中華民國第51屆中小學科學展覽會
作品說明書

國中組　數學科

最佳團隊合作獎

030410

n柱河內塔的捷徑建構與通式的尋找

學校名稱：新北市立福和國民中學

作者：	指導老師：
國二 黃祈昌	鄭釧鋒
國二 趙志嘉	洪駿源
國一 鄭百凱	

關鍵詞：河內塔、巴斯卡、遞迴式

n 柱河內塔的捷徑建構與通式的尋找

摘要

對於 n 柱河內塔的移動，當完成遊戲，其過程必存在「**半移動**」(名稱說明見 P_5)狀態。我們從「半移動」狀態中，尋找出如何達成「捷徑半移動」(名稱說明見 P_{14})的方法？此種方法為「滿格建構」(名稱說明見 P_{12})。進一步利用「捷徑半移動」，建構出「**河內塔的捷徑**」。並從「滿格建構」推導出的「滿格數量關係表」，發現其關係存在著「**巴斯卡三角圖形**」。利用「巴斯卡三角圖形」的關係，我們推導出 n 柱 m 環的通式。成功的解決了" Explorations in 4-peg Tower of Hanoi" (Ben Houston & Hassan Masum , 2004)這篇論文，所談及的『**百年來，河內塔 4 柱以上的移動是不能証明最優化**』。

壹、研究動機

在中華民國中小學第 50 屆科展評審中，教授用一個不屬於我們的研究策略，也能以 13 步完成 4 柱 5 環的河內塔移動。凸顯出，我們只是在既定的移動策略，做出優選，完成研究。並未考慮其它可能的移動策略，當然研究結果，不能令人信服。最終只能得到另外一位評審教授的安慰：「繼續加油！」

從失敗的過程中，我們左思右想，如何做出各種移動策略的比較？但越深入研究，越發現其複雜性，讓我們不得不放棄此方式。最後，只能重新思考，是否有其他方法？可以解決「**$n(n \geq 4)$ 柱河內塔，百年來未能證明最優化的問題。**」

貳、研究目的

1、是否能找到方法，判定移動的方式，是河內塔的捷徑？

2、若存在捷徑，能否建構出 n 柱河內塔的捷徑？

3、若存在捷徑，是否可推導出 n 柱河內塔遊戲完成移動的最少步數通式？

參、研究器材

紙、筆、筆電、Word、Excel、GSP 軟體、隨身碟。

肆、研究過程

一、 n 柱河內塔的移動規則：

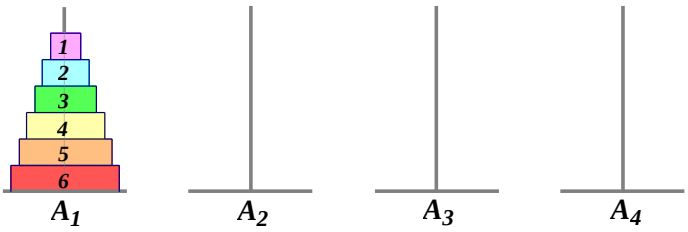
1、物件：包含

(1) n 根柱子，編號分別為 A_1 、 A_2 、……..、 A_n 。

(2) m 個大小不等的環，由小到大依序編號為 1、2、3、…、 m ，統稱為「**環號**」。

2、起始位置：將 m 個環由大到小、由下而上依序套在 A_1 柱子上，如圖(一)。

當 $n=4$ ， $m=6$



圖(一)

3、移動規則：須符合下面兩規定

(1) 一次只允許將套在柱子上最上層的 1 個環，從所在位置移動到別根柱子上，此移動的完成，我們稱為「1 次操作」記為「1 步」。

(2) 環號小的須置於環號大的上面。

4、最終位置：根據上述移動規則，當 A_1 柱子上的 m 個環全部被移動到 A_n 柱子上，則遊戲結束。

(實際上，最終位置可為不是 A_1 的任意柱，但為了方便說明，本研究定：

A_1 為「起點柱」、 A_n 為「終點柱」。)

二、 符號的意義：

1、 $H_n(m)$ ：表示在 n 柱河內塔的遊戲中，將 m 個環從 A_1 柱子移往 A_n 柱子上，當完成遊戲所需的最少步數，稱之。

例： $H_3(m)$ 表示三柱河內塔共 m 個環的移動，完成遊戲所需的最少步數。在文獻上，可查得 $H_3(m)=2^m-1$ 。

2、 $A_i \xrightarrow{k} A_j$ ：表示將環號 k 的環，根據遊戲規則，從 A_i 柱以一步移動到 A_j 。

何謂「捷徑」？

對於 n 柱 m 個環的河內塔，根據遊戲規則，能以最少步數完成 m 個環的移動，過程中的移動方式，稱之。

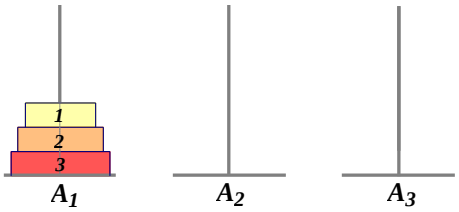
例如：3 柱 m 環，文獻上已記載當要以最少步數完成操作，其操作模式必為

$$H_3(m) = H_3(m-1) + 1 + H_3(m-$$

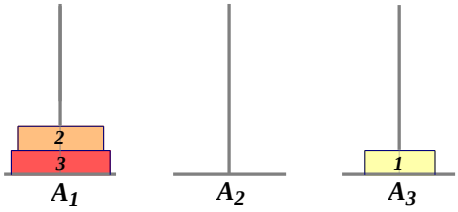
並根據此關係式，可推得公式為 $2^m - 1$ 。

觀察下面兩例：

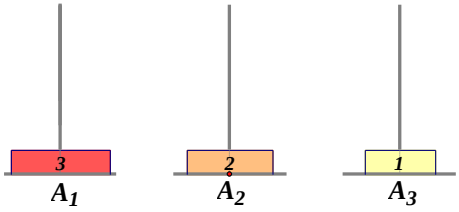
(1) 3 柱 3 環，完成移動<u>最少</u>為 $2^3 - 1 = 7$ 步。操作如下：



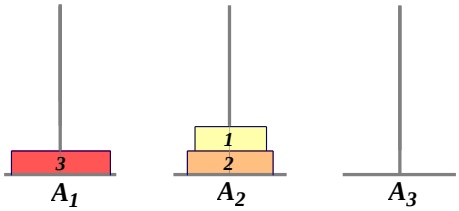
[起始位置]



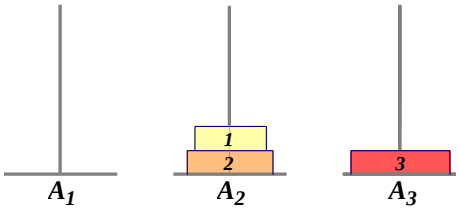
[步驟 1]



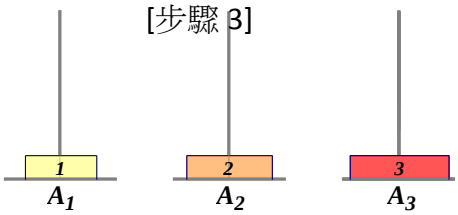
[步驟 2]



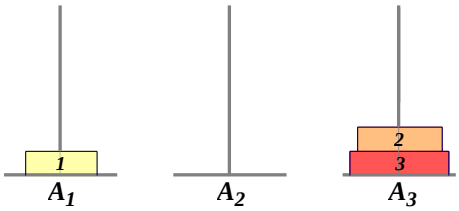
[步驟 3]



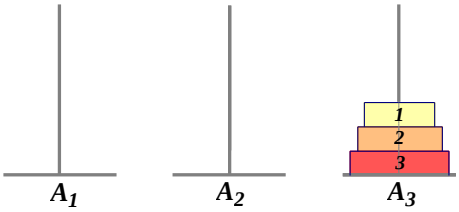
[步驟 4]



[步驟 5]



[步驟 6]



[步驟 7]---[目的位置]

將上述移動簡記如下

$A_1 \xrightarrow{1} A_3$　　$A_1 \xrightarrow{2} A_2$　　$A_3 \xrightarrow{1} A_2$　　$A_1 \xrightarrow{3} A_3$　　$A_2 \xrightarrow{1} A_1$

$A_2 \xrightarrow{2} A_3$　　$A_1 \xrightarrow{1} A_3$

上式的 7 步移動方式，即是「**捷徑**」。

將上述「捷徑」中的環號，依照操作先後順序，可得出數列$\langle M_3\rangle$：

$\langle M_3\rangle$：1，2，1，3，1，2，1

(2) 3 柱 4 環，完成移動<u>最少</u>為$2^4-1=15$步。操作簡記如下：

$A_1\xrightarrow{1}A_2$　$A_1\xrightarrow{2}A_3$　$A_2\xrightarrow{1}A_3$　$A_1\xrightarrow{3}A_2$　$A_3\xrightarrow{1}A_1$

$A_3\xrightarrow{2}A_2$　$A_1\xrightarrow{1}A_2$　$A_1\xrightarrow{4}A_3$　$A_2\xrightarrow{1}A_3$　$A_2\xrightarrow{2}A_1$

$A_3\xrightarrow{1}A_1$　$A_2\xrightarrow{3}A_3$　$A_1\xrightarrow{1}A_2$　$A_1\xrightarrow{2}A_3$　$A_2\xrightarrow{1}A_3$

上式的 15 步移動方式，即是「**捷徑**」。

將上述「捷徑」中的環號，依照操作先後順序，可得出數列$\langle M_4\rangle$：

$\langle M_4\rangle$：1，2，1，3，1，2，1，4，1，2，1，3，1，2，1

[發現]：(1) $\langle M_3\rangle$與$\langle M_4\rangle$兩數列分別以 3 及 4 為中心形成對稱。

(2) $\langle M_4\rangle$可由$\langle M_3\rangle$產生，即 $\langle M_4\rangle=\langle\langle M_3\rangle,4,\langle M_3\rangle\rangle$

為什麼會有此結果？此結果是否具備一般化？

分析：由於 3 柱m環的最少步數操作模式為

$$H_3(m)=H_3(m-1)+1+H_3(m-$$

根據此關係式，可得知上述發現結果是正確的。並利用此關係式，寫出

3 柱m環的「捷徑」數列之關係為　$\langle M_m\rangle=\langle\langle M_{m-1}\rangle,m,\langle M_{m-1}\rangle\rangle$

$\langle M_m\rangle=\langle\langle M_{m-1}\rangle,m,\langle M_{m-1}\rangle\rangle$的分析：

(1) $\langle M_m\rangle$為以m當中心所形成的對稱數列。

(2) 根據關係式，可知當完成「捷徑」操作，統計「捷徑」中環號出現的次數可得

環號	m	$m-1$	$m-2$	……	3	2	1
次數	1	2	2^2	……	2^{m-3}	2^{m-2}	2^{m-1}

由上表可推得　$H_3(m)=1+2+2^2+\cdots\cdots+2^{m-1}=2^m-1$

即文獻上 3 柱 m 環完成移動的最少步數公式。

此公式的得出方式有別於文獻上的三柱公式推導。我們想是否 n 柱 m 環皆可利用統計「捷徑」中環號出現的次數進而導出公式。

因此我們需先解決下面兩個問題：

[問題 1]：是否 $n(n \geq 4)$ 柱 m 環的操作皆可以以「對稱性操作」完成？

問題提出的想法，來自於「若具有對稱性，那麼移動步數僅需考慮一半」。

[問題 2]：如何確定何種移動方式是「捷徑」？

問題提出的想法，來自於「由於 3 柱 m 環，文獻上已記載，當要以最少步數完成操作，其移動策略唯一，即「捷徑」唯一。但 $n(n \geq 4)$ 柱 m 環的移動策略並不唯一，所以如何確定何種移動方式是「捷徑」？顯然很重要，實際上，也是我們研究是否能成功的關鍵。」

問題的解決：

[問題 1]：是否 $n(n \geq 4)$ 柱 m 環的操作皆可以以「對稱性操作」完成？

1. 我們先定義下列名詞：

(1) **半移動：對於河內塔 n 柱 m 環的遊戲中，當最大環號 m 第一次(註)移動至目的柱，我們稱此時的<u>移動狀態</u>為「半移動」。**（想法來自於「對稱性」）

註：根據河內塔的遊戲規則，若要以最少步數完成移動，則過程中，最大環號 m 必只移動 1 次(即 $A_1 \xrightarrow{m} A_n$)。

(2) 前操作：完成「半移動」之前的操作，稱之。

(3) 後操作：「半移動」之後至遊戲完成的操作，稱之。

何謂「對稱性操作」？

(4) 對稱性操作：對於河內塔 n 柱 m 環的遊戲中，以最大環號 m 為中心，當「前操作」與「後操作」之環號，依操作的先後順序所構成的數列，若形成對稱，則我們稱此操作為「對稱性操作」。

例如：4 柱 4 環

操作如下：

① $A_1 \xrightarrow{1} A_4$　② $A_1 \xrightarrow{2} A_3$　③ $A_1 \xrightarrow{3} A_2$　④ $A_4 \xrightarrow{1} A_3$　⑤ $A_1 \xrightarrow{4} A_4$

⑥ $A_3 \xrightarrow{1} A_1$　⑦ $A_2 \xrightarrow{3} A_4$　⑧ $A_3 \xrightarrow{2} A_4$　⑨ $A_1 \xrightarrow{1} A_4$

說明：(1) 當 4 柱 4 環的最大環(環號 4)到達終點柱(A_4)，即第 1 步操作至第 5 步操作，我們稱此移動完成為「半移動」。

(2) 第 1 步操作至第 4 步操作稱為「前操作」。

(3) 第 6 步操作至第 9 步操作稱為「後操作」。

按此操作結果，將環號依照操作先後順序記下，可得下面數列：

1，2，3，1，4，1，3，2，1

以 4 為中心形成「前操作」與「後操作」對稱，

我們將上述操作稱為**「對稱性操作」**。

2. 根據河內塔的遊戲規則，顯然規則具有移動的**[可逆性]**。

可推得　若 $A_i \xrightarrow{k} A_j$ 為可允許的移動，則 $A_j \xrightarrow{k} A_i$ 亦必為可允許的移動。

具體點，就如同影片的倒帶。

3. 是否 $n(n \geq 4)$ 柱 m 環的操作皆可以以對稱性完成操作？

(1) 以 4 柱 4 環為例

任意給出一個[半移動]

① $A_1 \xrightarrow{1} A_2$　② $A_1 \xrightarrow{2} A_3$　③ $A_2 \xrightarrow{1} A_3$　④ $A_1 \xrightarrow{3} A_2$　⑤ $A_1 \xrightarrow{4} A_4$

$A_1 \xleftarrow{1} A_2$　$A_1 \xleftarrow{2} A_3$　$A_2 \xleftarrow{1} A_3$　$A_1 \xleftarrow{3} A_2$　(「前操作」的倒帶)

⑨ $A_4 \xleftarrow{1} A_2$　⑧ $A_4 \xleftarrow{2} A_3$　⑦ $A_2 \xleftarrow{1} A_3$　⑥ $A_4 \xleftarrow{3} A_2$　(實際的「後操作」)

可得數列： 1，2，1，3，4，3，1，2，1　為「對稱性操作」。

(2) 將上述移動推廣成 n 柱 m 環的移動

對於任意給定的「半移動」，

① 針對「前操作」的每一個移動　$A_i \xrightarrow{k} A_j$

② 皆存在「倒帶」的對應移動 $A_j \xrightarrow{k} A_i$

③ 利用②的「倒帶」移動，產生的實際「後操作」，如下：

(i) 當 $i=1$，則實際操作 $A_j \xrightarrow{k} A_n$

(ii) 當 $i=n$，則實際操作 $A_j \xrightarrow{k} A_1$

(iii) 當 $i \neq 1$ 且 $i \neq n$，則實際操作 $A_j \xrightarrow{k} A_i$

由上述可證明：

[結論一]：$n(n \geq 4)$ 柱 m 環的操作皆可以以「對稱性」完成操作。 #

確定了 n 柱 m 環的操作皆可以以「對稱性」來完成操作。接著，只要我們研究出「前操作」的捷徑，那麼 n 柱 m 環就可以以最少步數完成移動。以下是我們需解決的[問題]：

[問題 2]：如何確定何種移動方式是「捷徑」？

對於盤子數越多時，其移動的方式大增。因此，從移動策略去研究，很難確定出何種方式的移動為「捷徑」？我們想這應是「**$n(n \geq 4)$ 柱河內塔，百年來未能證明最優化**」的關鍵吧！以下，我們試著看是否能在不考慮移動策略的情況下，確定何種方式為「捷徑」？

由[問題 1]的結果，可知河內塔遊戲皆可以以「對稱性操作」來完成。因此，我們只需考慮「半移動」。

為了方便說明，對於 n 柱 m 環，我們規定：

(1) A_1 為**起點柱**。

(2) A_i $(i \neq 1, i \neq n)$ 為**暫置柱**。

(3) A_n 為**終點柱**。

根據「半移動」的意義，表示當完成[半移動]時，**起點柱(A_1)**必定全空，**終點柱(A_n)**只有最大環，其他的環則分布在**暫置柱**。

如何讓「半移動」形成一個**最佳移動**的狀態？

以下本研究的討論，若不特別聲明，皆表示在**最佳移動**的原則下。

以 4 柱為例，當 $m=4$ 、5 、6 ，其「半移動」之狀態(其中 A_2 與 A_3 柱子的狀態可互換)，只能為下表，為何最佳？後面有理由說明

(表一)

柱子編號	4 柱 4 環	4 柱 5 環		4 柱 6 環	4 柱 6 環(註)
A_1(起點柱)					
A_2(暫置柱)	b	b c	b c c	b c c	3 2 1
A_3(暫置柱)	b c	b c	b	b c	5 4
A_4(終點柱)	a	a	a	a	6
最少步數	$H_4(4)=9$	$H_4(5)=13$		$H_4(6)=17$	

理由說明：

上表中，對所完成的「半移動」之狀態，各環所需的最少步數及其所代表的意義：

1. a：表示從起點柱只經 1 次移動就可到達終點柱，只能是最大的環，共 1 個。

   即 a 表示 1 步到位；實際上，對於 n 柱 m 環，a 永遠 1 個。

2. b：表示是位於**暫置柱**底部的環，最少經 1 次移動。再根據「對稱性」，可得 b 至終點柱，最少 2 步到位。

   即 b 表示 2 步到位；4 柱最多存在 2 個 b 。(因為暫置柱有 2 柱)

3. c：最少經 2 次移動。再根據「對稱性」，可得 c 至終點柱，最少 4 步到位。

   至於

   4 柱最多存在幾個 c ？

分析：(1) 當達「半移動」時，有 1 根空柱 A_1，所以第 1 個ⓑ (ⓑ群中最大環)上方至多可放 1 個ⓒ (為了達到最佳移動，此ⓒ必為ⓒ群中最大環。理由在分析(2)中)。

(2) 當第 1 個ⓑ到位時，會多出 1 根空柱(假設為 A_3)，此時ⓒ已在 A_1。假如 A_1 上的ⓒ所代表的環號是接續在ⓑ之後，則接著當ⓒ移往 A_4，此時會產生 2 根空柱(A_1 、 A_3)。又因為ⓒ群表示在「前操作」中只能移動 2 步，所以 A_2 上最多可允許擺放 2 個ⓒ。

綜合(1)、(2)可得 4 柱最多存在 3 個ⓒ。

即ⓒ表示 4 步到位；4 柱最多存在 3 個ⓒ。

由上述 1、2、3 說明與分析，可推得(表一)為最佳移動之「半移動」狀態圖。且此時，4 柱 6 環最佳移動之「半移動」狀態圖，其環號配置，只能為表(一)附註。

利用(表一)的狀態圖，再利用「對稱性」，即可得最佳移動。顯然按此移動方式，即為一條「**捷徑**」。

去年國展評審教授所提 4 柱 5 環的問題解決：

首先，利用其「半移動」狀態圖，填入可行環號，可得 8 種狀態圖，且每 1 種狀態圖的移動「**捷徑**」皆為 13 步

理由：因為 1 個ⓐ、2 個ⓑ、2 個ⓒ

所以共 $1+2\times2+4\times2=13$ 步 #

如下表：(表二)

柱子編號	4 柱 5 環			
A_1(起點柱)				
A_2(暫置柱)	2 1	3 1	3 2	3 2 1
A_3(暫置柱)	4 3	4 2	4 1	4
A_4(終點柱)	5	5	5	5
柱子編號	4 柱 5 環			
A_1(起點柱)				
A_2(暫置柱)	4 3	4 2	4 1	4
A_3(暫置柱)	2 1	3 1	3 2	3 2 1
A_4(終點柱)	5	5	5	5

利用上表，我們可得評審教授所提的例子，為上表第六個「半移動」狀態(表二之黃色區)，其 13 步的實際移動「**捷徑**」路線可根據下面步驟得出：

[步驟 1]：由其狀態可推得「後操作」的實際移動路線為

⑧ $A_2 \xrightarrow{2} A_1$　⑨ $A_2 \xrightarrow{4} A_4$　⑩ $A_3 \xrightarrow{1} A_2$

⑪ $A_3 \xrightarrow{3} A_4$　⑫ $A_1 \xrightarrow{2} A_4$　⑬ $A_2 \xrightarrow{1} A_4$

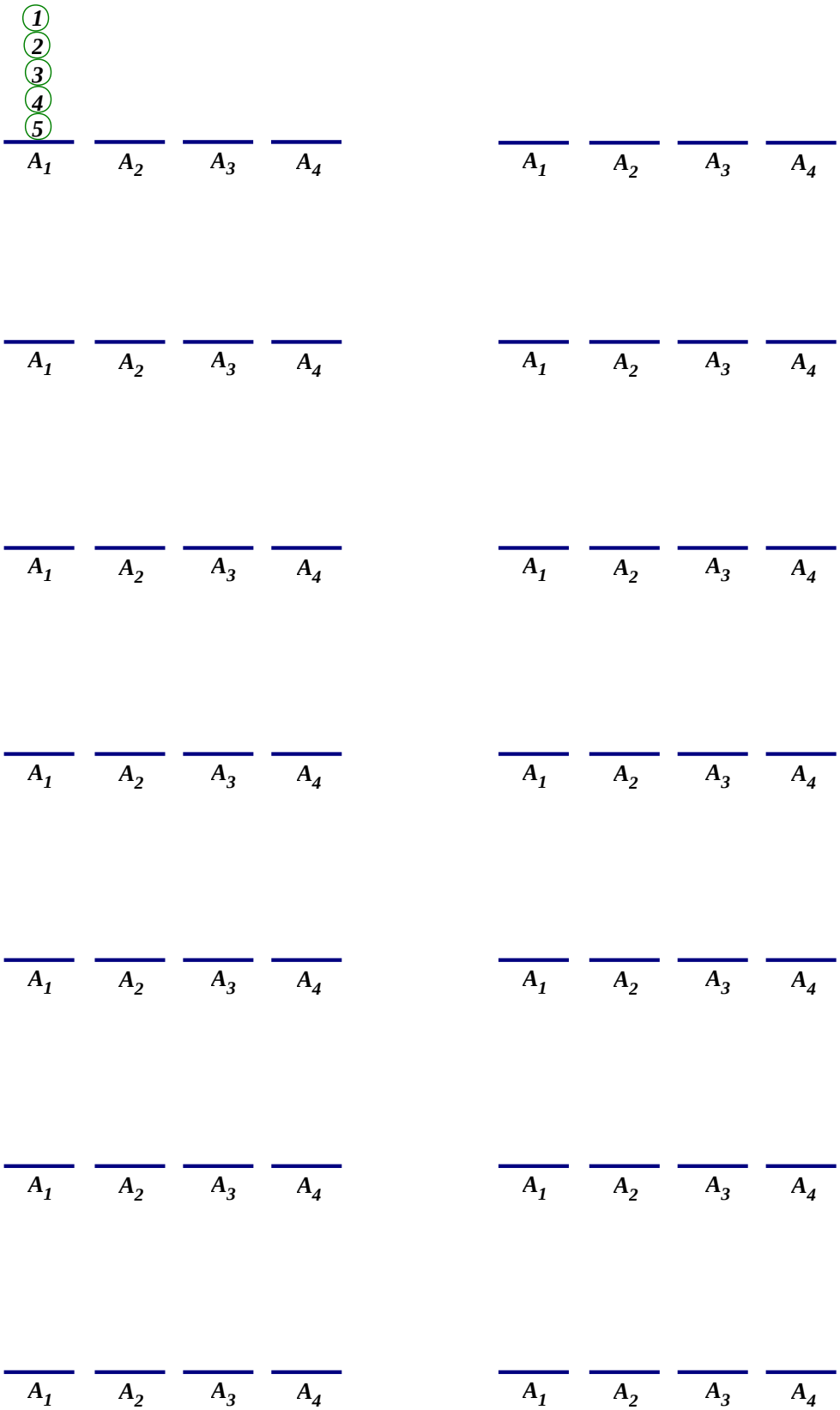
[步驟 2]：利用「前操作」與「後操作」的對稱性，可得「前操作」實際移動路線為

① $A_1 \xrightarrow{1} A_2$　② $A_1 \xrightarrow{2} A_4$　③ $A_1 \xrightarrow{3} A_3$

④ $A_2 \xrightarrow{1} A_3$　⑤ $A_1 \xrightarrow{4} A_2$　⑥ $A_4 \xrightarrow{2} A_2$

[步驟 3]：由「前操作」與「後操作」及對稱中心 $\boxed{⑦\ A_1 \xrightarrow{5} A_4}$，可得 4 柱 5 環的 13 步移動「**捷徑**」。

[4 柱 5 環試玩區]



預知結論：最少完成步數必為 **13** 步。且若以 **13** 步完成移動，則第 **7** 步必形成「半移動」狀態，且此狀態必存在於(表二)中。

為了方便說明，我們訂一些規定與名稱：

1. 我們稱ⓐ群、ⓑ群、ⓒ群、ⓓ群、……，分別為第 1 群、第 2 群、第 3 群、第 4 群、……。
2. 將上述「半移動」中ⓐ、ⓑ、ⓒ群皆填滿時，稱為「3 群滿格」；當ⓐ、ⓑ、ⓒ、ⓓ群皆填滿時，稱為「4 群滿格」；如此下去，可得「5 群滿格」、「6 群滿格」、……。並在不混淆時，皆簡稱為**「滿格」**。
3. 「半移動」狀態中，「暫置柱」底環(即ⓑ群)環號較大者置於編號較大之柱子上。

4 柱「3 群滿格」的推廣

在 4 柱 6 環的「3 群滿格」分析中，我們發現要達**「滿格」**，實際上只需依序透過兩個步驟，我們特稱為**「滿格建構」**：

[滿格步驟 1]：柱子 A_3 上的環，只能利用 3 柱(A_1 、 A_3 、 A_4)的移動來完成「後操作」。

[滿格步驟 2]：柱子 A_2 上的環，是以 4 柱(A_1 、 A_2 、 A_3 、 A_4)的移動來完成「後操作」。

1. 由[滿格步驟 1]，因為是 3 柱移動，所以可得結論：

   [結論二之 1]：柱子 A_3 上每一群的數量只能 1 個。

2. 由[滿格步驟 2]，可得下面結論：

   當達「α 群滿格」時，根據對稱性操作可得

   [結論二之 2]：柱子 A_2 上，各環的「後操作」，可視為 4 柱前 $(\alpha-1)$ 群各環的完成定位操作。

   所以柱子 A_2 上

   第 α 群的數量 = 第 $(\alpha-1)$ 群的總數量。

   [結論二之 3]：到達定位時，

   第 α 群的步數 = 第 $(\alpha-1)$ 群的步數 $\times 2$

由上述結論，可推得下表：

[表三之 1] ：「4 群滿格」、「5 群滿格」

柱子編號	4 柱 6 環	4 柱 10 環	4 柱 15 環
A_1(起點柱)			
A_2(暫置柱)	b c c	b c d c d d	b c d e c d e d e e
A_3(暫置柱)	b c	b c d	b c d e
A_4(終點柱)	a	a	a
最少步數	$H_4(6)=17$	$H_4(10)=49$	$H_4(15)=129$

說明：

1. 表中 4 柱 10 環柱子 A_2 上群的排列，根據[**結論二之 2**]，依序(A_4、A_3、A_2)將表中 4 柱 6 環作排列，接著以後續的英文字母替換得到。

   即 a b c b c c ⇒ b c d c d d

2. 表中「滿格」的步數計算，是根據[**結論二之 3**]先得到 a、b、c、d、e 群每一個分別代表 1、2、4、8、16 步。因此可得

   $H_4(10)=1\times1+2\times2+4\times3+8\times4=49$

   $H_4(15)=1\times1+2\times2+4\times3+8\times4+16\times5=129$

上述「滿格」配置的完成，因為各群要得到最多數量，所以環號配置必為[表三之 2]

[表三之 2]：最佳「半移動」之實際環號配置

柱子編號	4 柱 6 環	4 柱 10 環	4 柱 15 環
A_1(起點柱)			
A_2(暫置柱)	3 2 1	6 5 4 3 2 1	10 9 8 7 6 5 4 3 2 1
A_3(暫置柱)	5 4	9 8 7	14 13 12 11
A_4(終點柱)	6	10	15

4 柱 m 環的一般化

1. 由[結論二之 1]與[結論二之 2]可得到

   [結論二之 4]：4 柱移動，當達「α 群滿格」時，則第 α 群的數量 $=\alpha$ 個。

2. 由[結論二之 3]可得到

   [結論二之 5]：4 柱移動，第 α 群的步數 $=2^{\alpha-1}$

3. 利用[結論二之 4]與[結論二之 5]可得到 4 柱 m 環的通式為

   $H_4(m)=1\times1+2\times2+2^2\times3+\cdots\cdots+2^{\alpha-1}\times\alpha+2^{\alpha}\times\beta$

   其中：$m=1+2+3+\cdots\cdots+\alpha+\beta$，$0\le\beta\le\alpha$

從上述研究，我們可回答：

[問題 2]：如何確定何種移動方式是「捷徑」？

對於 n 柱 m 環的「半移動」狀態中，當滿足下面兩個配置：

[配置一]：達成「α 群滿格」，α 為最大值。

[配置二]：當完成[配置一]，尚有剩餘 β 個環，則此 β 個環須配置在第 $\alpha+1$ 群。

接著，以「倒帶」及「對稱性」完成實際的移動路線，即為一條「捷徑」。

我們稱滿足上述的兩個配置為「捷徑半移動」。

[問題 3]：如何建構 n 柱 m 環的「捷徑半移動」？

利用 4 柱 m 環的「滿格建構」，我們可推得 5 柱 m 環的「滿格建構」如下：

[滿格步驟 1]：柱子 A_4 上的環，只能利用 3 柱(A_1、A_4、A_5)的移動來完成「後操作」。

[滿格步驟 2]：柱子 A_3 上的環，是以 4 柱(A_1、A_3、A_4、A_5)的移動來完成「後操作」。

[滿格步驟 3]：柱子 A_2 上的環，是以 5 柱(A_1、A_2、A_3、A_4、A_5)的移動來完成「後操作」。

1. 由[滿格步驟 1]，因為是 3 柱移動，所以可得結論：

[結論三之 1]：柱子 A_4 上每一群的數量只能 1 個。----------(a)

2. 由[滿格步驟 2]，可得下面結論：

當達「α 群滿格」時，根據對稱性操作可得

[結論三之 2]：柱子 A_3 上，各環的「後操作」，可視為 4 柱 $\boxed{\text{前}(\alpha-1)\text{群}}$ 各環的完成定位操作。

所以柱子 A_3 上，第 α 群的數量

$=$(「$(\alpha-1)$ 群滿格」中，4 柱第 $(\alpha-1)$ 群的總數量）----------(b)

[結論三之 3]：到達定位時，

第 α 群的步數 $=$ 第 $(\alpha-1)$ 群的步數 $\times 2$

3. 由[滿格步驟 3]，可得下面結論：

當達「α 群滿格」時，根據對稱性操作可得

[結論三之 4]：柱子 A_2 上，各環的「後操作」，可視為 5 柱 $\boxed{\text{前}(\alpha-1)\text{群}}$ 各環的完成定位操作。

所以柱子 A_2 上，第 α 群的數量

$=$(「$(\alpha-1)$ 群滿格」中，5 柱第 $(\alpha-1)$ 群的總數量）----------(c)

[結論三之 5]：到達定位時，

第 α 群的步數 $=$ 第 $(\alpha-1)$ 群的步數 $\times 2$

由 (a)、(b)、(c)，可推得

(1) 5 柱第 α 群的總數量 $=$ 3 柱第 $(\alpha-1)$ 群的總數量

$+$ 4 柱第 $(\alpha-1)$ 群的總數量

$+$ 5 柱第 $(\alpha-1)$ 群的總數量 ----------(d)

(2) 5 柱移動，第 α 群的步數 $= 2^{\alpha-1}$

利用上述結果，可得[表四之 1]：

[表四之 1] ：「3 群滿格」、「4 群滿格」

柱子編號	5 柱 10 環	5 柱 20 環
A_1(起點柱)		
A_2(暫置柱)	b c c c	b c d c d d c d d d
A_3(暫置柱)	b c c	b c d c d d
A_4(暫置柱)	b c	b c d
A_5(終點柱)	a	a
最少步數	$H_5(10)=31$	$H_5(20)=111$

說明：5 柱中，當「4 群(即 a 、b 、c 、d 群)滿格」，由(c)式可知，

柱子 A_2 上

d 群的數量＝(「3 群滿格」中，c 群的總數量)$=1+2+3=6$

[表四之 2]第α群數量關係表：

柱數(n)	a	b	c	d	e	f	g
3	1	1	1	1	1	1	1
4	1	2	3	4	5	6	7
5	1	3	6	10	15	21	28

說明：5 柱中，當「4 群滿格」，由(d)式可知

d 群的數量＝(3 柱 c 群的總數量

＋4 柱 c 群的總數量

＋5 柱 c 群的總數量)

$=1+3+6=10$

同樣的方式，可推得n柱m環的「滿格建構」一般化步驟，如下：

[滿格步驟 1]：柱子 A_{n-1} 上的環，只能利用 3 柱(A_1 、 A_{n-1} 、 A_n)的移動來完成「後操作」。

[滿格步驟 2]：柱子 A_{n-2} 上的環，是以 4 柱(A_1 、 A_{n-2} 、 A_{n-1} 、 A_n)的移動來完成「後操作」。

⋮ ⋮

[滿格步驟 $n-2$]：柱子 A_2 上的環，是以 n 柱的移動來完成「後操作」。

對於[滿格步驟 i]，可得下面結論：

當達「α 群滿格」時，根據對稱性操作可得

[結論三之 6]：柱子 A_{n-i} 上，各環的「後操作」可視為 $(i+2)$ 柱，前 $(\alpha-1)$ 群各環的完成定位操作。

所以柱子 A_{n-i} 上，第 α 群的數量＝（「$(\alpha-1)$ 群滿格」中，$(i+2)$ 柱第 $(\alpha-1)$ 群的總數量）

[結論三之 7]：到達定位時，

第 α 群的步數＝第 $(\alpha-1)$ 群的步數 $\times 2$

由上述結論，可推得：

n 柱 m 環，當達「α 群滿格」時，根據對稱性操作可得

(1) n 柱第 α 群的總數量

$$=\sum_{i=1}^{n-2}[\text{柱子 } A_{n-i} \text{ 上，「}(\alpha-1)\text{ 群滿格」中，第 } \alpha \text{ 群的總數量}]$$

$$=\sum_{i=1}^{n-2}[(i+2)\text{ 柱第 }(\alpha-1)\text{ 群的總數量}]$$ ----------[群的公式]

(2) n 柱移動，第 α 群的步數 $=2^{\alpha-1}$

利用上述結果，可得「滿格數量關係表」：[表五]

柱數 (n)	a	b	c	d	e	f	g
3	1	1	1	1	1	1	1
4	1	2	3	4	5	6	7
5	1	3	6	10	15	21	28
6	1	4	10	20	35	56	84
7	1	5	15	35	70	126	210
8	1	6	21	56	126	252	462

說明：1. 8 柱ⓕ群的數量＝(3 柱ⓔ群的總數量＋4 柱ⓔ群的總數量

＋5 柱ⓔ群的總數量＋6 柱ⓔ群的總數量

＋7 柱ⓔ群的總數量＋8 柱ⓔ群的總數量)

＝1＋5＋15＋35＋70＋126

2. 上表中，以虛線的方向觀察，形成「巴斯卡」排列。

上表的應用：

1. $H_5(22)=1+2\times3+2^2\times6+2^3\times10+2^4\times2=143$

2. $H_6(22)=1+2\times4+2^2\times10+2^3\times7=105$

[問題 4]：$H_n(m)$ 的通式推導：

在上述研究中，我們可得到下面資料：

1. 第 1 群(a)、第 2 群(b)、第 3 群(c)、---、第 α 群，分別代表 1、2、2^2、⋯、$2^{\alpha-1}$ 步到位。
2. 當達「滿格」時，可發現：各柱中群的數量所成的數列，類似「巴斯卡」排列。
3. 在利用「滿格數量關係表」計算最少步數時，可知道需將 m 個環，做「滿格」的分配。因此，有必要計算出各柱當形成「滿格」時，環的總數量。所以，我們將「滿格數量關係表」中，n 柱中群的數量，所形成的數列，假設為 $<f_n(k)>$ 數列。如此，[表五]可改寫成[表六]：

[表六]

到位步數	1	2	2^2	2^3	2^4	2^5	2^6
$f_3(k)$	1	1	1	1	1	1	1
$f_4(k)$	1	2	3	4	5	6	7
$f_5(k)$	1	3	6	10	15	21	28
$f_6(k)$	1	4	10	20	35	56	84
$f_7(k)$	1	5	15	35	70	126	210
$f_8(k)$	1	6	21	56	126	252	462

從[表六]中，可得下列關係式：

(1) $f_n(k)=f_3(k-1)+f_4(k-1)+\cdots+f_n(k-1)$ …… (Ⅰ)式

(2) $f_n(k)=f_{n-1}(1)+f_{n-1}(2)+\cdots+f_{n-1}(k)$ …… (Ⅱ)式

(3) $f_n(k)=f_{n-1}(k)+f_n(k-1)$ …… (Ⅲ)式

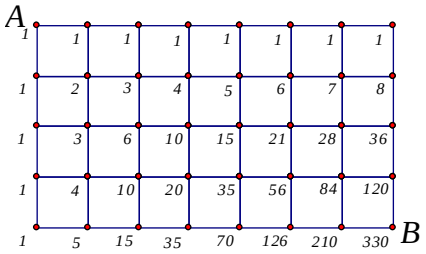
(Ⅰ)式即是[**群的公式**]，又(Ⅱ)式可取代(Ⅰ)式，因此可將[**群的公式**]轉化成(Ⅱ)式。

$f_n(k)$ 的通式：

我們先將[表五]的數列資料連結到棋盤捷徑方法數：

即對於 $m\times n$ 棋盤，由左上角 A 點移動到 B 點的捷徑之方法數：

以 4×7 為例：



可得 A 點移動到 B 點的捷徑之方法數為

$$\frac{11!}{4!7!}=\frac{11\times10\times9\times8}{4\times3\times2\times1}=330$$

同樣的，

我們可利用排列組合得 $<f_n(k)>$ 之一般項通式為

$$f_n(k)=\frac{(k+n-4)!}{(k-1)!(n-3)!}\qquad (n\geq 3,k\geq 1)$$

可得 ① $f_4(k)=k$

② $f_5(k)=\dfrac{k(k+1)}{2}$

③ $f_6(k)=\dfrac{k(k+1)(k+2)}{6}$

④ $f_7(k)=\dfrac{k(k+1)(k+2)(k+3)}{24}$

接下去，我們試著利用 $f_n(k)$ 的通式，推導 $H_n(m)$ 的通式。

$H_n(m)$ 的通式推導（$n\geq 4$）：

$$H_n(m)=f_n(1)\cdot 2^0+f_n(2)\cdot 2^1+f_n(3)\cdot 2^2+\cdots+f_n(k)\cdot 2^{k-1}+s\cdot 2^k$$

其中： $0\leq s<f_n(k+1)\Rightarrow 0\leq s<\dfrac{(k+n-3)}{k!(n-3)}$

令 $P_n(k)=f_n(1)\cdot 2^0+f_n(2)\cdot 2^1+f_n(3)\cdot 2^2+\cdots\cdots+f_n(k)\cdot 2^{k-1}$ …… ①

①×2 $\Rightarrow 2P_n(k)=f_n(1)\cdot 2^1+f_n(2)\cdot 2^2+f_n(3)\cdot 2^3+\cdots\cdots+f_n(k)\cdot 2^k$ …… ②

②－①可得

$$P_n(k)=f_n(k)\cdot 2^k-\{f_n(1)\cdot 2^0+[f_n(2)-f_n(1)]\cdot 2^1+[f_n(3)-f_n(2)]\cdot 2^2$$

$$+\cdots+[f_n(k)-f_n(k-1)]\cdot 2^{k-1}\}$$

$$=f_n(k)\cdot 2^k-[f_{n-1}(1)\cdot 2^0+f_{n-1}(2)\cdot 2^1+f_{n-1}(3)\cdot 2^2+\ldots+f_{n-1}(k)\cdot 2^{k-1}]$$

$$=f_n(k)\cdot 2^k-P_{n-1}(k)$$ ---------- [滿格通式]

又 $m=f_n(1)+f_n(2)+f_n(3)+f_n(4)+\cdots+f_n(k)$ ， $0\le s<f_n(k+1)$

$$=f_{n+1}(k)+s$$ [因為 $f_n(k)=f_{n-1}(1)+f_{n-1}(2)+\cdots+f_{n-1}(k)$]

所以 $f_{n+1}(k)\le m<f_{n+1}(k)+f_n(k+1)$

$f_{n+1}(k)\le m<f_{n+1}(k+1)$ [因為 $f_n(k)=f_{n-1}(k)+f_n(k-1)$]

因此可得： $H_n(m)=f_n(k)\cdot 2^k-P_{n-1}(k)+s\times 2^k$

$$=f_n(k)\cdot 2^k-P_{n-1}(k)+\left[m-f_{n+1}(k)\right]\times 2^k$$ [因為 $s=m-f_{n+1}(k)$]

$$=\left[m-f_{n+1}(k)+f_n(k)\right]\times 2^k-P_{n-1}(k)$$ ------- [n 柱公式]

其中 $f_{n+1}(k)\le m<f_{n+1}(k+1)$ ， $f_n(k)=\dfrac{(k+n-4)!}{(k-1)!(n-3)!}$ $(n\ge 3,k\ge 1)$ #

由 $H_n(m)$ 的通式，可得到

1. $H_4(m)=[m-f_5(k)+f_4(k)]\times 2^k-P_3(k)$

又 $P_3(k)=f_3(1)\cdot 2^0+f_3(2)\cdot 2^1+f_3(3)\cdot 2^2+\cdots\cdots+f_3(k)\cdot 2^{k-1}$

$$=2^0+2^1+2^2+\cdots+2^{k-1}$$

$$=2^k-1$$ [實際上， $P_3(k)=H_3(k)$]

$f_4(k)=k$ ， $f_5(k)=\dfrac{k(k+1)}{2}$

所以 $H_4(m)=[m-\dfrac{k(k+1)}{2}+k]\times 2^k-(2^k-1)$

$$=\left[m-\frac{1}{2}(k^2-k+2)\right]\times 2^k+1$$ ------- [4 柱公式]

其中 $f_5(k)\le m<f_5(k+1)$

2. $H_5(m)=[m-f_6(k)+f_5(k)]\times 2^k-P_4(k)$

又 $P_4(k)=f_4(k)\cdot 2^k-P_3(k)$

且 $P_3(k)=2^k-$ ， $f_4(k)=k$

得 $P_4(k)=f_4(k)\cdot 2^k-P_3(k)=k\cdot 2^k-(2^k-1)=(k-1)2^k+1$

又 $f_5(k)=\dfrac{k(k+1)}{2}$ ， $f_6(k)=\dfrac{k(k+1)(k+2)}{6}$

所以 $H_5(m)=[m-f_6(k)+f_5(k)]\times 2^k-P_4(k)$

$$=\left[m-\frac{k(k+1)(k+2)}{6}+\frac{k(k+1)}{2}\right]\times 2^k-\left[(k-1)\times 2^k+1\right]$$

$$=\left[m-\frac{1}{6}(k^3+5k-6)\right]\times 2^k-1$$ ------- [5 柱公式]

其中 $f_6(k)\le m<f_6(k+1)$

3. $H_6(m)=[m-f_7(k)+f_6(k)]\times 2^k-P_5(k)$

又 $P_5(k)=f_5(k)\cdot 2^k-P_4(k)$

$$=\frac{k(k+1)}{2}\cdot 2^k-[(k-1)2^k+1]$$

$$=(k^2-k+2)\times 2^{k-1}-1$$

所以 $H_6(m)=[m-f_7(k)+f_6(k)]\times 2^k-P_5(k)$

$$=\left[m-\frac{k(k+1)(k+2)(k+3)}{24}+\frac{k(k+1)(k+2)}{6}\right]\times 2^k-\left[(k^2-k+2)\times 2^{k-1}-1\right]$$

$$=\left[2m-\frac{1}{12}(k^4+2k^3+11k^2-14k+24)\right]\times 2^{k-1}+1$$ ------- [6 柱公式]

其中 $f_7(k)\le m<f_7(k+1)$

$H_n(m)$ 公式的使用

在 $H_n(m)$ 的公式中，我們需要知道 k 值與 $P_{n-1}(k)$ ，才能計算出 $H_n(m)$ 。但當 m 值太大時，則 k 值不易求得。因此，我們想到可利用 *G.S.P.* 軟體來尋找 k 值及計算 $H_n(m)$ 。以 $H_7(m)$ 為例：

k 值的尋找

從 $H_n(m)$ 的公式推導中，可知 k 值的尋找，需透過不等式 $f_{n+1}(k)\le m<f_{n+1}(k+1)$

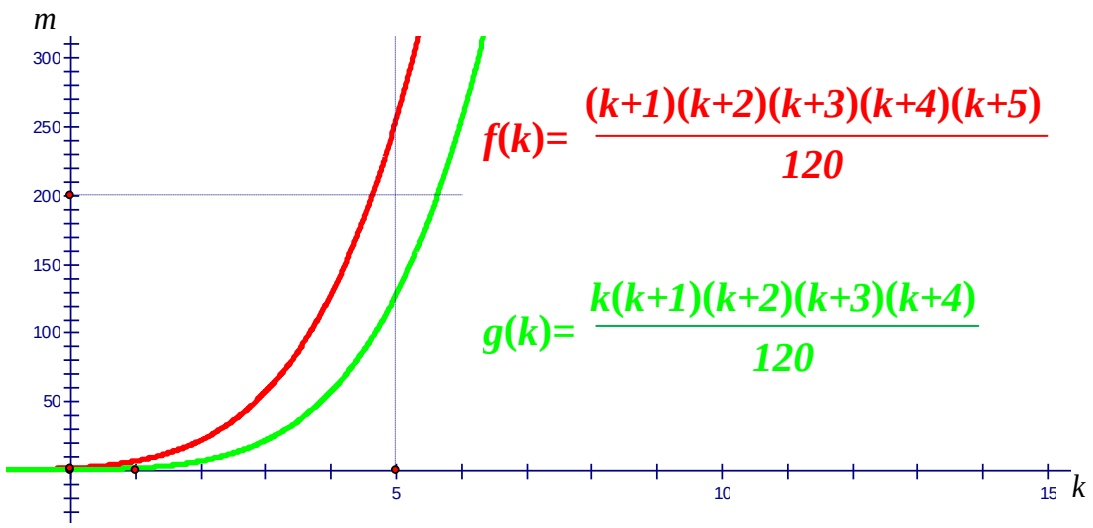
因為 $n=7$ ，所以 k 值滿足 $f_8(k)\le m<f_8(k+1)$

又因為 $f_n(k)=\dfrac{(k+n-4)!}{(k-1)!(n-3)!}$

所以 $f_8(k)=\dfrac{k(k+1)(k+2)(k+3)(k+4)}{5!}$ （假設為 $g(k)$）

$f_8(k+1)=\dfrac{(k+1)(k+2)(k+3)(k+4)(k+5)}{5!}$ （假設為 $f(k)$）

利用 $G.S.P.$ 繪出兩函數圖形如下：



註：橫軸為 k ，縱軸為 m

我們以 $m=200$ 為例，從上圖中，可知 $k=5$

又 $P_6(k)=f_6(k)\cdot 2^k-P_5(k)$

$$=\left[\frac{k(k+1)(k+2)}{6}\right]\cdot 2^k-\left[(k^2-k+2)\times 2^{k-1}-1\right]$$

$$=\frac{k^3+5k-6}{3}\times 2^{k-1}+1$$

得 $P_6(5)=\dfrac{5^3+5\times 5-6}{3}\times 2^{5-1}+1=769$

且 $f_7(5)=\dfrac{5(5+1)(5+2)(5+3)}{4!}=70$

$f_8(5)=\dfrac{5(5+1)(5+2)(5+3)(5+4)}{5!}=126$

由 $H_n(m)=\left[m-f_{n+1}(k)+f_n(k)\right]\times 2^k-P_{n-1}(k)$

可得 $H_7(200)=[200-f_8(5)+f_7(5)]\times 2^5-P_6(5)=[200-126+70]\times 2^5-769=3839$ #

伍、結論

1. 對於 n 柱河內塔的移動，當完成遊戲，其過程必存在「**半移動**」。
2. 對於 n 柱河內塔的移動，皆可以以「**對稱性操作**」來完成。
3. 利用「滿格建構」，可造出「捷徑半移動」。
4. 利用「捷徑半移動」，可作出移動捷徑。
5. 從「滿格數量關係表」中，作斜向觀察，可得排列形成「**巴斯卡三角圖形**」。
6. 利用「滿格數量關係表」，我們推導出 $H_n(m)$ 的通式：

$$H_n(m)=\left[m-f_{n+1}(k)+f_n(k)\right]\times 2^k-P_{n-1}(k) \qquad (n\geq 4)$$ ------ [n 柱公式]

7. 利用 $H_n(m)$ 的通式，可得

(1) $H_4(m)=\left[m-\frac{1}{2}(k^2-k+2)\right]\times 2^k+1$ ------- [4 柱公式]

(2) $H_5(m)=\left[m-\frac{1}{6}(k^3+5k-6)\right]\times 2^k-1$ ------- [5 柱公式]

(3) $H_6(m)=\left[2m-\frac{1}{12}(k^4+2k^3+11k^2-14k+24)\right]\times 2^{k-1}+1$ ------- [6 柱公式]

陸、參考資料：

1. 作者：陳璿、林亮辰、曹瑀、吳侑璋 (民 98)
   中華民國第 49 屆中小學科學展覽國中組數學---
   作品「三柱輪換之移動策略---雞尾酒法」。
2. 作者：Ben Houston & Hassan Masu (2004) “ Explorations in 4-peg Tower of Hanoi ”
3. 作者：黃祈昌、田慈安、陳福誌、王文忻(民 99)
   中華民國第 50 屆中小學科學展覽國中組數學---
   作品「 n 柱河內塔的策略研究與最佳化通式的尋找」。

柒、附錄　（感謝國立[redacted]大學[redacted]教授的提供）

Explorations in 4-peg Tower of Hanoi

Ben Houston　　Hassan Masum
exocortex.org　　hmasum.com

November 29, 2004

Abstract

Finding an optimal solution to the 4-peg version of the classic Tower of Hanoi problem has been an open problem since the 19th century, despite the existence of a _presumed-optimal_ solution. We verify that the presumed-optimal Frame-Stewart algorithm for 4-peg Tower of Hanoi is indeed optimal, for up to 20 discs. We also develop a distributed Tower of Hanoi algorithm, and present 2D and 3D representations of the state transition graphs. Finally, two variants (k-out-of-order and k-at-a-time) and an analogy with distributed agent software are suggested.

1 Introduction: History and Overview

The Tower of Hanoi is a well known problem in recreational mathematics; it has also been widely used in computer science as a paradigmatic teaching example for recursive solution methods. In the basic version, a stack of discs of mutually distinct sizes is arranged on one of three pegs, with the size restriction that no larger disc is atop a smaller disc. The problem is then to move the entire stack of discs to another of the three pegs by moving one disc at a time, and always maintaining the size restriction.

As the instruction sheet for the puzzle (_rst sold in 1883) said: "*According to an old Indian legend, the Brahmins have been following each other for a very long time on the steps of the altar in the Temple of Benares, carrying out the moving of the Sacred Tower of Brahma with sixty-four levels in _ne gold, trimmed with diamonds from Golconde. When all is _nished, the Tower and the Brahmins will fall, and that will be the end of the world*"

It is well known that this basic version is solvable in 2_n-1 moves. But in generalizing the problem to four or more pegs, the analysis gets much harder. (Details and variants of the 4-peg problem can be found in [Stockmeyer 1994].) There is a solution method - the Frame-Stewart algorithm or _presumed-optimal solution_ - which is the best known for 4 pegs, but has not been proved to be optimal. A lower bound to the order of magnitude of moves required is given in [Szegedy 1999], and these results are sharpened in[Chen and Shen 2004] to show that the presumed-optimal solution has the same order of magnitude as the actual optimal solution.

Since no optimal solution is known, we have veri_ed that the presumedoptimal solution is indeed optimal for up to 20 discs, via an exhaustive search of the possible moves. Our work builds

on that of [Hinz and Bode 1999],who found that the presumed-optimal solution was in fact best for up to 17 discs. We verify and extend their results, and give results for 5 and 6 pegs as well.

Two further approaches toward solving the problem are discussed. First, the solution method can be modi_ed to work on a distributed network of machines, which could allow veri_cation for larger numbers of discs. Second, we construct 2D and 3D representations of the state transition graphs for various Tower of Hanoi puzzles, which may help point out symmetries and suggest potential lines of attack on the general problem. Finally, we close with suggestions for k-out-of-order and k-at-a-time variants, and an analogy between the Hanoi problem and future distributed agent software.

2 Verifying 4-peg Tower of Hanoi

2.1 The Presumed-Optimal Solution

The presumed-optimal solution for 4-peg Tower of Hanoi (henceforth referred to as 4-peg ToH) has the following general form:Step 0 Number the pegs from 0 to 3. Our goal is to move all discs frompeg 0 to peg 3, respecting the constraints that only one disc moves at a time, and a smaller disc is never under a larger disc.

Step 1 Move the smallest d-k discs from peg 0 to peg 1, using all 4 pegs.

Step 2 Move the largest k discs from peg 0 to peg 3, without using peg1. (In other words, move these discs using the optimal 3-peg method,which is already known.)

Step 3 Move the smallest d-k discs from peg 1 to peg 3, using all 4 pegs.Denoting by T(d, p) the minimal number of moves for a d-disc problem on p pegs, we see that the above method implies a recursive solution:$T(d; 4) = 2 * T(d ¡ k; 4) + T(k; 3)$

Analysis of this recursion results in a closed-form solution for the minimizing value of k. One can also derive closed-form expressions for the total number of moves required (see [Stockmeyer 1994] and [Klavzar and Milutinovic 2002]).

However, this solution is only an answer to the original problem(how many moves does a 4-peg ToH require) under the assumption that theoptimal solution has the general structure given above. Despite a century of exploration into the problem, the optimality of the above solution has not been proven. It is therefore worthwhile to actually try all possible solutions to 4-peg ToH, to verify by demonstration that no shorter method exists.

【評語】030410

本作品係將中小學 50 屆科展作品未完整之部分加以延伸解決，基本上可以被視為一個兩年期的長期團隊合作作品，結論方面比起去年的作品明顯豐富，然而結論第 7 點 Hn(m)的通式明顯和△k 有關，而△k 的決定，並未有一個簡易有效的方法，是不足之處，考量三人團隊合作長久無間，決定頒于團隊合作獎。