僞幣問題之改良演算法設計與分析

The Designs and Analyses of Improved Algorithms for the Counterfeit Coins Problem

劉耀才　　林順喜*

國立台灣師範大學

資訊工程學系

台北市 117 大安區

linss@csie.ntnu.edu.tw

Yao-Tsai Liu and Shun-Shii Lin

Department of Computer Science and Information Engineering,
National Taiwan Normal University
Taipei City 117, Taiwan

linss@csie.ntnu.edu.tw

Received 20 September 2006; Revised 29 November 2006 ; Accepted 1 February 2007

摘　要

僞幣問題由來已久，有許多研究者考慮不同的條件，使得這個問題變得更具挑戰性也更加困難，也有許多人嘗試著提出各種不同的演算法去解決這些不同形式的僞幣問題。在本論文中，我們對兩枚僞幣不知其輕重、三枚僞幣知其輕重、三枚僞幣不知其輕重、四枚僞幣知其輕重、四枚僞幣不知其輕重等問題提出了改良的演算法，以及改進了李立中的三枚以上僞幣知其輕重演算法，使之成爲三枚以上僞幣不知其輕重的演算法。在最後我們也對一枚僞幣知其輕重、一枚僞幣不知其輕重、兩枚僞幣不知其輕重、三枚僞幣知其輕重、三枚僞幣不知其輕重、四枚僞幣知其輕重、四枚僞幣不知其輕重等問題，提出了分析，說明各個演算法相對於理論下限還有多少可以努力的空間。

關鍵詞：下限分析、僞幣問題、三分法

ABSTRACT

The counterfeit coin problem is a well-known problem. Many researchers have tried to make the problem more challenging by considering some constraints for the problem. There were also a lot of researchers presenting different algorithms for variants of the problem. In this paper, we propose some improved algorithms and strategies to solve some kinds of the counterfeit coin problems, including the 2-counterfeit coins problem with unknown weights, the 3-counterfeit coins problem with known weights, the 3-counterfeit coins problem with unknown weights, the 4-counterfeit coins problem with known weights, the 4-counterfeit coins problem with unknown weights. We also tackle the k-counterfeit coins problem with unknown weights by improving the algorithm proposed by Li-Jhong Li, in which he only dealt with the k-counterfeit coins problem with known weights. In addition, we provide the analyses of the algorithms for these counterfeit coins problems. According to the analyses, we know the theoretical lower bound of the numbers of weightings to identify the counterfeit coins in a mass of coins. Thus, we can know which strategy of the problem might be further improved.

Keywords: Lower bound analysis, The counterfeit coins problem, Trisection method

* 通訊作者

一、簡介

僞幣問題（The Counterfeit Coin Problem）是一個很著名的古老問題，有許多人對這個問題也做出了不少的研究以及分析。所謂的僞幣問題就是指在一堆外形都相同的硬幣之中，裡面有數枚僞幣，而這些僞幣除了重量與正常硬幣不同之外，其餘都跟正常硬幣無異，而僞幣的重量彼此也都是相同的。現在我們手上有一個只能分辨輕重但沒辦法秤出重量的天秤，應用這個天秤在最少的秤量次數中，把僞幣都找出來。然而，某些研究允許僞幣彼此之間重量可以不同，但也遵循著一定的規範，比如說現在有兩種重量的僞幣，一種比正常硬幣重，另一種比正常硬幣輕，但從這兩種僞幣中各挑出一個的重量和要等於兩枚正常硬幣的重量和。而就僞幣在一群硬幣中所佔的個數不同、僞幣較正常硬幣輕或是重、以及僞幣在這群硬幣之中的個數等事實的知道與否，可以分爲以下幾種類型，本論文將有整理以及討論，並且針對其中一些部分提出新的改良演算法。

最單純的一種，就是已經確知僞幣只有一枚，且也已知僞幣是較正常硬幣重或是輕，這類問題大部分人都可以想出以「三分法」來比較。

另一種類型就是僞幣數也是一枚，但不能確知僞幣較正常硬幣輕還是重，這種類型就需要比較特別的演算法才能夠比較出來，在「演算法的天空」[12]一書中有提出演算法來解決這類型的問題。

在僞幣數爲兩枚，且已知僞幣重量較正常硬幣重或輕的情形下，Pyrich [3]曾在網站上提出了一套找出這兩枚僞幣的演算法。在僞幣數一樣是兩枚，但不能確知僞幣較正常硬幣輕或是重的情形下，李立中曾對這類型的問題提出「交錯秤量法」[10]。

當然對於僞幣問題，最一般化（General case）的類型是，既不知道僞幣的數目，也不能確定僞幣較其他正常硬幣輕或是重。這種類型目前只有在 1994 年 Hu 等人[9]及 1997 年 Wan 等人[6]有提出過一些演算法，不過也還未找到最佳解。

在本論文中，我們將會針對僞幣數爲二、三、四枚，且僞幣較正常硬幣重或是輕爲已知或是未知的情形，分別提出演算法來找出僞幣。另外本論文也會針對李立中所提出的在三枚以上且已知僞幣較正常硬幣輕或是重的一般化演算法[10]加以改良，成爲可以不必知道僞幣較正常硬幣輕或是重也能找出僞幣的一般化演算法。

二、研究背景及目的

2.1 研究背景

僞幣問題最早是由誰提出來的，大概已經是不可考的問題了。有一說是在很久以前的時代，所羅門王命令鑄錢師鑄造出十二枚純金的金幣，來展示他的國威，但是這個鑄錢師卻偷工減料，把其中一枚硬幣偷偷的用銀作爲原料，再鍍上黃金，想要藉此騙過所羅門王，然而，最後所羅門王展現了他的智慧，只用了一把天秤，秤了三次就把僞幣找了出來。

後來隨著時間的演進，這個問題也有了很多種不同的類型。例如，不知道僞幣的數量，或者是不知道僞幣的輕重，當然，問題所給定的限制條件越少，所需的秤量次數將會越多。因此

我們將對各種不同條件所形成的不同問題類型，分別的加以討論，在這邊所提到的僞幣所指的就是與正常硬幣重量不相等的那些硬幣，而一般說來那些佔少數的硬幣，就被我們稱之爲僞幣。

若是依照一開始秤量時所知的條件：僞幣數是否已知、是否知道僞幣較正常硬幣是輕或是重，則我們就可以把此問題分爲四大類。在秤量者已經知道僞幣數量時，可以依照是否已知僞幣較正常硬幣輕或是重，來將此類問題加以分類，本論文就是針對此類型來討論。另外在僞幣數目不能確切知道的情形下，我們將不會討論此類問題。

事實上我們也可以把僞幣問題看成是一種演繹遊戲（Deductive games），也就是以特定的方法做測試，然後從測試的結果中得到某些資訊（在僞幣問題中，就是經由秤量的過程，得知一堆硬幣和另一堆硬幣間的輕重關係），而玩家（或稱推論者）則藉由這些資訊去推論出其背後所隱藏的資訊（在僞幣問題中就是找出那些和正常硬幣重量不同的僞幣，以及那些僞幣較正常硬幣輕還是重）。其他著名的演繹遊戲有 Mastermind[11]、AB Game[11]以及 Ulam's game[8]等。

2.2 一般化問題的相關研究

最早研究一般化問題（General case）的 Hu、Chen、Hwang 三人[9]，曾提出一個秤量的標準來比較秤量硬幣演算法的優劣，我們在此將做個簡單的介紹。

在他們的論文中定義了幾個表示秤量次數的函數：其中定義$W(n,d)$表示從 n 枚硬幣之中找出 d 枚較重的僞幣，所需的最少秤量次數；另外也定義了$W(n:d)$表示秤量者並不知道 d 是多少的情形下，從 n 枚硬幣之中，找出 d 枚較重僞幣，所需的最少秤量次數。這是因爲我們目前尚未有一最佳的演算法來從各種情況，在最少的次數內找出僞幣，所以就不知最少次數是多少，因而使用此函數來表示最少次數。從這邊我們可以很直覺的看出$W(n:d) \geq W(n,d)$。

另外也定義了$W_G(n:d)$則是以某個演算法 G，來從 n 枚硬幣中找出 d 枚僞幣，而在一開始時又不知道僞幣數目，所需的最少秤量次數。最後得到如下的式子：

對於所有 d 而言，$W_G(n:d) \leq c * W(n:d) + b$，其中 b、c 均爲常數。

其中 c 這個常數是用來比較各演算法間好壞的主要因素，可以看出假如 c 越小，則表示此演算法 G 所需的秤量次數也越少，就代表這個演算法越好。這個表示法在本論文中也會使用到。

除此之外我們也會引用到他們論文中的一些表示法：假設有一堆硬幣 A，|A|表示 A 這堆硬幣中的硬幣個數，||A||則表示 A 這堆硬幣的總重量。

2.3 一枚僞幣問題的相關研究

在僞幣數目只有一枚時，且僞幣較正常硬幣是輕還是重也是已知的情形下，這方面目前還沒有人特別提出論文，因爲這類問題大家都能以最直覺的「三分法」來找出僞幣，這個方法是在一開始把硬幣分成三堆，將其中兩堆放到天秤去秤量，假如秤出來的結果是有一堆較重，那代表僞幣是存在於較重的那堆裡面，若是秤出來的結果是等重的，那代表僞幣是不在天秤上的那兩堆中，然後我們就可以再把有僞幣的那堆再三分下去，遞迴的施行此方法，每次都可以把問題縮減爲原來的三分之一，也因此我們可以得到這個關係式：$W(n,1) = \lceil \log_3 n \rceil$。

而在僞幣跟正常硬幣的重量關係未知的情形下，在「演算法的天空」一書[12]中，對此類

問題提出了演算法（由於其中的演算法頗爲複雜，有興趣的讀者可以在「演算法的天空」[12]找到詳細的說明）。

假如我們以之前的表示法來表示秤量次數的話：

若僞幣較正常硬幣重則秤量次數爲$W(n,1)$；

若僞幣較正常硬幣輕則秤量次數爲$W(n,n-1)$。

在此我們針對僞幣較正常硬幣輕或重爲未知的情形，沿用了李立中[10]的定義方式：$S(n,d)$表示在僞幣較正常硬幣輕或重爲未知的情形下，從 n 枚硬幣中，找出 d 枚僞幣，所需的最少秤量次數。

依此定義方式，我們可以把「演算法的天空」一書[12]中所提到的演算法的秤量次數表示成$S_A(n,1)$。在網站[13]中提出了$S_A(n,1)$和 n 之間關係的分析，他假設在秤量 k 次後能在一堆硬幣，硬幣數爲$T(k)$中找出一枚不知輕重的僞幣：

$T(2)=3;\quad k>2,\quad T(k)=3(T(k-1)+1)$。

由這個關係式我們可以推得：$S_A(n,1)=\lceil \log_3(2n+3)\rceil$。

另外在 Lorenz 跟 Noebert 的一篇論文[4]中，在僞幣數爲一枚，僞幣較正常硬幣輕或是重爲未知，但有多個天秤可同時秤量的情形下，提出了下面的關係式：

$$n \le \frac{(2b+1)^w-1}{2}-b \ ,$$

其中 n 爲硬幣總數、b 爲天秤個數、w 爲秤量次數。

當我們把 b 設成 1 時，可得到如下關係式：

$w \le \lceil \log_3(2n+3)\rceil = S_A(n,1)$。

所以我們可以看出「演算法的天空」一書[12]中所提出的演算法，已經幾乎跟理論上的 lower bound $\lceil \log_3 2n\rceil$ 相等了。

2.4 兩枚僞幣問題的相關研究

在兩枚僞幣且知道僞幣輕重的情形下，在 Pyrich[3]的網站中，有提出解決此類問題的演算法，他的演算法主要是先把硬幣分成三堆：A、B、C，其中它們的個數關係爲：|A|=|B|, |（|A|-|C|）| = 1，也就是說 C 堆可能比 A、B 兩堆多一個或者少一個，再依照該演算法找出僞幣。在這裡我們不詳細說明 Pyrich[3]的演算法，只列出此演算法的秤量次數（有興趣的讀者可以參考 Pyrich[8]的網站或者是李立中[10]的論文中有詳細的說明），依照李立中[10]對 Pyrich[3]的演算法所做的分析，可以得出下列式子：

$$W_{Pyrich}(n,2)=Max\left\{W_{Pyrich}\left(\frac{n}{3},2\right)+2,2W\left(\frac{n}{3},1\right)+2\right\}$$

$$=2W\left(\frac{n}{3},1\right)+2=2\left\lceil \log_3\frac{n}{3}\right\rceil+2=2\lceil \log_3 n\rceil$$

在兩枚僞幣，且未知僞幣輕重的情形下，李立中對此類問題提出了「交錯秤量法」[10]，他的演算法的主要概念是：交錯的把硬幣分成兩堆（二分法）或者是三堆（三分法），並把現有

已知的演算法（一枚硬幣已知/未知輕重、兩枚硬幣已知輕重）套入其中，在此我們也不詳細說明此演算法（有興趣的讀者可以參考李立中[10]的論文），此演算法的秤量次數分析如下：

$$S_w(n,2)=Max\left\{S\left(\frac{n}{2},1\right)+W\left(\frac{n}{2},1\right)+1,W_{Pyrich}\left(\frac{n}{6},2\right)+3,W\left(\frac{n}{3},1\right)+W\left(\frac{n}{6},1\right)+5,\right.$$
$$\left.2W\left(\frac{n}{6},1\right)+4,W_{Pyrich}\left(\frac{n}{6},2\right)+4,W_{Pyrich}\left(\frac{n}{3},2\right)+5,W\left(\frac{n}{12},1\right)+W\left(\frac{n}{6},1\right)+5\right\}$$
$$=Max\left\{\left\lceil\log_3(n+3)\right\rceil+\left\lfloor\log_3\frac{n}{2}\right\rfloor+1,2\left\lfloor\log_3\frac{n}{2}\right\rfloor+1,\left\lceil\log_3 n\right\rceil+\left\lfloor\log_3\frac{n}{2}\right\rfloor+3,2\left\lfloor\log_3\frac{n}{2}\right\rfloor+2,\right.$$
$$\left.2\left\lfloor\log_3\frac{n}{2}\right\rfloor+2,2\left\lfloor\log_3\frac{n}{2}\right\rfloor+3,\frac{3}{2}\left\lfloor\log_3\frac{n}{2}\right\rfloor+3\right\}$$
$$=2\left\lceil\log_3 n\right\rceil+3$$

另外在兩枚僞幣且未知僞幣輕重的情形下，Liu 跟 Nie[7]也針對此問題提出了數學上的證明，他們的觀點是從秤量次數下手的，也就是以天秤秤了 k 次，最多能從多少硬幣中，找出兩枚未知輕重的僞幣。然而就秤量次數而言，在理論上秤了 k 次，會有一個 Upper bound（$U(k)$）來表示最多可從$U(k)$枚硬幣中找出兩枚僞幣來，另外假設有一個方法能在 k 次秤量中，可以從最多$N(k)$個硬幣中找出兩枚僞幣的話，很直覺的可以看出$N(k)\le U(k)$，而他們就是證明出：當 k 爲偶數時，$N(k)=U(k)$；當 k 爲奇數時，$N(k)\approx U(k)$，且$N(k)$可以一直逼近$U(k)$。

2.5 三枚以上僞幣問題的相關研究

在三枚以上的僞幣問題中，在已知僞幣較正常硬幣輕或是重的情形下，李立中[10]在他的論文中也提出了一個一般化的方法來解決此類問題，在他的論文中稱之爲 Find 演算法，這個方法主要是以遞迴的方式產生可能的僞幣分佈表格，然後藉由秤量所得到的資訊與每堆硬幣總數的限制來消除表格中不可能的分佈情形，如此遞迴執行 Find 直到找出所有僞幣爲止。（在此我們並不詳細解說此演算法，有興趣的讀者可以參考李立中的論文[10]）。

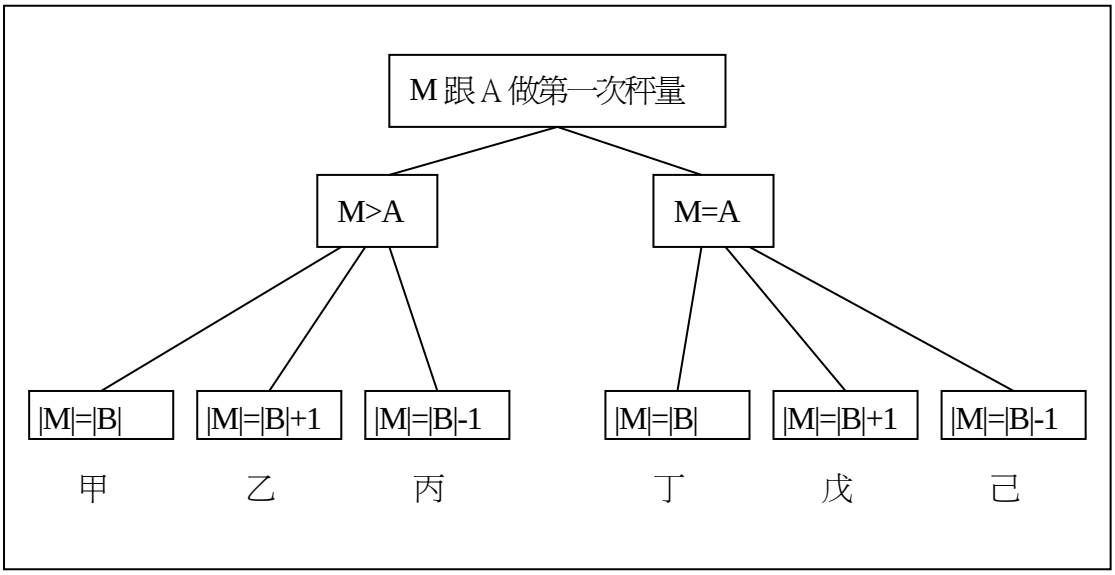
三、 我們提出的新方法與改良的方法

在開始說明我們所提出的演算法之前，先說明我們所提出的演算法的架構，我們所提出的演算法皆是以此架構爲出發點，然後各自分成六個分支來做討論，我們將以「甲、乙、丙、丁、戊、己」來做爲敘述這六個分支時所用的代號。

我們的演算法在一開始時將會把 n 枚硬幣分成三堆，分別叫做 M、A、B，而這三堆的硬幣個數的關係如下：

|M| = |A|,　|M| - |B| = 0 、 1 或 -1

也就是 M 的硬幣個數 = A 的硬幣個數，而 B 的硬幣個數可能比 M 多一個、少一個或是相同，我們可以注意到這三堆硬幣的數量各自均≤$\left\lceil\frac{n}{3}\right\rceil$，當$n=3p$時，$|M|=|A|=|B|=\frac{n}{3}$；當



圖一：改良演算法架構圖

$n=3p+1$ 時，$|M|=|A|=p=\left\lfloor\frac{n}{3}\right\rfloor$，$|B|=p+1=\left\lceil\frac{n}{3}\right\rceil$；當 $n=3p+2$ 時，$|M|=|A|=p+1=\left\lceil\frac{n}{3}\right\rceil$，$|B|=p=\left\lfloor\frac{n}{3}\right\rfloor$。首先我們會把 M 拿來跟 A 做秤量，將會有三種情形：M>A、M=A、M<A，在這三種情形中，其中的 M<A，我們只要把硬幣堆 M 的角色跟 A 互換，就會等於 M > A 的情形，所以我們只會討論 M > A 跟 M = A 二種狀況。

由於 M、A 與 B 的硬幣個數的關係有三種，再搭配上第一次秤量的情形，總共就會有六種狀況要分析，圖一就是我們所提改良演算法的架構圖。

我們之後的討論會以符號（mab）來表示 M、A、B 的僞幣個數（分別爲 m、a、b）分布，這時的總秤量次數我們就可以寫成 $W(|M|,m)+W(|A|,a)+W(|B|,b)$，當 m、a、b 其中一項爲 0 時，該項的 W 函數值就爲 0，我們舉個例來看，假如僞幣分布爲（012）代表 M 中沒有僞幣，A 有一枚僞幣，B 有兩枚僞幣，而此時的秤量總次數就可以寫成 $W(|M|,0)+W(|A|,1)+W(|B|,2)$，由於 m 爲 0，所以就會變成 $W(|A|,1)+W(|B|,2)$，當然之後這個式子會再以 M、A、B 的硬幣個數代入後化簡成較簡潔的式子 。

在後面的討論中，我們也會遇到一種情形，假設僞幣分布爲（121），但此時我們不知僞幣是較正常硬幣輕或是重，我們可能會把僞幣個數是 1 的其中一堆拿出來做一枚僞幣未知輕重的演算法，在我們知道僞幣是較正常硬幣輕或是重後，剩餘的硬幣堆我們就會用已知輕重的演算法代入，照上面的例子來看，我們假設是把 M 拿來做一枚僞幣未知輕重演算法，此時的總秤量次數就會是 $S(|M|,1)+W(|A|,2)+W(|B|,1)$，當然之後也會根據 M、A、B 的硬幣個數代入後化簡成較簡潔的式子，在我們的論文中的討論將會直接以化簡後的簡潔式子來做表示。

3.1 兩枚僞幣未知輕重的演算法

首先我們對於兩枚僞幣，且僞幣輕重未知的情形下，提出了一個新的演算法。在前人的研究中，李立中曾對這個問題提出了「交錯秤量法」[10]，該演算法的最大秤量次數爲 $2\lceil \log_3 n \rceil + 3$。而在本論文中我們所提出的改良演算法的最大秤量次數爲 $2\lceil \log_3 n \rceil + 2$，距離此問題理論上的秤量次數下限 $\lceil 2\log_3 n \rceil$，大約還有兩次的差距。我們的演算法主要是以三分法爲基礎，把硬幣分成三堆後，再根據每堆的僞幣分佈特性，再決定接下來要如何秤量。在這個演算法當中，我們將會利用到已知的演算法（一枚僞幣已知輕重、兩枚僞幣已知輕重），所以，我們的演算法的主要目標是：找出僞幣在硬幣堆中的分佈情形，以及僞幣較正常硬幣輕還是重。當我們知道了這兩件事情之後，自然就可以套用現有已知的演算法來繼續找出僞幣了。一枚僞幣已知輕重的演算法，我們是採用「三分法」；一枚僞幣未知輕重的演算法，我們是採用「演算法的天空」一書[12]的方法；而在兩枚僞幣已知輕重的演算法，我們是採用 Pyrich[3]的演算法。接下來就來說明我們的演算法。

首先我們先把硬幣分成三堆，分別叫做 M、A、B，而這三堆的硬幣個數的關係如下：

$|M| = |A|, \quad |M| - |B| = 0 \ 、\ 1 \text{ 或 } -1$

也就是 M 的硬幣個數等於 A 的硬幣個數，而 B 的硬幣個數可能比 M 多一個或者是少一個或是相同。此時僞幣的可能分佈情形如表一所示，其中表格中最左方欄位的「+」表示僞幣較正常硬幣重，「-」表示僞幣較正常硬幣輕。

表一：兩枚僞幣可能的分布情形

	M	A	B
+	2	0	0
+	1	1	0
+	1	0	1
+	0	2	0
+	0	1	1
+	0	0	2
-	2	0	0
-	1	1	0
-	1	0	1
-	0	2	0
-	0	1	1
-	0	0	2

接下來，我們演算法的第一步就是 M 跟 A 秤量，如此一來會有三種結果：M>A、M=A、M<A，而其中的偽幣可能分佈情形如表二、三、四所示。

從表二與表三的表格中可以發現，我們只要把表三中的 M 跟 A 互換，就會等於表二的表格了，所以我們接下來只會討論 M>A 跟 M=A 的部分。

表二：第一步 M>A 時偽幣的可能分布情形

	M	A	B
+	2	0	0
+	1	0	1
-	0	2	0
-	0	1	1

表三：第一步 M<A 時偽幣的可能分布情形

	M	A	B
+	0	2	0
+	0	1	1
-	2	0	0
-	1	0	1

表四：第一步 M=A 時偽幣的可能分布情形

	M	A	B
+	0	0	2
+	1	1	0
-	0	0	2
-	1	1	0

表五：第二步 M>B 時偽幣的可能分布情形

	M	A	B
+	2	0	0
-	0	1	1

表六：第二步 M=B 時偽幣的可能分布情形

	M	A	B
+	1	0	1
-	0	2	0

接著我們演算法的第二步是由 M 跟 B 秤，但由於 B 的個數可能會大於、等於或者是小於 M 的個數，所以接下來總共會分成六個分支來做討論（M>A 且|M|=|B|、M>A 且|M|=|B|+1、M>A 且|M|=|B|-1、M=A 且|M|=|B|、M=A 且|M|=|B|+1、M=A 且|M|=|B|-1）。

分支甲：M>A, |M|=|B|

由於 M 的個數跟 B 的個數一樣，所以可以直接秤量 M 跟 B，秤量的結果會得到表五及表六。

（甲.1）我們先對表五做分析，在此我們會針對 B 是奇數或是偶數做不同的討論。

（甲.1.1）B 爲奇數時：我們從 B 拿出一個 b'，然後 B 分對半秤量。

（甲.1.1.1）假如天秤是不平的，那表示 b'是正常硬幣，且僞幣是較正常硬幣輕的，且僞幣的分布爲（011）。所以接下來我們只要對 A 跟 B 分別做一枚僞幣已知輕重的演算法就可以找出僞幣了，這種情形的秤量次數爲 $2W\left(\frac{n}{3},1\right)+3$ 。

（甲.1.1.2）假如天秤是平的：我們再從 B 拿出另一硬幣 b”，然後把 b'放回 B，B 分對半秤量。

（甲.1.1.2.1）假如天秤是不平的，那表示僞幣是較正常硬幣輕的，且 b’爲僞幣，僞幣的分布變爲（010），所以接下來我們只要對 A 做一次一枚僞幣已知輕重的演算法就可以找出僞幣，這種情形的秤量次數爲 $W\left(\frac{n}{3},1\right)+4$ 。

（甲.1.1.2.2）假如天秤是平的，那表示僞幣是較正常硬幣重的，且僞幣的分布爲（200），之後只要對 M 做一次二枚僞幣已知輕重的演算法就能找出僞幣了，這種情形的秤量次數爲 $W\left(\frac{n}{3},2\right)+4$ 。

（甲.1.2）B 爲偶數時：我們就可以直接將 B 分對半秤量。

（甲.1.2.1）假如天秤是不平的，我們就可以知道僞幣是較正常硬幣輕的，且僞幣的分布爲（011），此時只要再對 A 跟 B 各做一次一枚僞幣已知輕重的演算法，就能夠找出僞幣，這種情形的秤量次數爲 $2W\left(\frac{n}{3},1\right)+3$ 。

（甲.1.2.2）假如天秤是平的，我們就可以知道僞幣是較正常硬幣重的，且僞幣的分布爲（200），此時只要再對 M 做一次二枚僞幣已知輕重的演算法，就能夠找出僞幣，這種情形的秤量次數爲 $W\left(\frac{n}{3},2\right)+3$ 。

（甲.2）再來我們對表六來做分析，一樣會根據 B 是奇數或是偶數來做討論。

（甲.2.1）B 爲奇數時：我們從 B 拿出一個 b’ ，然後 B 分對半秤量。

（甲.2.1.1）假如天秤是不平的，就表示僞幣是較正常硬幣重的，且僞幣的分布情形爲（101），此時只要再針對 M 跟 B 各做一次一枚僞幣已知輕重的演算法，就可以找出僞幣，這種情形的秤量次數爲 $2W\left(\frac{n}{3},1\right)+3$ 。

（甲.2.1.2）假如天秤是平的：我們再從 B 拿出 b”，然後把 b’ 放回 B，B 分對半秤量。

（甲.2.1.2.1）假如天秤是平的，就表示僞幣是較正常硬幣輕的，且僞幣的分布情形爲（020），此時只要再對 A 做一次二枚僞幣已知輕重的演算法，就能夠找出僞幣，這種情形的秤量次數爲 $W\left(\frac{n}{3},2\right)+4$ 。

（甲.2.1.2.2）假如天秤是不平的，就表示僞幣是較正常硬幣重的，且 b’ 爲僞幣，僞幣的分布情形變爲（100），此時只要再對 M 做一次一枚僞幣已知輕重的演算法，就能夠找出僞幣，這種情形的秤量次數爲 $W\left(\frac{n}{3},1\right)+4$ 。

（甲.2.2）B 爲偶數時：我們就可以直接將 B 分對半秤量。

（甲.2.2.1）假如天秤是不平的，我們就可以知道僞幣是較正常硬幣重的，且僞幣的分布爲（101），此時只要再對 M 跟 B 各做一次一枚僞幣已知輕重的演算法，就能夠找出僞幣，這種情形的秤量次數爲 $2W\left(\frac{n}{3},1\right)+3$ 。

（甲.2.2.2）假如天秤是平的，我們就可以知道僞幣是較正常硬幣輕的，且僞幣的分布爲（020），此時只要再對 A 做一次二枚僞幣已知輕重的演算法，就能夠找出僞幣，這種情形的秤量次數爲 $W\left(\frac{n}{3},2\right)+3$ 。

表七：分支乙 M>B 時僞幣的可能分布情形

	M	A	B	m’
+	2	0	0	
+	1	0	0	1
-	0	1	1	

表八：分支乙 M=B 時僞幣的可能分布情形

	M	A	B	m’
+	1	0	1	
-	0	2	0	

表九：分支乙 M<B 時僞幣的可能分布情形

	M	A	B	m’
+	0	0	1	1

分支乙：M>A, |M|=|B|+1

由於M比B個數多了一個，所以我們就先從M拿出一個硬幣m’，再秤量M跟B，所得到的結果會是表七、八、及九。

表格最右方欄m’等於1時，是表示我們拿出的m’剛好是僞幣時的情形。

（乙.1）我們先對表七做分析，在此我們會針對B是奇數或是偶數做不同的討論。

（乙.1.1）B爲奇數時：A就爲偶數，所以我們將A對半秤量

（乙.1.1.1）假如天秤是不平的，那表示僞幣是較正常硬幣輕的，且僞幣的分布爲（011）。所以接下來我們只要對A跟B各做一次一枚僞幣已知輕重的演算法就可以找出僞幣了，這種情形的秤量次數爲$W\left(\left\lceil \frac{n}{3} \right\rceil,1\right)+W\left(\left\lfloor \frac{n}{3} \right\rfloor,1\right)+3$。

（乙.1.1.2）假如天秤是平的，那表示僞幣是較正常硬幣重的，我們把m’放回M，僞幣的分布就變爲（200）。此時只要再對M做一次二枚僞幣已知輕重的演算法，就能夠找出僞幣，這種情形的秤量次數爲$W\left(\left\lceil \frac{n}{3} \right\rceil,2\right)+3$。

（乙.1.2）B爲偶數時：我們就可以直接將B分對半秤量。

（乙.1.2.1）假如天秤是不平的，我們就可以知道僞幣是較正常硬幣輕的，且僞幣的分布爲（011），此時只要再對A跟B各做一次一枚僞幣已知輕重的演算法，就能夠找出僞幣，這種情形的秤量次數爲$W\left(\left\lceil \frac{n}{3} \right\rceil,1\right)+W\left(\left\lfloor \frac{n}{3} \right\rfloor,1\right)+3$。

（乙.1.2.2）假如天秤是平的，我們就可以知道僞幣是較正常硬幣重的，我們把m’放回M，僞幣的分布就成爲（200），此時只要再對M做一次二枚僞幣已知輕重的演算法，就能夠找出僞幣，這種情形的秤量次數爲$W\left(\left\lceil \frac{n}{3} \right\rceil,2\right)+3$。

（乙.2）接下來我們對表八來做分析，一樣會根據B是奇數或是偶數來做討論。

（乙.2.1）B爲奇數時：我們把m’放到B，所以此時B就變成偶數了，然後B分對半秤量。

（乙.2.1.1）假如天秤是不平的，就表示僞幣是較正常硬幣重的，且僞幣的分布情形爲（101），此時只要再針對M跟B各做一次一枚僞幣已知輕重的演算法，就可以找出僞幣，這種情形的秤量次數爲$W\left(\left\lceil \frac{n}{3} \right\rceil,1\right)+W\left(\left\lfloor \frac{n}{3} \right\rfloor,1\right)+3$。

（乙.2.1.2）假如天秤是平的，就表示僞幣是較正常硬幣輕的，且僞幣的分布情形爲（020），此時只要再針對A做一次二枚僞幣已知輕重的演算法，就可以找出僞幣，這種情形的秤量次數爲$W\left(\left\lceil \frac{n}{3} \right\rceil,2\right)+3$。

（乙.2.2）B爲偶數時：我們就可以直接將B分對半秤量。

（乙.2.2.1）假如天秤是不平的，我們就可以知道僞幣是較正常硬幣重的，且僞幣的分布爲（101），此時只要再對M跟B各做一次一枚僞幣已知輕重的演算法，就能夠找

出偽幣，這種情形的秤量次數爲$W\left(\left\lceil \frac{n}{3} \right\rceil,1\right)+W\left(\left\lfloor \frac{n}{3} \right\rfloor,1\right)+3$。

（乙.2.2.2）假如天秤是平的，我們就可以知道偽幣是較正常硬幣輕的，且偽幣的分布爲（020），此時只要再對 A 做一次二枚偽幣已知輕重的演算法，就能夠找出偽幣，這種情形的秤量次數爲$W\left(\left\lceil \frac{n}{3} \right\rceil,2\right)+3$。

（乙.3）在表九的部分，由於只剩一種可能性，我們就不必繼續秤量下去了，在這種情形下，我們可以知道偽幣是較正常硬幣要重的，且 m’ 也爲偽幣，偽幣的分布情形是（001），所以我們只要再對 B 做一次一枚偽幣已知輕重的演算法，就可以找出偽幣，這種情形的秤量次數爲$W\left(\left\lfloor \frac{n}{3} \right\rfloor,1\right)+2$。

分支丙：M>A, |M|=|B|-1

由於 B 比 M 個數多了一個，所以我們就先從 B 拿出一個硬幣 b’，再秤量 M 跟 B，所得到的結果會是表十及十一。

（丙.1）我們先對表十做分析，在此我們會針對 B 是奇數或是偶數做不同的討論。

（丙.1.1）B 爲奇數時：A 就爲偶數，我們把 A 對半秤量。

（丙.1.1.1）假如天秤是不平的，那表示偽幣是較正常硬幣輕的，且偽幣的分布爲（011）。所以接下來我們只要對 A 跟 B 各做一次一枚偽幣已知輕重的演算法就可以找出偽幣了，這種情形的秤量次數爲$W\left(\left\lfloor \frac{n}{3} \right\rfloor,1\right)+W\left(\left\lceil \frac{n}{3} \right\rceil,1\right)+3$。

（丙.1.1.2）假如天秤是平的，我們就可以知道偽幣是較正常硬幣重的，我們把 b’ 放到 M，偽幣的分布就成爲（200），此時只要再對 M 做一次二枚偽幣已知輕重的演算法，就能夠找出偽幣，這種情形的秤量次數爲$W\left(\left\lfloor \frac{n}{3} \right\rfloor,2\right)+3$。

表十：分支丙 M>B 時偽幣的可能分布情形

	M	A	B	b’
+	2	0	0	
+	1	0	0	1
-	0	1	1	

表十一：分支丙 M=B 時偽幣的可能分布情形

	M	A	B	b’
+	1	0	1	
-	0	2	0	
-	0	1	0	1

（丙.1.2）B 爲偶數時：我們就可以直接將 B 分對半秤量。

（丙.1.2.1）假如天秤是不平的，我們就可以知道僞幣是較正常硬幣輕的，且僞幣的分布爲（011），此時只要再對 A 跟 B 各做一次一枚僞幣已知輕重的演算法，就能夠找出僞幣，這種情形的秤量次數爲$W\left(\left\lfloor\frac{n}{3}\right\rfloor,1\right)+W\left(\left\lceil\frac{n}{3}\right\rceil,1\right)+3$。

（丙.1.2.2）假如天秤是平的，我們就可以知道僞幣是較正常硬幣重的，我們把 b’ 放到 M，僞幣的分布就成爲（200），此時只要再對 M 做一次二枚僞幣已知輕重的演算法，就能夠找出僞幣，這種情形的秤量次數爲$W\left(\left\lfloor\frac{n}{3}\right\rfloor,2\right)+3$。

（丙.2）接下來我們對表十一來做分析，一樣會根據 B 是奇數或是偶數來做討論。

（丙.2.1）B 爲奇數時：我們從 B 拿出一個 b”，然後 B 分對半秤量

（丙.2.1.1）假如天秤是不平的，就表示僞幣是較正常硬幣重的，且僞幣的分布情形爲（101），此時只要再針對 M 跟 B 各做一次一枚僞幣已知輕重的演算法，就可以找出僞幣，這種情形的秤量次數爲$W\left(\left\lfloor\frac{n}{3}\right\rfloor,1\right)+W\left(\left\lceil\frac{n}{3}\right\rceil,1\right)+3$。

（丙.2.1.2）假如天秤是平的：我們再從 B 拿出 b”’，然後把 b” 放回 B，B 分對半秤量。

（丙.2.1.1.1）假如天秤是不平的，那表示僞幣是較正常硬幣重的，且 b” 爲僞幣，僞幣的分布變爲（100），所以接下來我們只要對 M 做一次一枚僞幣已知輕重的演算法就可以找出僞幣，這種情形的秤量次數爲$W\left(\left\lfloor\frac{n}{3}\right\rfloor,1\right)+4$。

（丙.2.1.1.2）假如天秤是平的，那表示僞幣是較正常硬幣輕的，我們把 b’ 放到 A，僞幣的分布就成爲（020），之後只要對 A 做一次二枚僞幣已知輕重的演算法就能找出僞幣了，這種情形的秤量次數爲$W\left(\left\lfloor\frac{n}{3}\right\rfloor,2\right)+4$。

（丙.2.2）B 爲偶數時：我們就可以直接將 B 分對半秤量。

（丙.2.2.1）假如天秤是不平的，我們就可以知道僞幣是較正常硬幣重的，且僞幣的分布爲（101），此時只要再對 M 跟 B 各做一次一枚僞幣已知輕重的演算法，就能夠找出僞幣，這種情形的秤量次數爲$W\left(\left\lfloor\frac{n}{3}\right\rfloor,1\right)+W\left(\left\lceil\frac{n}{3}\right\rceil,1\right)+3$。

（丙.2.2.2）假如天秤是平的，我們就可以知道僞幣是較正常硬幣輕的，我們把 b’ 放到 A，僞幣的分布就成爲（020），此時只要再對 A 做一次二枚僞幣已知輕重的演算法，就能夠找出僞幣，這種情形的秤量次數爲$W\left(\left\lfloor\frac{n}{3}\right\rfloor,2\right)+3$。

分支丁：M=A, |M|=|B|

由於 M 的個數跟 B 的個數一樣，所以可以直接秤量 M 跟 B，秤量的結果會得到表十二及十三。

表十二：分支丁 M>B 時僞幣的可能分布情形

	M	A	B
+	1	1	0
-	0	0	2

表十三：分支丁 M<B 時僞幣的可能分布情形

	M	A	B
+	0	0	2
-	1	1	0

（丁.1）我們先對表十二做分析，在此我們會針對 M 是奇數或是偶數做不同的討論。

（丁.1.1）M 爲奇數時：我們從 M 拿出一個 m’ ，然後 M 分對半秤量。

（丁.1.1.1）假如天秤是不平的，那表示僞幣是較正常硬幣重的，且僞幣的分布爲（110）。所以接下來我們只要對 M 跟 A 做兩次一枚僞幣已知輕重的演算法就可以找出僞幣了，這種情形的秤量次數爲 $2W\left(\frac{n}{3},1\right)+3$ 。

（丁.1.1.2）假如天秤是平的：我們再從 M 拿出 m” ，然後把 m’ 放回 M，M 分對半秤量。

（丁.1.1.2.1）假如天秤是不平的，那表示僞幣是較正常硬幣重的，且 m’ 爲僞幣，僞幣的分布變爲（010），所以接下來我們只要對 A 做一次一枚僞幣已知輕重的演算法就可以找出僞幣，這種情形的秤量次數爲 $W\left(\frac{n}{3},1\right)+4$ 。

（丁.1.1.2.2）假如天秤是平的，那表示僞幣是較正常硬幣輕的，僞幣的分布爲（002），之後只要對 B 做一次二枚僞幣已知輕重的演算法就能找出僞幣了，這種情形的秤量次數爲 $W\left(\frac{n}{3},2\right)+4$ 。

（丁.1.2）M 爲偶數時：我們就可以直接將 M 分對半秤量。

（丁.1.2.1）假如天秤是不平的，我們就可以知道僞幣是較正常硬幣重的，且僞幣的分布爲（110），此時只要再對 M 跟 A 各做一次一枚僞幣已知輕重的演算法，就能夠找出僞幣，這種情形的秤量次數爲 $2W\left(\frac{n}{3},1\right)+3$ 。

（丁.1..2.2）假如天秤是平的，我們就可以知道僞幣是較正常硬幣輕的，僞幣的分布爲（002），此時只要再對 B 做一次二枚僞幣已知輕重的演算法，就能夠找出僞幣，這種情形的秤量次數爲 $W\left(\frac{n}{3},2\right)+3$ 。

（丁.2）接下來我們對表十三來做分析，一樣會根據 M 是奇數或是偶數來做討論。

（丁.2.1）M 爲奇數時：我們從 M 拿出一個 m’ ，然後 M 分對半秤量。

（丁.2.1.1）假如天秤是不平的，那表示僞幣是較正常硬幣輕的，且僞幣的分布爲（110）。所以接下來我們只要對 M 跟 A 做兩次一枚僞幣已知輕重的演算法就可以找出僞幣了，這種情形的秤量次數爲 $2W\left(\frac{n}{3},1\right)+3$ 。

（丁.2.1.2）假如天秤是平的：我們再從 M 拿出 m” ，然後把 m’ 放回 M，M 分對半秤量。

（丁.2.1.2.1）假如天秤是不平的，那表示僞幣是較正常硬幣輕的，且 m’ 爲僞幣，僞幣的分布變爲（010），所以接下來我們只要對 A 做一次一枚僞幣已知輕重的演算法就可以找出僞幣，這種情形的秤量次數爲 $W\left(\frac{n}{3},1\right)+4$ 。

（丁.2.1.2.2）假如天秤是平的，那表示僞幣是較正常硬幣重的，僞幣的分布爲（002），之後只要對 B 做一次二枚僞幣已知輕重的演算法就能找出僞幣了，這種情形的秤量次數爲 $W\left(\frac{n}{3},2\right)+4$ 。

（丁.2.2）M 爲偶數時：我們就可以直接將 M 分對半秤量。

（丁.2.2.1）假如天秤是不平的，我們就可以知道僞幣是較正常硬幣輕的，且僞幣的分布爲（110），此時只要再對 M 跟 A 各做一次一枚僞幣已知輕重的演算法，就能夠找出僞幣，這種情形的秤量次數爲 $2W\left(\frac{n}{3},1\right)+3$ 。

（丁.2.2.2）假如天秤是平的，我們就可以知道僞幣是較正常硬幣重的，僞幣的分布爲（002），此時只要再對 B 做一次二枚僞幣已知輕重的演算法，就能夠找出僞幣，這種情形的秤量次數爲 $W\left(\frac{n}{3},2\right)+3$ 。

分支戊：M=A, |M|=|B|+1

由於 M 比 B 個數多了一個，所以我們就先從 M 拿出一個硬幣 m’ ，再秤量 M 跟 B，所得到的結果會是表十四、十五、十六。

（戊.1）我們先對表十四做分析，在此我們會針對 M 是奇數或是偶數做不同的討論。

（戊.1.1）M 爲奇數時：我們把 m’ 放回 M，所以此時 M 就變成偶數了，然後 M 分對半秤量。

（戊.1.1.1）假如天秤是不平的，就表示僞幣是較正常硬幣重的，且僞幣的分布情形爲（110），此時只要再針對 M 跟 A 各做一次一枚僞幣已知輕重的演算法，就可以找出僞幣，這種情形的秤量次數爲 $2W\left(\left\lceil\frac{n}{3}\right\rceil,1\right)+3$ 。

表十四：分支戊 M>B 時僞幣的可能分布情形

	M	A	B	m’
+	1	1	0	
-	0	0	2	

表十五：分支戊 M=B 時僞幣的可能分布情形

	M	A	B	m’
+	0	1	0	1
-	0	1	0	1

表十六：分支戊 M<B 時僞幣的可能分布情形

	M	A	B	m’
+	0	0	2	
-	1	1	0	

（戊.1.1.2）假如天秤是平的，就表示僞幣是較正常硬幣輕的，且僞幣的分布情形爲（002），此時只要再針對 B 做一次二枚僞幣已知輕重的演算法，就可以找出僞幣，這種情形的秤量次數爲 $W\left(\left\lfloor \frac{n}{3} \right\rfloor,2\right)+3$ 。

（戊.1.2）M 爲偶數時：我們就可以直接將 M 分對半秤量。

（戊.1.2.1）假如天秤是不平的，我們就可以知道僞幣是較正常硬幣重的，且僞幣的分布爲（110），此時只要再對 M 跟 A 各做一次一枚僞幣已知輕重的演算法，就能夠找出僞幣，這種情形的秤量次數爲 $2W\left(\left\lceil \frac{n}{3} \right\rceil,1\right)+3$ 。

（戊.1.2.2）假如天秤是平的，我們就可以知道僞幣是較正常硬幣輕的，且僞幣的分布爲（002），此時只要再對 B 做一次二枚僞幣已知輕重的演算法，就能夠找出僞幣，這種情形的秤量次數爲 $W\left(\left\lfloor \frac{n}{3} \right\rfloor,2\right)+3$ 。

（戊.2）接下來我們對表十六來做分析，一樣會根據 M 是奇數或是偶數來做討論。

（戊.2.1）M 爲奇數時：我們把 m’ 放回 M，所以此時 M 就變成偶數了，然後 M 分對半秤量。

（戊.2.1.1）假如天秤是不平的，就表示僞幣是較正常硬幣輕的，且僞幣的分布情形爲（110），此時只要再針對 M 跟 A 各做一次一枚僞幣已知輕重的演算法，就可以找出僞幣，這種情形的秤量次數爲 $2W\left(\left\lceil \frac{n}{3} \right\rceil,1\right)+3$ 。

（戊.2.1.2）假如天秤是平的，就表示偽幣是較正常硬幣重的，且偽幣的分布情形爲（002），此時只要再針對 B 做一次二枚偽幣已知輕重的演算法，就可以找出偽幣，這種情形的秤量次數爲 $W\left(\left\lfloor \frac{n}{3} \right\rfloor, 2\right)+3$ 。

（戊.2.2）M 爲偶數時：我們就可以直接將 M 分對半秤量。

（戊.2.2.1）假如天秤是不平的，我們就可以知道偽幣是較正常硬幣輕的，且偽幣的分布爲（110），此時只要再對 M 跟 A 各做一次一枚偽幣已知輕重的演算法，就能夠找出偽幣，這種情形的秤量次數爲 $2W\left(\left\lceil \frac{n}{3} \right\rceil, 1\right)+3$ 。

（戊.2.2.2）假如天秤是平的，我們就可以知道偽幣是較正常硬幣重的，且偽幣的分布爲（002），此時只要再對 B 做一次二枚偽幣已知輕重的演算法，就能夠找出偽幣，這種情形的秤量次數爲 $W\left(\left\lfloor \frac{n}{3} \right\rfloor, 2\right)+3$ 。

（戊.3）最後我們對表十五來做分析，我們從 B 拿出 b’ ，我們可以確定 b’ 是正常硬幣，所以我們就秤量 b’ 跟 m’ ：

（戊.3.1）假如 m’ 爲較輕偽幣的話，偽幣分布爲（010），接下來只要對 A 做一次一枚偽幣已知輕重的演算法就能找到偽幣，這種情形的秤量次數爲 $W\left(\left\lceil \frac{n}{3} \right\rceil, 1\right)+3$ 。

（戊.3.2）假如 m’ 爲較重偽幣的話，偽幣分布爲（010），接下來只要對 A 做一次一枚偽幣已知輕重的演算法就能找到偽幣，這種情形的秤量次數爲 $W\left(\left\lceil \frac{n}{3} \right\rceil, 1\right)+3$ 。

分支己：M=A, |M|=|B|-1

由於 B 比 M 個數多了一個，所以我們就先從 B 拿出一個硬幣 b’ ，再秤量 M 跟 B，所得到的結果會是表十七及十八。

表十七：分支己 M>B 時偽幣的可能分布情形

	M	A	B	b’
+	1	1	0	
-	0	0	2	
-	0	0	1	1

表十八：分支己 M<B 時偽幣的可能分布情形

	M	A	B	b’
+	0	0	2	
+	0	0	1	1
-	1	1	0	

（己.1）我們先對表十七做分析，在此我們會針對 M 是奇數或是偶數做不同的討論。

（己.1.1）M 爲奇數時：我們從 M 拿出一個 m’ ，然後 M 分對半秤量。

（己.1.1.1）假如天秤是不平的，那表示僞幣是較正常硬幣重的，且僞幣的分布爲（110）。所以接下來我們只要對 M 跟 A 做兩次一枚僞幣已知輕重的演算法就可以找出僞幣了，這種情形的秤量次數爲 $2W\left(\left\lfloor \frac{n}{3} \right\rfloor,1\right)+3$ 。

（己.1.1.2）假如天秤是平的：我們再從 M 拿出 m” ，然後把 m’ 放回 M，M 分對半秤量。

（己.1.1.2.1）假如天秤是不平的，那表示僞幣是較正常硬幣重的，且 m’ 爲僞幣，僞幣的分布變爲（010），所以接下來我們只要對 A 做一次一枚僞幣已知輕重的演算法就可以找出僞幣，這種情形的秤量次數爲 $W\left(\left\lfloor \frac{n}{3} \right\rfloor,1\right)+4$ 。

（己.1.1.2.2）假如天秤是平的，那表示僞幣是較正常硬幣輕的，僞幣的分布爲（002），之後只要對 B 做一次二枚僞幣已知輕重的演算法就能找出僞幣了，這種情形的秤量次數爲 $W\left(\left\lceil \frac{n}{3} \right\rceil,2\right)+4$ 。

（己.1.2）M 爲偶數時：我們就可以直接將 M 分對半秤量。

（己.1.2.1）假如天秤是不平的，我們就可以知道僞幣是較正常硬幣重的，且僞幣的分布爲（110），此時只要再對 M 跟 A 各做一次一枚僞幣已知輕重的演算法，就能夠找出僞幣，這種情形的秤量次數爲 $2W\left(\left\lfloor \frac{n}{3} \right\rfloor,1\right)+3$ 。

（己.1.2.2）假如天秤是平的，我們就可以知道僞幣是較正常硬幣輕的，我們把 b’ 放回 B，僞幣的分布就成爲（002），此時只要再對 B 做一次二枚僞幣已知輕重的演算法，就能夠找出僞幣，這種情形的秤量次數爲 $W\left(\left\lceil \frac{n}{3} \right\rceil,2\right)+3$ 。

（己.2）接下來我們對表十八來做分析，一樣會根據 M 是奇數或是偶數來做討論。

（己.2.1）M 爲奇數時：我們從 M 拿出一個 m’ ，然後 M 分對半秤量。

（己.2.1.1）假如天秤是不平的，那表示僞幣是較正常硬幣輕的，且僞幣的分布爲（110）。所以接下來我們只要對 M 跟 A 做兩次一枚僞幣已知輕重的演算法就可以找出僞幣了，這種情形的秤量次數爲 $2W\left(\left\lfloor \frac{n}{3} \right\rfloor,1\right)+3$ 。

（己.2.1.2）假如天秤是平的：我們再從 M 拿出 m” ，然後把 m’ 放回 M，M 分對半秤量。

（己.2.1.2.1）假如天秤是不平的，那表示僞幣是較正常硬幣輕的，且 m’ 爲僞幣，僞幣的分布變爲（010），所以接下來我們只要對 A 做一次一枚僞幣已知輕重的

演算法就可以找出僞幣，這種情形的秤量次數爲$W\left(\left\lfloor \frac{n}{3} \right\rfloor,1\right)+4$。

（己.2.1.2.2）假如天秤是平的，那表示僞幣是較正常硬幣重的，我們把b’放回B，僞幣的分布就成爲（002），之後只要對B做一次二枚僞幣已知輕重的演算法就能找出僞幣了，這種情形的秤量次數爲$W\left(\left\lceil \frac{n}{3} \right\rceil,2\right)+4$。

（己.2.2）M爲偶數時：我們就可以直接將M分對半秤量。

（己.2.2.1）假如天秤是不平的，我們就可以知道僞幣是較正常硬幣輕的，且僞幣的分布爲（110），此時只要再對M跟A各做一次一枚僞幣已知輕重的演算法，就能夠找出僞幣，這種情形的秤量次數爲$2W\left(\left\lfloor \frac{n}{3} \right\rfloor,1\right)+3$。

（己.2.2.2）假如天秤是平的，我們就可以知道僞幣是較正常硬幣重的，我們把b’放回B，僞幣的分布就成爲（002），此時只要再對B做一次二枚僞幣已知輕重的演算法，就能夠找出僞幣，這種情形的秤量次數爲$W\left(\left\lceil \frac{n}{3} \right\rceil,2\right)+3$。

綜合上面所有六個分支的秤量次數，我們可以推算出此演算法的最大秤量次數如下（詳見表十九）：

$$S(n,2)=\begin{cases}2\lceil \log_3 n \rceil+2 & ,當 n=3p\\ 2\lceil \log_3 (n+2) \rceil+2 & ,當 n=3p+1\\ 2\lceil \log_3 (n+1) \rceil+1 & ,當 n=3p+2\end{cases}$$

其中n≥5

從上面的結果可以看出，我們的演算法在最大秤量次數上，會比李立中的「交錯秤量法」[10]還要再少一次秤量次數。下面圖二我們以一簡易的樹狀架構來表示這整個演算法的虛擬碼。

在此由於僞幣問題在假設僞幣個數時必須較正常硬幣要少，所以當僞幣數爲兩枚時，正常硬幣數最少要爲3枚（2+1），總共硬幣數最少爲5枚（2+3）。所以當僞幣數爲L枚時，我們的總硬幣個數至少要爲2L+1枚。

```
第一次秤量 (M 跟 A 秤)
- M > A
    第二次秤量 (M 跟 B 秤)
    (甲) |M| = |B|
        - M > B
            - |B|爲奇數，則第三次秤量, 011(輕)；第四次秤量, 200(重), 010(輕)
            - |B|爲偶數，則第三次秤量, 011(輕), 200(重)
        - M = B
            - |B|爲奇數，則第三次秤量, 101(重)；第四次秤量, 020(輕), 100(重)
            - |B|爲偶數，則第三次秤量, 101(重), 020(輕)
```

（乙）|M|=|B|+1

- M>B
  - |B|爲奇數，則第三次秤量, 011(輕), 200(重)
  - |B|爲偶數，則第三次秤量, 011(輕), 200(重)
- M=B
  - |B|爲奇數，則第三次秤量, 101(重), 020(輕)
  - |B|爲偶數，則第三次秤量, 101(重), 020(輕)
- M<B
  - 只有一組候選人, 不必再秤了, 001(重)

（丙）*|M|=|B|-1*

- M>B
  - |B|爲奇數，則第三次秤量, 011(輕), 200(重)
  - |B|爲偶數，則第三次秤量, 011(輕), 200(重)
- M=B
  - |B|爲奇數，則第三次秤量, 101(重)；第四次秤量, 020(輕), 100(重)
  - |B|爲偶數，則第三次秤量, 101(重), 020(輕)

- M=A

第二次秤量 (M 跟 B 秤)

（丁）|M|=|B|

- M>B
  - |M|爲奇數，則第三次秤量, 110(重)；第四次秤量,002(輕), 010(重)
  - |M|爲偶數，則第三次秤量, 110(重), 002(輕)
- M<B
  - |M|爲奇數，則第三次秤量, 110(輕)；第四次秤量,002(重), 010(輕)
  - |M|爲偶數，則第三次秤量, 110(輕), 002(重)

（戊）|M|=|B|+1

- M>B
  - |M|爲奇數，則第三次秤量, 110(重), 002(輕)
  - |M|爲偶數，則第三次秤量, 110(重), 002(輕)
- M<B
  - |M|爲奇數，則第三次秤量, 110(輕), 002(重)
  - |M|爲偶數，則第三次秤量, 110(輕), 002(重)
- M=B

  第三次秤量, 010(輕), 010(重)

（己）|M|=|B|-1

- M>B
  - |M|爲奇數，則第三次秤量, 110(重)；第四次秤量,002(輕), 010(重)
  - |M|爲偶數，則第三次秤量, 110(重), 002(輕)
- M<B
  - |M|爲奇數，則第三次秤量, 110(輕)；第四次秤量,002(重), 010(輕)
  - |M|爲偶數，則第三次秤量, 110(輕), 002(重)

圖二：整個演算法的虛擬碼

表十九：本演算法的最大秤量次數

分支		秤量次數	最大秤量次數
甲 n=3p	（甲.1.1.1）	$2W\left(\frac{n}{3},1\right)+3=2\lceil\log_3 n\rceil+1$	$2\lceil\log_3 n\rceil+2$
	（甲.1.1.2.1）	$W\left(\frac{n}{3},1\right)+4=\lceil\log_3 n\rceil+3$	
	（甲.1.1.2.2）	$W\left(\frac{n}{3},2\right)+4=2\lceil\log_3 n\rceil+2$	
	（甲.1.2.1）	$2W\left(\frac{n}{3},1\right)+3=2\lceil\log_3 n\rceil+1$	
	（甲.1.2.2）	$W\left(\frac{n}{3},2\right)+3=2\lceil\log_3 n\rceil+1$	
	（甲.2.1.1）	$2W\left(\frac{n}{3},1\right)+3=2\lceil\log_3 n\rceil+1$	
	（甲.2.1.2.1）	$W\left(\frac{n}{3},2\right)+4=2\lceil\log_3 n\rceil+2$	
	（甲.2.1.2.2）	$W\left(\frac{n}{3},1\right)+4=\lceil\log_3 n\rceil+3$	
	（甲.2.2.1）	$2W\left(\frac{n}{3},1\right)+3=2\lceil\log_3 n\rceil+1$	
	（甲.2.2.2）	$W\left(\frac{n}{3},2\right)+3=2\lceil\log_3 n\rceil+1$	
乙 n=3p+2	（乙.1.1.1）	$W\left(\left\lceil\frac{n}{3}\right\rceil,1\right)+W\left(\left\lfloor\frac{n}{3}\right\rfloor,1\right)+3$ $=\lceil\log_3(n+1)\rceil+\lceil\log_3(n-2)\rceil+1$	$2\lceil\log_3(n+1)\rceil+1$
	（乙.1.1.2）	$W\left(\left\lceil\frac{n}{3}\right\rceil,2\right)+3=2\lceil\log_3(n+1)\rceil+1$	
	（乙.1.2.1）	$W\left(\left\lceil\frac{n}{3}\right\rceil,1\right)+W\left(\left\lfloor\frac{n}{3}\right\rfloor,1\right)+3$ $=\lceil\log_3(n+1)\rceil+\lceil\log_3(n-2)\rceil+1$	
	（乙.1.2.2）	$W\left(\left\lceil\frac{n}{3}\right\rceil,2\right)+3=2\lceil\log_3(n+1)\rceil+1$	
	（乙.2.1.1）	$W\left(\left\lceil\frac{n}{3}\right\rceil,1\right)+W\left(\left\lfloor\frac{n}{3}\right\rfloor,1\right)+3$ $=\lceil\log_3(n+1)\rceil+\lceil\log_3(n-2)\rceil+1$	

	（乙.2.1.2）	$W\left(\left\lceil \frac{n}{3} \right\rceil,2\right)+3=2\lceil \log_3(n+1)\rceil+1$	
	（乙.2.2.1）	$W\left(\left\lceil \frac{n}{3} \right\rceil,1\right)+W\left(\left\lfloor \frac{n}{3} \right\rfloor,1\right)+3$ $=\lceil \log_3(n+1)\rceil+\lceil \log_3(n-2)\rceil+1$	
	（乙.2.2.2）	$W\left(\left\lceil \frac{n}{3} \right\rceil,2\right)+3=2\lceil \log_3(n+1)\rceil+1$	
	（乙.3）	$W\left(\left\lfloor \frac{n}{3} \right\rfloor,1\right)+2=\lceil \log_3(n-2)\rceil+1$	
丙 n=3p+1	（丙.1.1.1）	$W\left(\left\lfloor \frac{n}{3} \right\rfloor,1\right)+W\left(\left\lceil \frac{n}{3} \right\rceil,1\right)+3$ $=\lceil \log_3(n-1)\rceil+\lceil \log_3(n+2)\rceil+1$	$2\lceil \log_3(n-1)\rceil+2$
	（丙.1.1.2）	$W\left(\left\lfloor \frac{n}{3} \right\rfloor,2\right)+3=2\lceil \log_3(n-1)\rceil+1$	
	（丙.1.2.1）	$W\left(\left\lfloor \frac{n}{3} \right\rfloor,1\right)+W\left(\left\lceil \frac{n}{3} \right\rceil,1\right)+3$ $=\lceil \log_3(n-1)\rceil+\lceil \log_3(n+2)\rceil+1$	
	（丙.1.2.2）	$W\left(\left\lfloor \frac{n}{3} \right\rfloor,2\right)+3=2\lceil \log_3(n-1)\rceil+1$	
	（丙.2.1.1）	$W\left(\left\lfloor \frac{n}{3} \right\rfloor,1\right)+W\left(\left\lceil \frac{n}{3} \right\rceil,1\right)+3$ $=\lceil \log_3(n-1)\rceil+\lceil \log_3(n+2)\rceil+1$	
	（丙.2.1.1.1）	$W\left(\left\lfloor \frac{n}{3} \right\rfloor,1\right)+4=\lceil \log_3(n-1)\rceil+3$	
	（丙.2.1.1.2）	$W\left(\left\lfloor \frac{n}{3} \right\rfloor,2\right)+4=2\lceil \log_3(n-1)\rceil+2$	
	（丙.2.2.1）	$W\left(\left\lfloor \frac{n}{3} \right\rfloor,1\right)+W\left(\left\lceil \frac{n}{3} \right\rceil,1\right)+3$ $=\lceil \log_3(n-1)\rceil+\lceil \log_3(n+2)\rceil+1$	
	（丙.2.2.2）	$W\left(\left\lfloor \frac{n}{3} \right\rfloor,2\right)+3=2\lceil \log_3(n-1)\rceil+1$	
丁 n=3p	（丁.1.1.1）	$2W\left(\frac{n}{3},1\right)+3=2\lceil \log_3 n\rceil+1$	$2\lceil \log_3 n\rceil+2$
	（丁.1.1.2.1）	$W\left(\frac{n}{3},1\right)+4=\lceil \log_3 n\rceil+3$	

	（丁.1.1.2.2）	$W\left(\frac{n}{3},2\right)+4=2\lceil\log_3 n\rceil+2$	
	（丁.1.2.1）	$2W\left(\frac{n}{3},1\right)+3=2\lceil\log_3 n\rceil+1$	
	（丁.1..2.2）	$W\left(\frac{n}{3},2\right)+3=2\lceil\log_3 n\rceil+1$	
	（丁.2.1.1）	$2W\left(\frac{n}{3},1\right)+3=2\lceil\log_3 n\rceil+1$	
	（丁.2.1.2.1）	$W\left(\frac{n}{3},1\right)+4=\lceil\log_3 n\rceil+3$	
	（丁.2.1.2.2）	$W\left(\frac{n}{3},2\right)+4=2\lceil\log_3 n\rceil+2$	
	（丁.2.2.1）	$2W\left(\frac{n}{3},1\right)+3=2\lceil\log_3 n\rceil+1$	
	（丁.2.2.2）	$W\left(\frac{n}{3},2\right)+3=2\lceil\log_3 n\rceil+1$	
戊 n=3p+2	（戊.1.1.1）	$2W\left(\left\lceil\frac{n}{3}\right\rceil,1\right)+3=2\lceil\log_3(n+1)\rceil+1$	$2\lceil\log_3(n+1)\rceil+1$
	（戊.1.1.2）	$W\left(\left\lfloor\frac{n}{3}\right\rfloor,2\right)+3=2\lceil\log_3(n-2)\rceil+1$	
	（戊.1.2.1）	$2W\left(\left\lceil\frac{n}{3}\right\rceil,1\right)+3=2\lceil\log_3(n+1)\rceil+1$	
	（戊.1.2.2）	$W\left(\left\lfloor\frac{n}{3}\right\rfloor,2\right)+3=2\lceil\log_3(n-2)\rceil+1$	
	（戊.2.1.1）	$2W\left(\left\lceil\frac{n}{3}\right\rceil,1\right)+3=2\lceil\log_3(n+1)\rceil+1$	
	（戊.2.1.2）	$W\left(\left\lfloor\frac{n}{3}\right\rfloor,2\right)+3=2\lceil\log_3(n-2)\rceil+1$	
	（戊.2.2.1）	$2W\left(\left\lceil\frac{n}{3}\right\rceil,1\right)+3=2\lceil\log_3(n+1)\rceil+1$	
	（戊.2.2.2）	$W\left(\left\lfloor\frac{n}{3}\right\rfloor,2\right)+3=2\lceil\log_3(n-2)\rceil+1$	
	（戊.3.1）	$W\left(\left\lceil\frac{n}{3}\right\rceil,1\right)+3=\lceil\log_3(n+1)\rceil+1$	
	（戊.3.2）	$W\left(\left\lceil\frac{n}{3}\right\rceil,1\right)+3=\lceil\log_3(n+1)\rceil+1$	

己 n=3p+1	（己.1.1.1）	$2W\left(\left\lfloor\frac{n}{3}\right\rfloor,1\right)+3=2\lceil\log_3(n-1)\rceil+1$	$2\lceil\log_3(n+2)\rceil+2$
	（己.1.1.2.1）	$W\left(\left\lfloor\frac{n}{3}\right\rfloor,1\right)+4=\lceil\log_3(n-1)\rceil+3$	
	（己.1.1.2.2）	$W\left(\left\lceil\frac{n}{3}\right\rceil,2\right)+4=2\lceil\log_3(n+2)\rceil+2$	
	（己.1.2.1）	$2W\left(\left\lfloor\frac{n}{3}\right\rfloor,1\right)+3=2\lceil\log_3(n-1)\rceil+1$	
	（己.1.2.2）	$W\left(\left\lceil\frac{n}{3}\right\rceil,2\right)+3=2\lceil\log_3(n+2)\rceil+1$	
	（己.2.1.1）	$2W\left(\left\lfloor\frac{n}{3}\right\rfloor,1\right)+3=2\lceil\log_3(n-1)\rceil+1$	
	（己.2.1.2.1）	$W\left(\left\lfloor\frac{n}{3}\right\rfloor,1\right)+4=\lceil\log_3(n-1)\rceil+3$	
	（己.2.1.2.2）	$W\left(\left\lceil\frac{n}{3}\right\rceil,2\right)+4=2\lceil\log_3(n+2)\rceil+2$	
	（己.2.2.1）	$2W\left(\left\lfloor\frac{n}{3}\right\rfloor,1\right)+3=2\lceil\log_3(n-1)\rceil+1$	
	（己.2.2.2）	$W\left(\left\lceil\frac{n}{3}\right\rceil,2\right)+3=2\lceil\log_3(n+2)\rceil+1$	

當 n=3p 時

甲：$2\lceil\log_3 n\rceil+2$ ＝ 丁：$2\lceil\log_3 n\rceil+2$

當 n=3p+1 時

丙：$2\lceil\log_3(n-1)\rceil+2$ ＜ 己：$2\lceil\log_3(n+2)\rceil+2$

當 n=3p+2 時

乙：$2\lceil\log_3(n+1)\rceil+1$ ＝ 戊：$2\lceil\log_3(n+1)\rceil+1$

所以我們最後可以得知最大秤量次數如下：（n ≥ 5）

$$S(n,2)=\begin{cases}2\lceil\log_3 n\rceil+2 & ,當 n=3p\\ 2\lceil\log_3(n+2)\rceil+2 & ,當 n=3p+1\\ 2\lceil\log_3(n+1)\rceil+1 & ,當 n=3p+2\end{cases}$$

3.2 其它狀況的研究成果

在我們的研究中，我們對於兩枚僞幣未知輕重、三枚僞幣已知輕重、三枚僞幣未知輕重、四枚僞幣已知輕重以及四枚僞幣未知輕重，均提出了各種演算法來解決這些僞幣問題。由於篇幅限制，在此僅能大略列出結果如下。有興趣的讀者，可參閱[14]。

三枚僞幣已知輕重：

$$W(n,3)=\begin{cases}3\lceil \log_3 n\rceil & ,當 n=3p\\ 2\lceil \log_3(n+2)\rceil+\lceil \log_3(n-1)\rceil & ,當 n=3p+1 \quad ,n\ge 7\\ 2\lceil \log_3(n+1)\rceil+\lceil \log_3(n-2)\rceil & ,當 n=3p+2\end{cases}$$

三枚僞幣未知輕重：

$$S(n,3)=\begin{cases}3\lceil \log_3 n\rceil & ,當 n=3p\\ \max\{3\lceil \log_3(n-1)\rceil+1, 3\lceil \log_3(n+2)\rceil\} & ,當 n=3p+1 \quad ,n\ge 7\\ 3\lceil \log_3(n+1)\rceil+1 & ,當 n=3p+2\end{cases}$$

四枚僞幣已知輕重：

$$W(n,4)=\begin{cases}4\lceil \log_3 n\rceil & ,當 n=3p\\ 4\lceil \log_3(n+2)\rceil & ,當 n=3p+1 \quad ,n\ge 9\\ 4\lceil \log_3(n+1)\rceil & ,當 n=3p+2\end{cases}$$

四枚僞幣未知輕重：

$$S(n,4)=\begin{cases}4\lceil \log_3 n\rceil+2 & ,當 n=3p\\ 4\lceil \log_3(n-1)\rceil+2 & ,當 n=3p+1 \quad ,n\ge 9\\ 4\lceil \log_3(n+1)\rceil+2 & ,當 n=3p+2\end{cases}$$

在三枚與四枚僞幣的問題中，我們所採用的分析手法與兩枚僞幣的分析手法是很相似的，一開始都是以三分法爲基礎，把硬幣分成 M、A、B 三堆，再以如同兩枚僞幣的演算法架構圖把結果分類成甲、乙、丙、丁、戊、己來討論，顯而易見的當僞幣數目越多，我們所要分析的情況也越複雜，表二十至廿七就先列出三枚以及四枚僞幣經過演算法架構圖的第一次秤量（M 與 A 秤量）後的僞幣分布情形，分成 M>A 以及 M=A（從先前兩枚僞幣的討論中我們可以知道，M<A 的情形可以等同 M>A 來看待）：

表二十：三枚僞幣，已知僞幣輕重第一次秤量後 M>A 的僞幣分布情形

M	A	B
3	0	0
2	1	0
2	0	1
1	0	2

表廿一：三枚偽幣，已知偽幣輕重第一次秤量後 M=A 的偽幣分布情形

M	A	B
0	0	3
1	1	1

表廿二：三枚偽幣，未知偽幣輕重第一次秤量後 M>A 的偽幣分布情形

	M	A	B
	3	0	0
	2	1	0
	2	0	1
	1	0	2
-	1	2	0
-	0	3	0
-	0	2	1
-	0	1	2

表廿三：三枚偽幣，未知偽幣輕重第一次秤量後 M=A 的偽幣分布情形

	M	A	B
	1	1	1
	0	0	3
-	1	1	1
-	0	0	3

表廿四：四枚偽幣，已知偽幣輕重第一次秤量後 M>A 的偽幣分布情形

M	A	B
4	0	0
3	1	0
3	0	1
2	1	1
2	0	2
1	0	3

表廿五：四枚僞幣，已知僞幣輕重第一次秤量後 M=A 的僞幣分布情形

M	A	B
2	2	0
1	1	2
0	0	4

表廿六：四枚僞幣，未知僞幣輕重第一次秤量後 M>A 的僞幣分布情形

	M	A	B
+	4	0	0
+	3	1	0
+	3	0	1
+	2	1	1
+	2	0	2
+	1	0	3
−	1	3	0
−	1	2	1
−	0	4	0
−	0	3	1
−	0	2	2
−	0	1	3

表廿七：四枚僞幣，未知僞幣輕重第一次秤量後 M=A 的僞幣分布情形

‘	M	A	B
+	2	2	0
+	1	1	2
+	0	0	4
−	2	2	0
−	1	1	2
−	0	0	4

在此之後的演算法細節討論由於篇幅的關係，在此並不會去進一步討論，有興趣的讀者可以前往師範校院聯合博碩士論文系統（ http://140.122.127.247/cgi-bin/gs/gsweb.cgi?o=d1 ）來作進一步的查詢。

四、結論

自古以來偽幣問題就是人們不斷動腦筋思考的一個問題，從最單純的一枚偽幣，一直到多枚偽幣，知道偽幣輕重與不知偽幣輕重，還有像偽幣的重量可以不同等等的條件，使得問題本身變得更爲複雜，形成各式各樣的偽幣問題。

在我們的研究中，我們也對李立中的 Find 演算法[10]加以改良，使之成爲能夠解決「多枚偽幣未知輕重」偽幣問題的一般化演算法。而這個改良後的演算法，其實也可以用來解決「多枚偽幣未知輕重且偽幣個數未知」的偽幣問題，或許此方法並不能求出最佳解，但至少提供了此類問題一個參考的秤量次數上限（Upper bound）。

在未來的研究方向上，我們希望能夠從這些演算法中找出更多規律，看是否能以圖形分割的概念來解決偽幣問題，並且能夠更逼近最後的理論值。

誌謝

本研究部份由國科會專案補助，計畫編號爲 NSC-92-2213-E-003-006，特此誌謝。

參考文獻

[1] A. Born, C. A. J. Hurkens, G. J. Woeginger, "How to detect a counterfeit coin: adaptive versus non-adaptive solutions," *Information Processing Letters*, Vol. 86, pp. 137-141, 2003.

[2] A. D. Bonis, L. Gargano, U. Vaccaro, "Optimal detection of a counterfeit coin with multi-arms balances," *Discrete Applied Mathematics*, Vol. 61, pp. 121-131, 1995.

[3] J. A. Pyrich, *The counterfeit coin problem*, http://www.97cc.com/~japyrich/ projects/coins.shtml , 2001.

[4] L. Halbeisen, N. Hungerbuhler, "The general counterfeit coin problem," *Discrete Mathematics*, Vol.147, pp.139-150, 1995.

[5] P. J. Wan, D. Z. Du, "A $\left(\log_2 3+\frac{1}{2}\right)$-competitive algorithm for counterfeit coin problem," *Discrete Mathematics*, Vol.163, pp.173-200, 1997.

[6] P. J. Wan, Q. Yang, D. Kelley, "A $\left(\frac{3}{2}\log 3\right)$-competitive algorithm for the counterfeit coin problem," *Theoretical Computer Science*, Vol.181, pp.347-356, 1997.

[7] W. A. Liu, Z. K. Nie, "Optimal detection of two counterfeit coins with two-arms balance," *Discrete Applied Mathematics*, Vol.137, pp.267-291, 2004.

[8] W. Guzicki, "Ulam's searching game with two lies," *Journal of Combinatorial Theory*, Vol.54, pp.1-19, 1990.

[9] X. D. Hu, P. D. Chen, F. K. Hwang, "A new competitive algorithm for the counterfeit coin problem," *Information Processing Letters*, Vol.51, pp.213-218, 1994.

[10] 李立中，*多枚偽幣問題之演算法設計與分析*，國立台灣師範大學資訊教育學研究所碩士論文，2000。

[11] 陳善泰，*演繹競局及相關問題最佳化演算法之研究*，國立台灣師範大學資訊教育學研究所博士論文，2004。

[12] 楊錦潭、鄭又齊、馬德強、王授民合著，*演算法的天空*，松崗書局，1989。

[13] 馬德強、陳國正，*以歸納法處理偽幣問題*，http://wwwedu.nknu.edu.tw/ 48552001/，2002。

[14] 劉耀才，*偽幣問題之改良演算法設計與分析*，國立台灣師範大學資訊工程研究所碩士論文，2005。