投稿類別：數學類

篇名：
分類最優化---猜數字遊戲的最佳策略

作者：
何俊緯。高雄市立高雄高級中學。高一 24 班
吳政軒。高雄市立高雄高級中學。高一 24 班
陳光穎。高雄市立高雄高級中學。高一 24 班

指導老師：
江國宏老師

壹●前言

「幾 A 幾 B」的數字猜謎遊戲，目的是利用數字與排列順序正確與否的暗示，迅速猜出 4 個不重複而隨機排列的數字組合。換句話說，當玩家隨意猜出 4 個不同數字排列時，若數字與位置皆相同，則會得到 A 的反應。若數字相同，但位置不同時，則會得到 B 的反應。遊戲進行時，每個人都可能有不同的遊戲邏輯或技巧。例如：大部分的人會先將每個數字都答一遍後，再慢慢整理其數字位置；匿名為天才者，則提出使用先鎖定是哪些數字，再決定數字的順序；盡量不要讓數字順序重複，如此可以一邊決定數字、一邊決定順序的猜測數字方法(天才，2005)。我們曾經利用天才所設定的方法試驗，結果平均每次遊戲大概要猜 6~8 次，才會猜到答案。有一種玩法，是每次猜測後，將可能符合條件的所有數字排列組合，逐一猜測，並在每次猜測後，排除不可能存在的排列組合(匿名者，2005)。我們亦利用匿名者的方法試驗，結果有時需要 9 次。有匿名為南洋大兜蟲者，利用 C 程式語言自製演算法，保證在 7 次內猜出答案(南洋大兜蟲，2008)。但謝名傑具名回應有例外的數字排列組合，會超過 7 次以上的猜測(謝名傑，2015)。此後，就無搜尋到，對此遊戲有具體猜測方案的資訊。

我們不禁疑問是否有一種方法，會比目前已知演算法更為快速？存在比網路上流行的技巧或猜測方法，有更好的演算法？是否真的存在一個演算法可以保證在 7 次以內猜得出來的演算法？分析目前所查詢的方法，重點都在歸納與排除法。因此，我們假設若配合邏輯篩選，則可能會出現有更快速的猜測演算法。

貳●正文

一、遊戲規則

0-9 共 10 個數字，參與者必須從中選出 4 個不同的數字，而系統會給兩個傳回值，其一是 A，代表有幾個數字與位置接正確；另一為 B，代表有幾個數字正確，位置錯誤。而遊戲目的便是從現有資料中用最少的次數中猜出正確答案為何。

二、名詞解釋

(一)最佳解

猜測演算法所算出的最好的下一個猜測，以及所有和其等價的

猜測(見下文「猜測的等價」)的集合，稱之為「最佳解」。演算法的輸出值∈最佳解。

(二)回傳值

猜測任何一個線索皆能獲得 A，B 之線索，而此稱為回傳值。

(三)演算法等價

當有兩種演算法保證猜中的次數一樣多時，稱此二演算法等價

三、目前已知的演算法

(一)從數字到順序

這種演算法目的在縮小範圍，先求出四個數字，過程中藉由數字位置的改換確定其順序，這種演算法考驗的是玩家的邏輯、猜測的策略性。但其不需艱深的理論與深厚的數學基礎，因此這也是目前最普遍的猜法。

(二)刪去法

這種演算法利用窮舉的方式，列出所有可能答案，再從可能答案中隨機挑一個當作猜測。這種方法雖然必須使用程式計算，但因其沒什麼技巧，因此成為了程式設計很經典的例題。

四、演算法的缺點

(一)定義不夠明確：從數字到順序僅憑著直覺猜測，刪去法則是隨機猜測，兩者皆沒有一個嚴謹的方式。

(二)從數字到順序中是利用個人的邏輯猜測，猜測的方式因人而異

(三)刪去法隨機猜測，猜測的方式因時而異

(四)可能答案的效能未必較佳：詳見下文「分類」

五、本研究的演算法

由於網路上的演算法都不能保證 7 次以猜中，因此我們嘗試提出一種新的演算法，以下是這種演算法的說明

(一)「篩選」與「剩餘可能答案」

綜觀目前網路上查得到的演算法，重點都在於利用每一次猜測時所得到的 A 和 B 個數，排除不可能是答案的排列組合，因此我們將每一次的猜測稱為一次「篩選」。每一次篩選都會排除掉一些不可能是答案的排列組合。而每一次篩選後所剩餘的可能是答案的數字排列所構成的集合，我們定義為「剩餘可能答案」。第 N+1 次猜測的剩餘可能答案必為第 N 次的剩餘可能答案的子集。

(二)分類

除了篩選的概念之外，我們另外建立一個概念，就是分類。每一次的猜測，都會得到A和B的回覆，回覆共有14種，分別為0A0B、0A1B、0A2B、0A3B、0A4B、1A0B、1A1B、1A2B、1A3B、2A0B、2A1B、2A2B、3A0B、4A0B。猜測後得到哪一種回覆，將取決這次篩選過後的剩餘可能答案。演算法在正式輸出決定要做哪一個猜測之前，不會知道接下來會得到這 14 種回覆中的哪一種，但是在正式輸出前，仍可以對將會得到的回覆和將會得到的剩餘可能答案做假設。假設這個猜測將會得到 1A2B 的回覆，而這個回覆使這次猜選後的剩餘可能答案，我們稱為「被分類到 1A2B 的答案」。被分到 14 類各類的答案為兩兩互不相交的集合。演算法在輸出之前可以先模擬出被分到各類的答案個數。模擬出來的結果，如: 被分類到 0A0B 的答案共 76 個，被分類到 0A1B 的答案共 80 個……的機率密度分布，稱為「分類情況」。而在正是輸出後，其傳回值便是間接的告訴我們正確答案在哪一類裡面

(四)分類標準

由於不同的猜測數字將取決不同的分類情況，我們將猜測的數字稱為「分類標準」

(三)猜測的方式

如前所述，數字的分類情況會依據分類標準而定。此時所猜的

數字就便扮演了很重要的角色

舉例來說，如果某次遊戲的進行中，現有的條件為 0123 0A1B 1456 1A1B　2406 1A1B　3478 0A2B，我們可以推得其所有可能解為 2754　2854　4396　4936　9346(程式計算)，而怎麼樣的猜測是我們的演算法的最佳解？答案不在這5個數裡面，而是0397(程式計算)。理由是，當做了這個猜測後，如果答案是 2754，其傳回值為 0A1B(2754 被分在 0A1B 的這一類)；如果答案是 2854，其傳回值為 0A0B(2854 被分在 0A0B 的這一類)；如果答案是 4396，其傳回值為 2A0B(4396 被分在 2A0B 的這一類)；如果答案是 4936，其傳回值為 0A2B(4936 被分在 0A2B 的這一類)；如果答案是 9346，其傳回值為 1A1B(9346 被分在 1A1B 的這一類)，這樣只要兩次就可以猜到正確答案了(0391 一次、正確答案一次)。反之，若是以其他猜測，則不會有如此效果。以 2754 為例，這個猜測雖有一次猜中的機率，但 4396　4936　9346 會被分在同一類，如果程式傳回 0A1B，顯然猜測的次數是比最佳解多的。那麼，究竟什麼樣的猜測是最佳解？按照前面的分類論，0397 把 2754　2854　4396　4936　9346「平均」的分成五類(實際上是 14 類，只是有些是空集合)，而 2754 卻只將 2754 與 2854 分出，4396　4936　9346 則被分在同一類，因此，我們給最佳解下一個定義：能把數字分的最「平均」者，即是最佳解。

(四)平均的定義

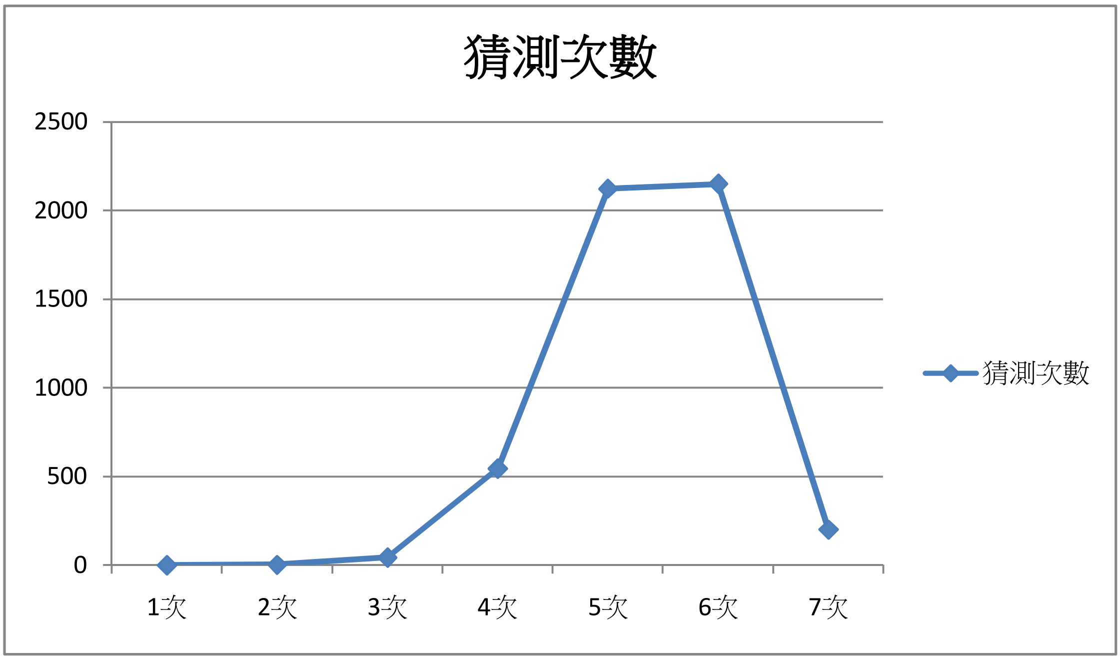
上文中提到「能把數字分的最平均者，即是最佳解」。那麼，何謂「平均」？我們就我們的目的入手。我們所探討的是在幾次以內「必然」猜中，也就是不用考慮機率分布的問題。因此我們要討論就是「運氣最差」的時候，具體一點就是「每次答案都在所有分類中元素最多的那一類」，而我們要降低猜測的次數，首要之務便是把這最多元素的元素量降到最低。因此，我們給平均下一個定義：「使元素量的最高峰降至最低」

(五)猜測的等價

上文中對平均的定義為最高峰元素量最低時，然而，當有兩種猜測其最高峰元素數量相等時，我們稱其等價，而當兩種猜測等價時，我們一律依數字遞增排序順序最前者為演算法的輸出值。

六、結果

我們使用我們的演算法來測試其效能，我們一一測試 0123~9876 這 5040 個答案，測試結果如下：



圖一：猜測次數測試結果

202 組答案需要猜 7 次.
2151 組答案需要猜 6.次
2124 組答案需要猜 5.次
515 組答案需要猜 4.次
44 組答案需要猜 3.次
3 組答案需要猜 2.次
1 組答案需要猜 1.次
最高紀錄: 7 次，機率 4%(202/5040)

七、演算法的瑕疵

在這種演算法中，我們定義當兩種猜測其最高峰元素數量相等時兩種猜測等價。然而，這樣的定義是有瑕疵的，理由是：我們定義了最高峰，卻忽略其第二大的值等其他可能影響「平均程度」的因素。再者，每次的分類也都會影響到下一次的分類，單依「平均程度」判斷等價是不嚴謹的。我們期待在下一次的研究中，這方面可以做得更完善

八、演算法的最低次數

在找出七次必能猜中的演算法後，我們想知道是否存在更少次數的

演算法。考慮任何的演算法都可以看成一種分類，而最佳的猜測必為其最平均者，而完全平均者稱理想狀態。因此，我們計算在理想狀態下幾次可以保證猜中。

對所有演算法，在第一猜測中，不管分類標準為何，必會有 360 個答案被分到 0A0B 這一類、1440 個答案被分到 0A1B 這一類、1260 個答案被分到 0A2B、264 個答案被分到 0A3B、9 個被分到 0A4B、480 個被分到 1A0B、720 個被分到 1A1B、216 個被分到 1A2B、8 個被分到 1A3B、180 個被分到 2A0B、72 個被分到 2A1B、6 個被分到 2A2B、24 個被分到 3A0B、1 個被分到 4A0B。其中 0A1B 這類的元素個數 1440 最多，4A0B 這類的元素個數 1 最少。假設有一個分類最有效率的演算法，則此演算法則可在第 2 次猜測將這 1440 個答案完全平均的分到 14 類，1440/14=102.8571，故會有 12 類分到 103 個，2 類分到 102 個。然而，這是不可能的，因為某些分類並無法容納如此大的數量，如 4A0B 中，元素量必然≦1(因為 4A0B 代表正確答案，而正確答案只有一個，如果被分在 4A0B 的元素量是 0，代表分類標準並不可能為正確答案)，同理，2A2B 最多只能容納 6 個元素，1A3B 最多只能容納 8 個元素，0A4B 最多只能容納 9 個元素，而在扣除這特定類別的元素的最大量限制後，我們在第二次的分類中結果如下

4A1B 有 1 個， 2A2B 有 6 個，1A3B 有 8 個，0A4B 有 9 個，3A0B 有 24 個，2A1B 有 72 個，其餘皆為 165 個，最大值 165。
註：(1440-1-6-8-9-24-72)/(14-6)=165

而接下來我們反覆運用這種方式，第三次猜測結果: 4A1B 有 1 個，2A2B 有 6 個，1A3B 有 8 個，0A4B 有 9 個，其餘有一類 15 個，9 類分到 14 個。第四次猜測結果: 有 1 類分到 2 個，13 類分到 1 個。第五和第六次猜測就猜第四次的兩種可能答案。

所以，即便是分類最有效率的演算法，也需要分類猜 6 次才猜得到答案。因此，我們得出一個結論：不存在 5 次必然猜中的演算法。

參●結論

針對網路上的演算法，首先我們改良了其定義不周全的問題，定義出一種可以明確界定且不因時因人而異的演算法猜測。我們認為每一種演算法都是一種分類，也就是說所謂猜測其實是分類的問題。在這個基礎下，我們首先定義了何謂最佳的分類，以及平均的定義，制訂出了這一套新的演算法。

而後，我們將這一套演算法化為程式碼，利用窮舉的方式，讓電腦從 0123 猜到 9876，而其結論為：這種演算法可以保證在 7 次內猜對，而需要猜到 7 次的機率約 4%，這便間接的證明了我們的演篹法是比網路上其他演算法都還要優良的

接著，我們利用「猜測即是分類」的基礎概念，將平均擴展到極致，討論最理想的狀態下其最少次數為何，證明了不存在五次必然猜中的演算法。

然而，我們並沒有說明是否存在六次的演算法。因此，我們希望未來可以再深入研究，即找出 6 次必然猜中的方式(我們考慮定義一個新的函數，可以判斷某次分類中同一子集的相似度，相似度較低者，可能對下一次的分類有幫助)，或證明這種演算法不存在。

肆●引註資料

1.張垚(2011)。**數學奧林匹亞小叢書--組合數學(第 2 版)**。上海市：華東師範大學出版社

2.葉鴻國(2009)如何做出最佳選擇--組合最優化與線性規畫。**科學月刊，40**(12)，頁 779-784

3.～C 語言程式設計～電腦猜數字～。104 年 2 月 21 日，取自 http://blog.xuite.net/funnyfamily/tonyjan/21045725-%EF%BD%9EC%E8%AA%9E%E8%A8%80%E7%A8%8B%E5%BC%8F%E8%A8%AD%E8%A8%88%EF%BD%9E%E9%9B%BB%E8%85%A6%E7%8C%9C%E6%95%B8%E5%AD%97%EF%BD%9E

4.猜數字的技巧。104 年 2 月 21 日，取自 https://tw.knowledge.yahoo.com/question/question?qid=1405122211286

5.關於「幾 A 幾 B」遊戲的程式碼。104 年 2 月 21 日，取自 https://tw.knowledge.yahoo.com/question/question;_ylt=A8tUwZbO6NBUK30AZwtr1gt.;_ylu=X3oDMTE0ZzBjdDczBHNlYwNzcgRwb3MDMgRjb2xvA3R3MQR2dGlkA1ZJUFRXODJfMQ--?qid=1005030401632