 Ex:(1,1,1,5)	大部分皆成功，而失敗的數對倒數第二步中，最大數為最小數的兩倍，故無法形成穩定狀態。 Ex:(1,3,3,9)	大部分皆失敗，而成功的數對中，倒數第二步比例皆為 1:2:2:3 Ex:(3,3,3,15)	成功與失敗的組合約各占一半。	
			成功的組合	失敗的組合
			成功的組合中，倒數第二步比例皆為 1:2:2:3 Ex:(1,3,4,24)	有兩種情況: 1.倒數第二步中，最大數為最小數的兩倍 Ex:(1,3,6,22) 2.某個固定的組合，出現一次過後，經若干步後又重複出現。 Ex:(1,3,7,21)

以提出公因數的方法，列出關係式，並找出各數的所有分配情形:

16		
(a,b,c,d)	關係式	說明
2A+2B+2C+2D	A+B+C+D=8	全部成功
4A+4B+4C+4D	A+B+C+D=4	只有一組解
8A+8B+8C+8D	A+B+C+D=2	無正整數解
16A+16B+16C+16D	A+B+C+D=1	無正整數解
24		
(a,b,c,d)	關係式	說明
2A+2B+2C+2D	A+B+C+D=12	全部失敗
3A+3B+3C+3D	A+B+C+D=8	全部成功
4A+4B+4C+4D	A+B+C+D=6	全部失敗
6A+6B+6C+6D	A+B+C+D=4	只有一組解
8A+8B+8C+8D	A+B+C+D=3	無正整數解
12A+12B+12C+12D	A+B+C+D=2	無正整數解
24A+24B+24C+24D	A+B+C+D=1	無正整數解
32		
(a,b,c,d)	關係式	說明
2A+2B+2C+2D	A+B+C+D=16	部分成功
4A+4B+4C+4D	A+B+C+D=8	全部成功
8A+8B+8C+8D	A+B+C+D=4	只有一組解
16A+16B+16C+16D	A+B+C+D=2	無正整數解
32A+32B+32C+32D	A+B+C+D=1	無正整數解

2.分為五堆，∑為10、20、30、40時，能否形成穩定狀態的情形:

10	20	30	40	
所有組合皆可成功 所有數對倒數第二步的比例皆為 1:2:2:2:3 Ex:(1,1,1,3 ,4)	大部分皆成功，而失敗的數對倒數第二步中，最大數為最小數的兩倍，故無法形成穩定狀態。 Ex:(2,2,3,3,10)	大部分皆失敗，而成功的數對中，倒數第二步比例皆為 1:2:2:2:3 Ex:(3,3,6,9,9)	成功與失敗的組合約各占一半。	
			成功的組合	失敗的組合
			成功的組合中，倒數第二步比例皆為 1:2:2:2:3 Ex:(1,1,2,2,34)	有兩種情況: 1.倒數第二步中，最大數為最小數的兩倍 Ex:(1,1,3,6,29) 2.某個固定的組合，出現一次過後，經若干步後又重複出現。 Ex:(1,2,7,10,20)

以提出公因數的方法，列出關係式，並找出各數的所有分配情形:

20		
(a,b,c,d,e)	關係式	說明
2A+2B+2C+2D+2E	A+B+C+D+E=10	全部成功
4A+4B+4C+4D+4E	A+B+C+D+E=5	只有一組解
5A+5B+5C+5D+5E	A+B+C+D+E=4	無正整數解
10A+10B+10C+10D+10E	A+B+C+D+E=2	無正整數解
20A+20B+20C+20D+20E	A+B+C+D+E=1	無正整數解
30		
(a,b,c,d,e)	關係式	說明
2A+2B+2C+2D+2E	A+B+C+D+E=15	全部失敗
3A+3B+3C+3D+3E	A+B+C+D+E=10	全部成功
5A+5B+5C+5D+5E	A+B+C+D+E=6	全部失敗
6A+6B+6C+6D+6E	A+B+C+D+E=5	只有一組解
10A+10B+10C+10D+10E	A+B+C+D+E=3	無正整數解
15A+15B+15C+15D+15E	A+B+C+D+E=2	無正整數解
30A+30B+30C+30D+30E	A+B+C+D+E=1	無正整數解
40		
(a,b,c,d,e)	關係式	說明
2A+2B+2C+2D+2E	A+B+C+D+E=20	部分成功
4A+4B+4C+4D+4E	A+B+C+D+E=10	全部成功
5A+5B+5C+5D+5E	A+B+C+D+E=8	全部失敗
8A+8B+8C+8D+8E	A+B+C+D+E=5	只有一組解
10A+10B+10C+10D+10E	A+B+C+D+E=4	無正整數解
20A+20B+20C+20D+20E	A+B+C+D+E=2	無正整數解
40A+40B+40C+40D+40E	A+B+C+D+E =1	無正整數解

3.分為六堆，∑為12、24、36、48時，能否形成穩定狀態的情形:

12	24	36	48	
所有組合皆可成功 所有數對倒數第二步的比例皆為 1:2:2:2:2:3 Ex:(1,1,1,1,1,7)	大部分皆成功，而失敗的數對倒數第二步中，最大數為最小數的兩倍，故無法形成穩定狀態。 Ex: (1,1,3,3,6,10)	大部分皆失敗，而成功的數對中，倒數第二步比例皆為 1:2:2:2:2:3 Ex:(3,3,3,3,3,21)	成功與失敗的組合約各占一半。	
			成功的組合	失敗的組合
			成功的組合中，倒數第二步比例皆為 1:2:2:2:2:3 Ex:(1,1,1,2,16,27)	有兩種情況: 1.倒數第二步中，最大數為最小數的兩倍 Ex:(1,1,1,3,3,39) 2.某個固定的組合，出現一次過後，經若干步後又重複出現。 Ex:(1,1,1,6,11,28)

Ex: 以提出公因數的方法，列出關係式，並找出各數的所有分配情形:

24		
(a,b,c,d,e,f)	關係式	說明
2A+2B+2C+2D+2E+2F	A+B+C+D+E+F=12	全部成功
3A+3B+3C+3D+3E+3F	A+B+C+D+E+F=8	全部失敗
4A+4B+4C+4D+4E+4F	A+B+C+D+E+F=6	只有一組解
6A+6B+6C+6D+6E+6F	A+B+C+D+E+F=4	無正整數解
8A+8B+8C+8D+8E+8F	A+B+C+D+E+F=3	無正整數解
12A+12B+12C+12D+12E+12F	A+B+C+D+E+F=2	無正整數解
24A+24B+24C+24D+24E+24F	A+B+C+D+E+F=1	無正整數解
36		
(a,b,c,d,e,f)	關係式	說明
2A+2B+2C+2D+2E+2F	A+B+C+D+E+F=18	全部失敗
3A+3B+3C+3D+3E+3F	A+B+C+D+E+F=12	全部成功
4A+4B+4C+4D+4E+4F	A+B+C+D+E+F=9	全部失敗
6A+6B+6C+6D+6E+6F	A+B+C+D+E+F=6	只有一組解
9A+9B+9C+9D+9E+9F	A+B+C+D+E+F=4	無正整數解
12A+12B+12C+12D+12E+12F	A+B+C+D+E+F=3	無正整數解
18A+18B+18C+18D+18E+18F	A+B+C+D+E+F=2	無正整數解
36A+36B+36C+36D+36E+36F	A+B+C+D+E+F=1	無正整數解
48		
(a,b,c,d,e,f)	關係式	說明
2A+2B+2C+2D+2E+2F	A+B+C+D+E+F=24	部分成功
3A+3B+3C+3D+3E+3F	A+B+C+D+E+F=16	全部失敗
4A+4B+4C+4D+4E+4F	A+B+C+D+E+F=12	全部成功
6A+6B+6C+6D+6E+6F	A+B+C+D+E+F=8	全部失敗
8A+8B+8C+8D+8E+8F	A+B+C+D+E+F=6	只有一組解
12A+12B+12C+12D+12E+12F	A+B+C+D+E+F=4	無正整數解
16A+16B+16C+16D+16E+16F	A+B+C+D+E+F=3	無正整數解
24A+24B+24C+24D+24E+24F	A+B+C+D+E+F=2	無正整數解
48A+48B+48C+48D+48E+48F	A+B+C+D+E+F=1	無正整數解

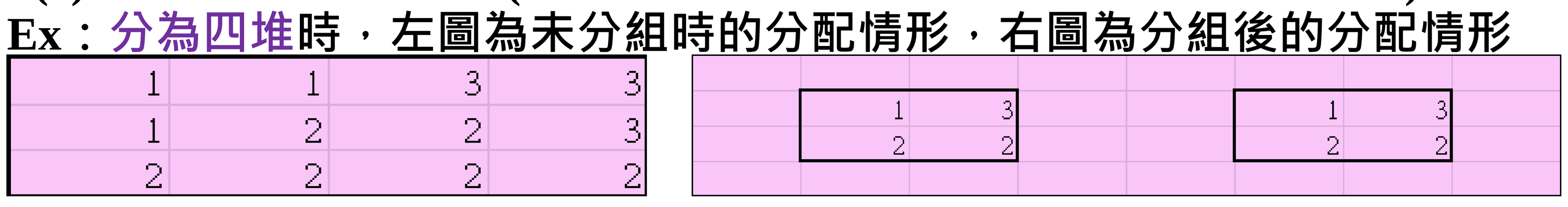
三、分組拆法：

定義:在分為多堆時，將堆數分成為若干個小組後，再進行分配。

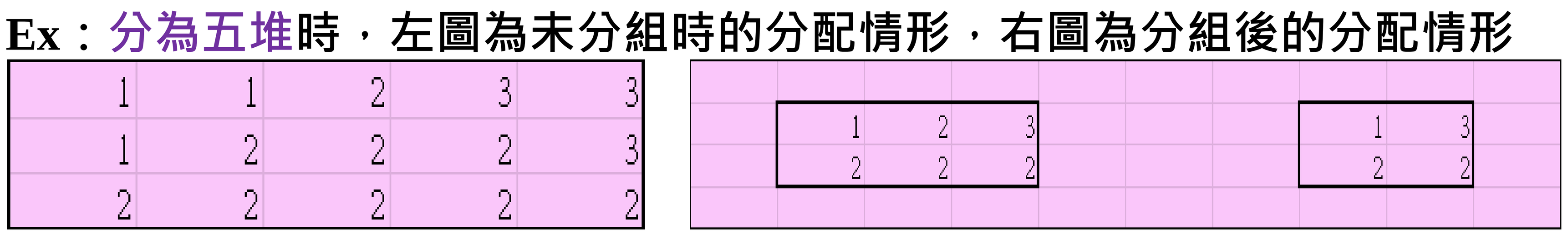
例:分為四堆→分為兩堆、兩堆

1. 分為四堆以上時，才可使用分組拆法，可將分配步驟減短。
2. 若分為x個小組，其步驟會變為分組前的1/x 。
3. 可使用分組拆法的起始組合之情形:
   (1) 最大數與最小數之數量相同。
   (2) 所有數成等差數列時(並非所有等差數列都可使用，只限於公差為2)。

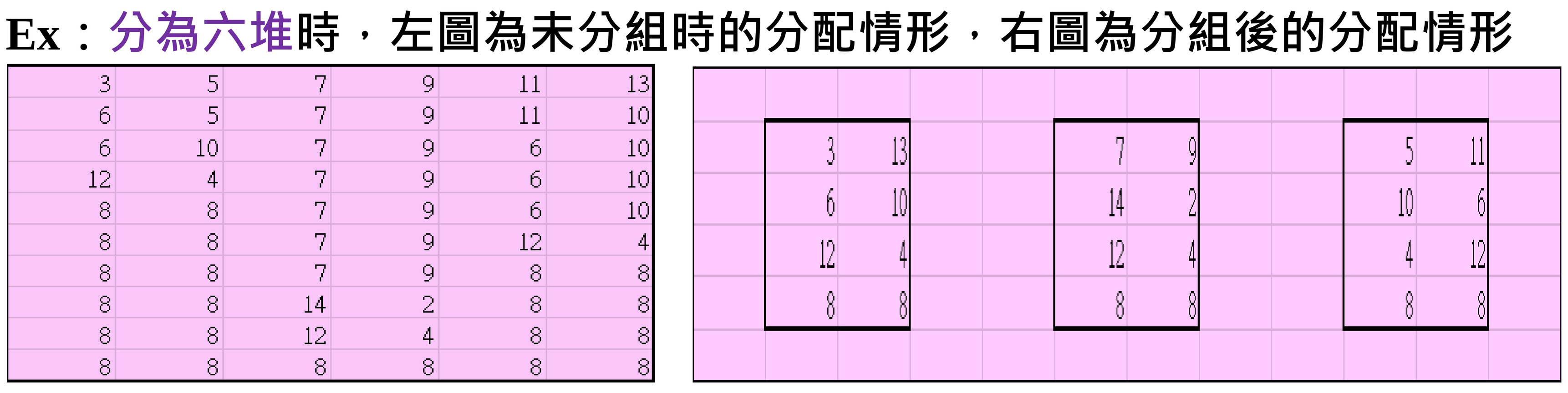
Ex：分為四堆時，左圖為未分組時的分配情形，右圖為分組後的分配情形



Ex：分為五堆時，左圖為未分組時的分配情形，右圖為分組後的分配情形



Ex：分為六堆時，左圖為未分組時的分配情形，右圖為分組後的分配情形



四、通式: $\sum=n\times p\times 2^k$ (此通式為第56屆所列出的，但其缺少證明)

設組合中的數堆彈珠數量總和為∑，且分為n堆相加，

其中p為正奇數(1≤p)，而1≤k。

(一) 若某總和能分解成此通式的形式，那麼此總和至少會有一組以上的數對會收斂。

1. 全部組合皆成功的數，分解後p一定為1；
2. 部分組合成功的數，分解後p不一定為1；
3. 當∑為2n乘上一個奇質數的倍數時，p必為此奇質數的倍數；

   Ex: ∑=a+b+c+d+e+f+g+h=48(16×3)

   依通式分解為$48=8\times3\times2^1$，因48為16的3倍，所以p為3的倍數。
4. 當2n乘上2的次方倍時，p等於1(依照通式分解後，p一定是1)。

【證明】：

有 n 堆彈珠的總和(a+b+c+d+……=∑)

若將 a+b+c+d+……分解成 $1:\underbrace{2:2:\cdots\cdots:2}_{(n-2)}:3$ 的比例時，必成功。

令 a=m 、b=2m 、c=2m……、最後一堆=3m

a+b+c+d+……=m+2m‧(n-2)+3m=2m‧n=∑(1≤m ，$m\in\mathbb{N}$)有兩種情況：

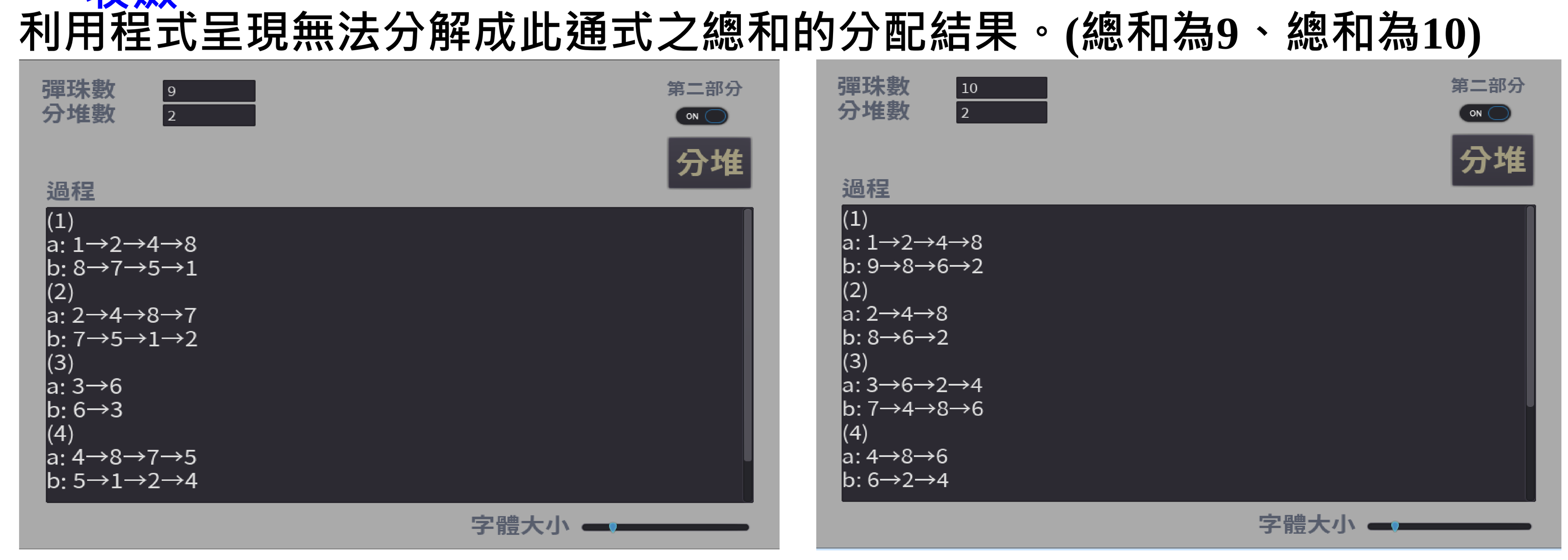
情況(1): m 為奇數，2×m×n

情況(2): m 為偶數，所以可將m 分解為 $2^k\times p$(奇數)

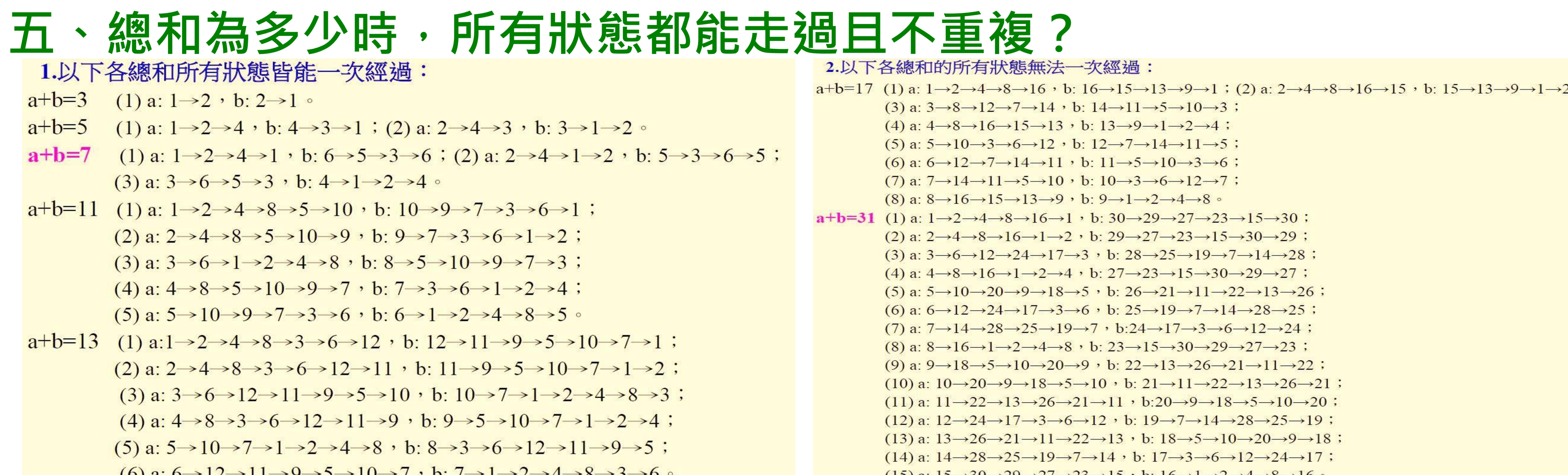
由情況(1)、(2)可推得；通式$\sum=n\times p\times 2^k$，且 1≤k，k 為自然數。

(二) 若某總和不能分解成此通式的形式，那麼此總和的所有數對都不會收斂。

利用程式呈現無法分解成此通式之總和的分配結果。(總和為9、總和為10)



五、總和為多少時，所有狀態都能走過且不重複？



陸、研究結果

一、當推廣到多堆數目，在∑分為數堆不同數目相加時，比較其相同或相異之處。再進一步分析，部分會均分的數，在其無法均分時，將其狀況歸納整理出來。

當**分為n堆且可形成穩定狀態**的數對中，在形成穩定狀態的**前一步會出現(n-2)堆是(∑ / n)，失敗的可能情形為：**

(1) a+b+c(三堆)、a+b+c+d(四堆)、a+b+c+d+e(五堆)、a+b+c+d+e+f(六堆)中，經過若干步後重複某組合而失敗的必含有(7,7,10)、(6,7,11)、(5,7,12)、(6,9,9) 。若出現在四堆、五堆、六堆的組合中，除了某組合的三數外的其他數字不會改變(因為是中間數，非最大數或最小數)。Ex: (1,2,7,11,19)。

(2) 當任意組合的倒數第二步最大數為最小數的 2 倍時，必定失敗(再經一步分配後兩數互換) Ex: (2,2,8,12)。

二、任意數對組合在分配過程中，當推廣到多堆彈珠時，同時探討其奇偶數性質所存在的規律性，及與總和(∑)的關聯性。

1.任意的數對組合中之奇偶數變化有四種可能性，其過程如下:

(1)(偶數,…,偶數) → (偶數,…,偶數)；(2)(偶數,…,奇數) → (奇數,…,偶數)；(3)(奇數,…,偶數) → (偶數,…,奇數)；(4)(奇數,…,奇數) → (偶數,…,偶數)。

2.由第1點可觀察到：經過四種可能性可發現，偶數的數量只會變多或不變，卻不會變少。

(1) 變多的情形：(奇數, …,奇數) → (偶數,…,偶數)

(2) 不變的情形：(偶數,…,偶數) → (偶數,…,偶數)；
(奇數,…,偶數) → (偶數,…,奇數)；
(偶數,…,奇數) → (奇數,…,偶數)

3.由第1點亦可知:任意可形成穩定狀態的組合中，**倒數第二步中，最大與最小的數必須為(奇數,…,奇數)或(偶數,…,偶數)才可成功分配。**

註:因成功組合一定全部偶數，故最大數與最小數為(奇數,…,偶數)不可能成功分配。

4.倘若過程中出現了經過若干步後，會出現重複的情形，則從第一次出現重複數對到最後一步(即為重複數對的第二次出現)的過程中，奇、偶數的數量都會相同，不會增加或減少；因為上述情形的最大與最小數皆分別為一奇數及一偶數或都是偶數，所以不管再經過幾步分配，其奇、偶數數量都不會改變。

即(奇數,…,偶數) → (偶數,…,奇數)；(偶數,…,奇數) → (奇數,…,偶數)

5.任意可形成穩定狀態的組合中，只要有出現奇數，奇數的數量一定會有偶數個。因此，**可成功分配的數必為偶數(因分配完後必為若干堆偶數)，而偶數堆奇數相加，也才可形成偶數。**

三、當分堆問題會收斂時，其過程是不是會有均分的情況?當有均分的情形發生時，其過程會不會收斂?總和為何時，所有狀態都能走過且不重複？

1.當有均分的情形發生時，其過程會不會收斂?

(1) 當分為兩堆，總和為2v時，且v為奇數，除了起始值為(v,v)，其他起始值經分配後，都不會收斂。

【證明】：

第一：設起始值為(2t+1,2i+1)，而i>t，則下次分配會變為(4t+2,2(i-t))，因此每堆都是偶數，所以不可能收斂。

第二：如果起始值為(v,v)，下次分配會變為(2v,0)，則(v,v)就會收斂。

第三：若起始值為(2m,2l)時，也不會收斂。

(2)均分(達穩定狀態時)一定會收斂。(此為均分與收斂之關係式)

【證明】：設為n堆時，總和為2nr， 所以(2r,…,2r) →(4r,2r…, 2r,0)

由過程中會先達穩定狀態(2r,…,2r)，再經一次分配後變為(4r,2r…, 2r,0)，也就是每次分配後都會回到自己時，就會收斂。

(3)若收斂時有一堆為0，其前一步必為穩定狀態。

2.總和為多少時，所有狀態都能走過且不重複？

經推導觀察後，我們發現只有分為兩堆，且迴圈個數為1的奇質數，才有可能所有狀態皆走過(因只有一條迴圈時，所有狀態皆會在此唯一的迴圈上)。下列各數為**所有狀態都能走過且不重複的奇質數**(100以內)

總和為：3、 5、 7、11、13、19、23、29、37、47、53、59、61、67、71、79、83。

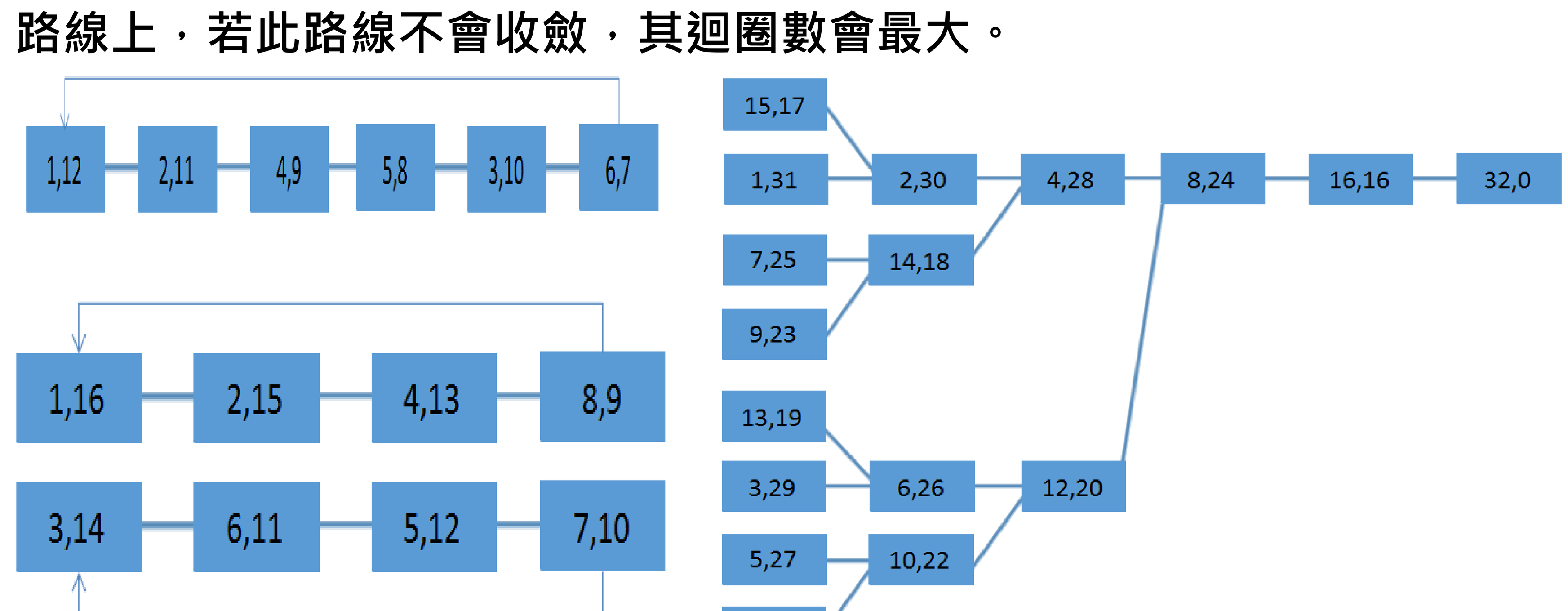
四、若不收斂，起始值為何時，會形成最大迴圈或最小迴圈？其迴圈數會是多少？而迴圈的個數又會是多少？

以下規則**只適用於無法收斂**的情形中

(一) 總和不同時，形成最大迴圈或最小迴圈之情形：

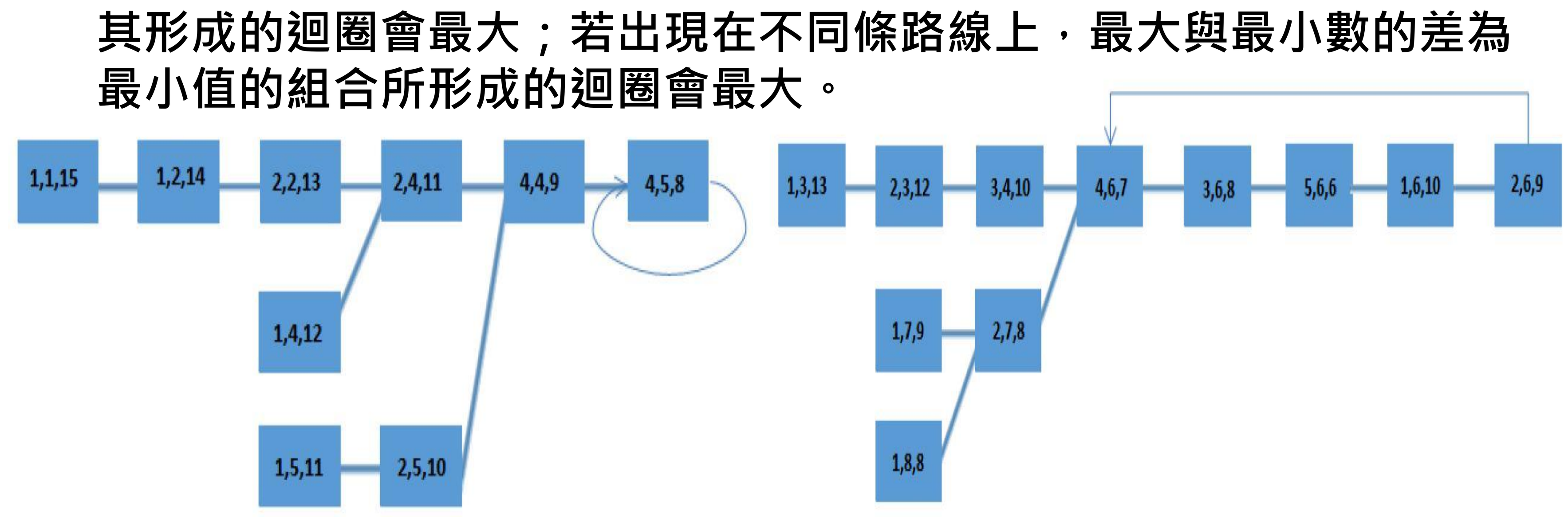
1.分為兩堆：

最大數和最小數的差為最小值的組合和最大值的兩個組合必出現在同一條路線上，若此路線不會收斂，其迴圈數會最大。

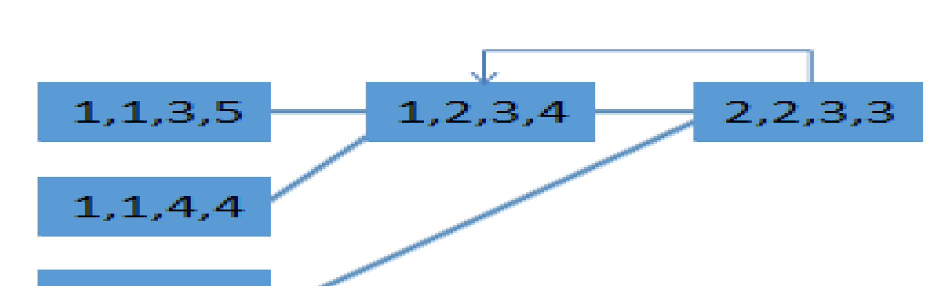


2.分為n堆：

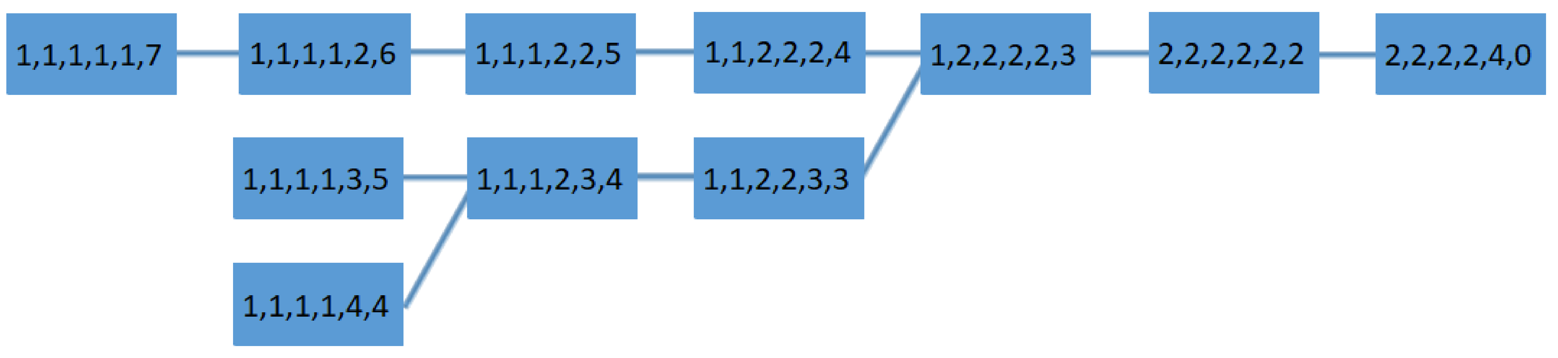
(1) 如果Σ為奇數且非n的倍數時，其最大與最小數的差為最小值和最大值的兩個組合不一定都會出現在同一條路線。若出現在同一條路線上，其形成的迴圈會最大；若出現在不同條路線上，最大與最小數的差為最小值的組合所形成的迴圈會最大。



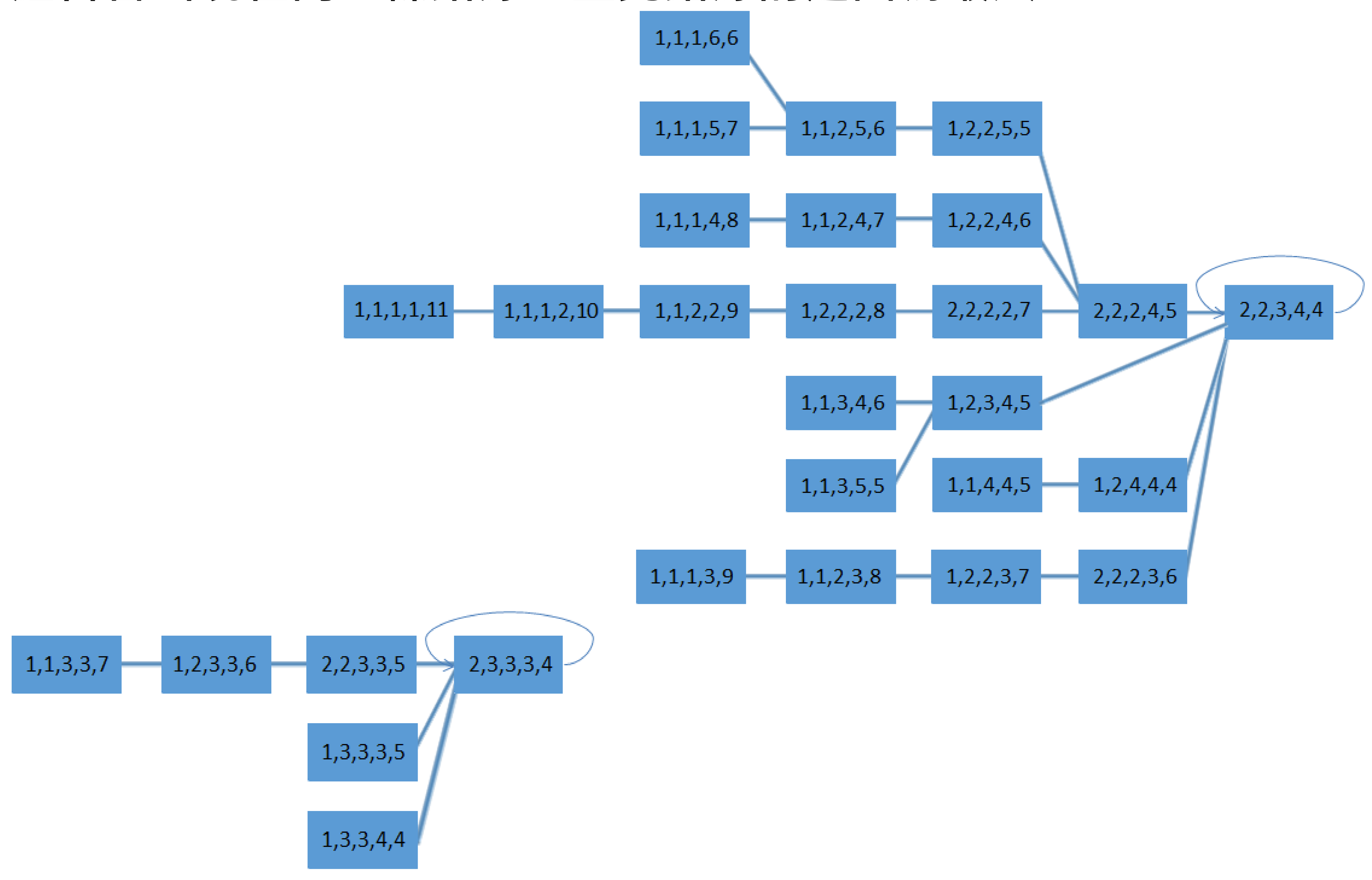
(2) 如果Σ為偶數且非n的倍數時，其最大與最小數的差為最小值和最大值的兩個組合不會出現在同一條路線，且最大與最小數的差為最小值的組合所形成的迴圈會最大。



(3) 若Σ為2n的倍數時，其最大與最小數的差為最小值和最大值的兩個組合會出現在同一條路線上，但不一定是最大迴圈。



(4) 若Σ為n的奇數倍時，其最大與最小數的差為最小值和最大值的兩個組合會出現在同一條路線，且此路線的迴圈為最大。



(二) 若不收斂，其最大迴圈與最小迴圈的迴圈數分別是多少？

1.以下三張圖是分為兩堆且總和不同時的最大與最小迴圈數之列表:

總和	最大迴圈數	最小迴圈數
4		
5	2	
6	1	
7	3	
8		
9	3	1
10	3	2
11	5	
12		
13	6	
14		
15	4	2
16		
17	4	
18	3	1
19	9	

總和	最大迴圈數	最小迴圈數
20	2	
21	6	1
22	5	
23	11	
24	1	
25	10	2
26		
27	9	1
28	3	
29	9	
30		
31	5	
32		
33	5	1
34	4	
35	12	2

總和	最大迴圈數	最小迴圈數
36	3	
37	18	
38	9	
39	12	1
40	2	
41	10	
42	6	1
43	7	
44	6	
45	12	1
46	11	
47	23	
48	1	
49	21	3
50	10	2

2.當分為兩堆時，最大迴圈數不會超過總和的一半。

【證明】:

(1)迴圈數之最小值:當最大數為最小數的兩倍時，迴圈數為1

(2)迴圈數之最大值:當所有狀態皆能走過且不重複時，共有(∑-1)/2個組合，所以迴圈數為(∑-1)/2，不超過∑/2。

(三) 右下兩張圖是總和為奇質數時，迴圈個數之列表：

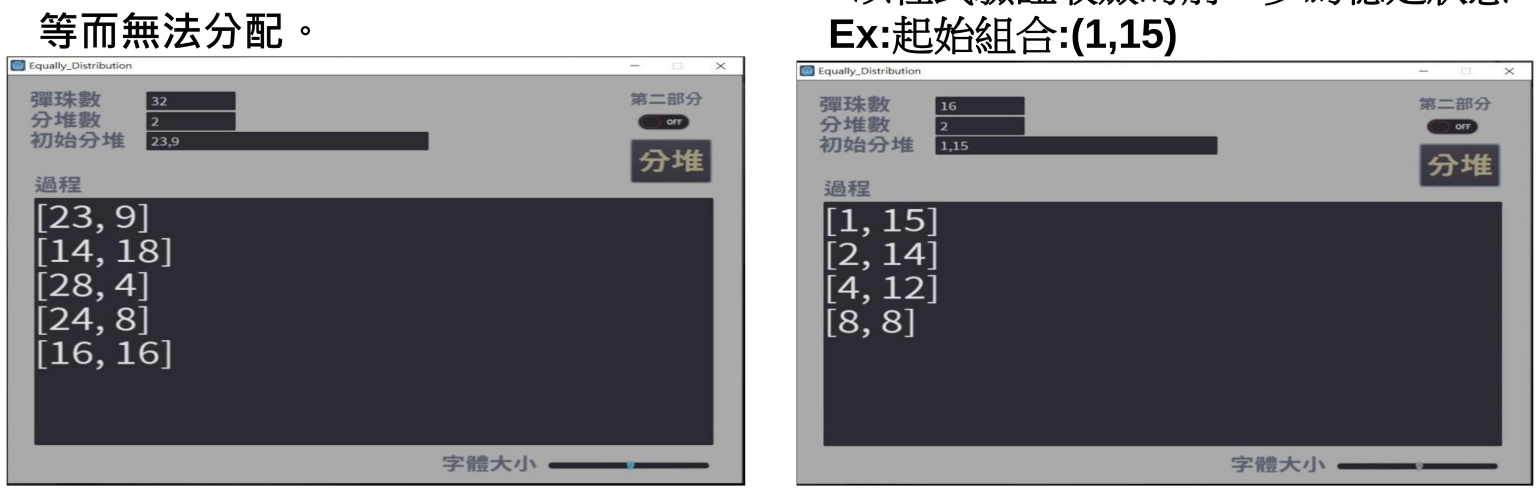
總和	迴圈個數
3	1
5	1
7	1
11	1
13	1
17	2
19	1
23	1
29	1
31	3
37	1
41	2

總和	迴圈個數
43	3
47	1
53	1
59	1
61	1
67	1
71	1
73	4
79	1
83	1
89	4
97	2

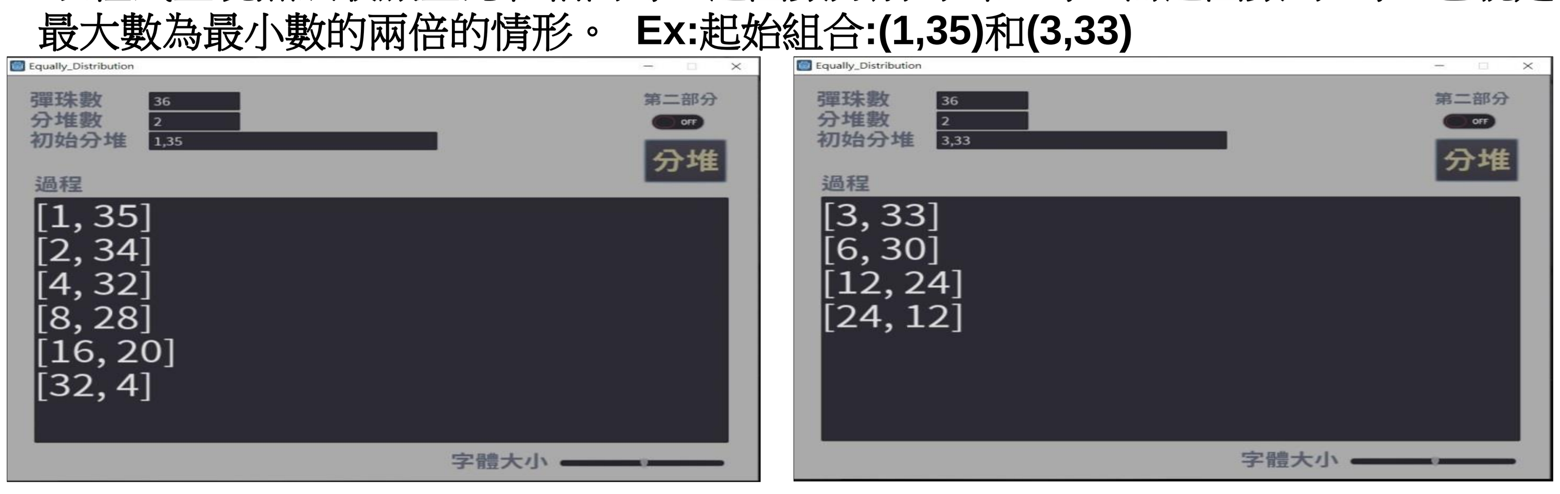
五、先利用Excel把數堆彈珠分配過程詳列後，再將分配過程所找出的規律運用數學邏輯並加入條件因素，寫成演算法，以程式GDScript動態語言寫出執行，呈現其分配過程。

1.兩堆彈珠，一堆是23個彈珠，另一堆是9個彈珠，如何分配讓這兩堆數目不一樣的彈珠進行平分？同時在每次進行分配時，都從數量較多的那一堆拿出當下較少數量那堆的個數，將其放到數量較少的那堆，反覆進行這樣的動作，直到因兩堆數量相等而無法分配。

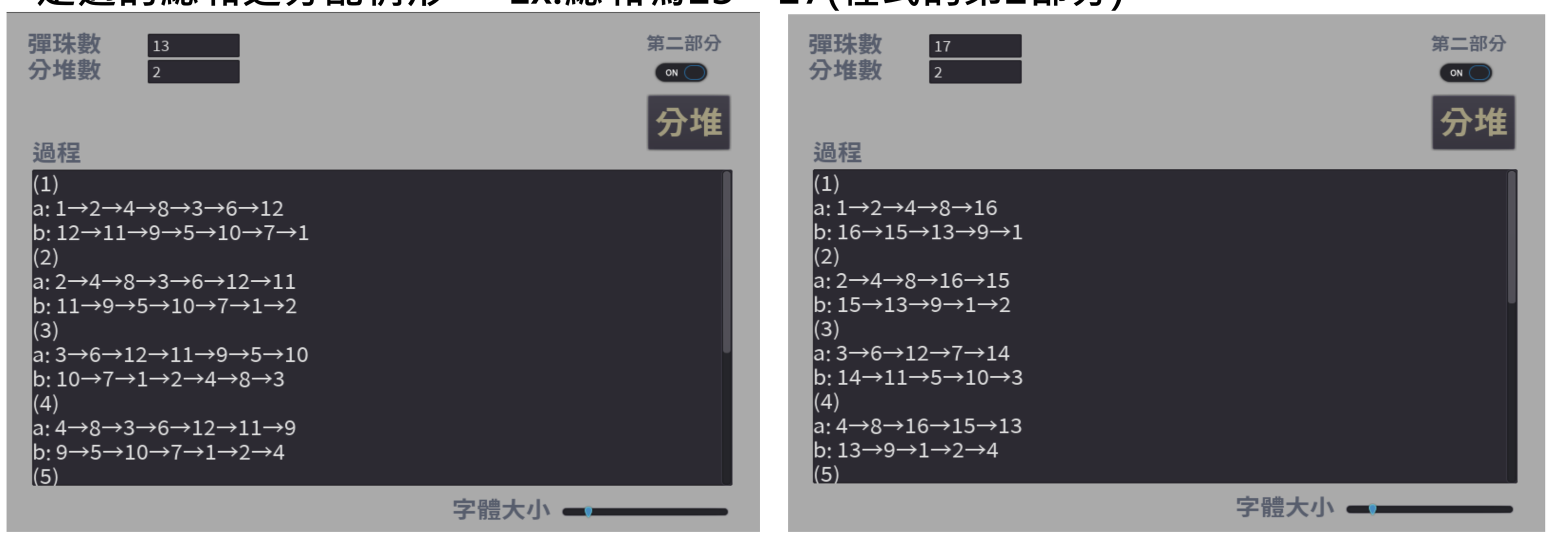
2.以程式驗證收斂的前一步為穩定狀態。Ex:起始組合:(1,15)



3.以程式呈現無法收斂且總和相同時，迴圈數分別為1和3時，而迴圈數為1時，也就是最大數為最小數的兩倍的情形。 Ex:起始組合:(1,35)和(3,33)



4.當總和為奇質數且分為2堆時，以程式呈現所有狀態都能走過且不重複，及無法都走過的總和之分配情形。 Ex:總和為13、17(程式的第2部分)



六、分別推導分堆問題及迴圈問題，我們找到方法寫出其演算法，能快速且無遺漏地列出迴圈，以程式GDScript動態語言寫出執行。

【主程式的演算法部分】：

#如果最大堆等於最小堆，停止分堆

1.假設最小數為a，最大數為b。

2.如果最大數等於最小數，則停止分配(當a=b)。

3.如果最大數不等於最小數，則繼續分配(當a≠b)。

#分堆進行　#最大堆減去最小堆　#最小堆加上自己的數量

1.假設最小數為a，最大數為b。

2.因分配規則為最大數減去最小數，所以b-a。

3.從最大數拿最小數的值給最小數時，最小數將加上自己本身。(a+a)

柒、討論

一、當∑分為數堆不同數目相加時，部分可均勻分配的數，會產生無法均勻分配時，我們歸納統整出的失敗情形有兩種：(一)最大數為最小數的兩倍；(二)出現某特定組合後，再經過若干步數之分配後，出現重複。然而，形成這兩種不同情形的條件為何？這還需要更多的時間去進一步觀察研討。

二、我們觀察到如果在迴圈中要所有組合都會經過且不重複，一定要分為兩堆且Σ是奇質數，然而Σ為某些奇質數時，卻不會所有組合都會經過且不重複，舉例：在100以內的數有17、31、41、43、73、89、97等共7個數。依循觀察其結果，我們日後可再進一步探究「為何只有奇質數才可能會所有組合都會經過且不重複？而其他數字卻一定不可能」

三、本作品還未找出: (一)不同總和與其最大迴圈之關聯性；(二)起始組合與收斂之關聯性。能否找出通式判別最大迴圈數與可否收斂？

四、原先我們設定的定義規則是「一次從最大數拿最小數的值放到最小數上面」，倘若並非這樣呢？比如說「三個數相加時，從最大數拿[總和平均數]到最小數。」這個方向值得之後繼續探討。

捌、結論

就參考文獻及我們所研究的作品，將研究範疇與貢獻統整如下表格：

全國科展	組 別	作 品 名 稱	研 究 範 疇 與 貢 獻
第55屆	國中	從平分問題到動態穩定	將兩堆小石子利用輾轉相除法記錄移動過程，使平分問題轉化成二進位制的移位問題！
第56屆	國中	再探均分問題的動態穩定	將三堆石頭移動成數量皆相等的狀態，利用二進位制歸納三堆時的移動次數型態，找出可形成穩定狀態的條件及移動方法與次數。
第58屆	國小	公平分配遊戲	將兩堆糖果公平分配成功的情形，觀察出A+B=4n至少有一組成功解及A+B=2^n 一律成功但都缺少了一般化的證明。
第60屆	國中	分堆問題之收斂性的探討 (本作品是我們所研究的)	1.當起始值可以收斂時： (1)探究分為數堆不同數目相加時，比較其相同或相異之處，並探究奇偶數變化與總和(∑)的關聯性。 (2)部分組合可收斂的數，在無法收斂時，先將其狀況歸納整理出來，再探究穩定狀態與收斂之關聯性。 2.當起始值無法收斂時: (1)在無法收斂並形成迴圈時，歸納在固定總和下，形成最大迴圈或最小迴圈之條件，與不同總和的迴圈數之值。 (2)歸納出當總和不同時，其形成的迴圈個數分別有多少個，並以表格呈現。 (3)觀察出只有分為兩堆且總和為奇質數時，才有可能所有組合皆走過且不重複，而非奇質數一定不可能形成。 3.將分配過程中所找出的規律，運用數學邏輯並加入條件因素，寫成演算法，並以程式GDScript動態語言執行，呈現其分配過程。(第56屆沒有程式輔助)

玖、參考資料及其他

一、參考資料：

(一)、團康輔導社/主編，有趣的數學遊戲，出版社:文國書局

(二)、徐筠悉、林佳妤、卓品瑜、陳楷翔，第五十八屆全國科展國小組數學科「公平分配遊戲」

(三)、林建銘、高暐芬、廖昇瑋，第五十六屆全國科展國中組數學科佳作「再探均分問題的動態穩定」

(四)、陳奕均，第五十五屆全國科展國中組數學科佳作「從平分問題到動態穩定」

(五)、馬榮喜、陳世易，國中數學第四冊，康軒文教事業股份有限公司，2015

(六)、許志農 主編，高中數學第二冊，龍騰文化事業股份有限公司，2011

(七)、徐力行 著，沒有數字的數學，天下文化，2003