# 數論輕鬆遊

沈淵源

數論, 特別是模算術 (modular arithmetic) [9] 在近代密碼學當中是非常重要的。這是大數學家戈德福瑞·哈洛德·哈地[1](Godfrey Harold Hardy) 在上一世紀的四十年代所無法想像的事情。當年他曾提到 [12]: 數論乃純數中的純數, 是遠離一般人類活動的一門學科。其實早在三十多年前, 數論就已經用在一般人類活動中, 如資訊傳輸裏的錯誤更正碼及密碼。我們在此介紹密碼學中用到的一些基本數論 [5, 14, 17], 至於更專門的課題如因數分解 [10] 與質數檢驗 [6]、離散對數 [7] 及橢圓曲線理論 [15, 20] 就等另外的時候再作介紹。不過首先要進場的是與我們爲伴的數學運算大師MATHEMATICA。其他的數學套裝軟體如 MATLAB 或是任何你所熟悉的系統都可拿來使用。

## 1. 數學運算大師簡介

數學運算大師MATHEMATICA是美國 Wolfram Research 公司所研發出來的一套由電腦來演算數學的系統。自從 1988 年發行上市以來, 由於其多才多藝, 早已建立起自己的形象而成爲衆多使用者所選擇的電腦代數系統 [8] (Computer Algebra System, 簡稱爲 CAS)。它提供了一強有力的數學程式環境包括數值的 (numerical)、符號的 (symbolical) 及圖形的 (graphical) 工具, 來協助我們解決數學方面的問題。已有相當多的人使用它來觀察並分析工程、數學、物理學、經濟學及其他科學領域上的問題。它也可當成高階的程式語言來使用。其工作的平台相當廣泛, 從 Cray 的超級電腦到桌上型及膝上型輕便電腦皆可。在將近兩百萬個使用者中其分佈大約如下 [22]: 工程25%, 物理學20%, 數學 18%, 電腦科學14%, 化學及化工6%, 財經4%, 生命科學4%, 社會科學3%, 其他6%。約有三分之二的使用者是在工業界及政府部門工作, 而僅僅8%爲學生使用者。

MATHEMATICA 是由兩大部分組合而成的, 就是所謂的前端 (Front End) 以及核心 (Kernel)。核心乃是其計算的引擎, 負責所有的計算工作。前端則透過記事本介面 (notebook

[1]哈地(1877-1947) 英國數學家。跟李特伍德 (J.E. Littlewood, 1885-1977) 在 1910 至 1945 年間, 兩人合作發表了將近 100 篇的論文: 包括數論、不等式及黎曼假設方面的著作。在黎曼假設方面證明了 Zeta 函數在直線 $x = \frac{1}{2}$ 上有無限多個零點。哈地曾邀請印度數學家雷曼紐卷 (Ramanujan, 1887-1920) 到英格蘭訪問, 兩人在 1914 至 1917 年間合作, 特別是在數的分割之研究最具原創性, 可參見數學傳播, 27卷3, 4期, 顏一清「探求『無限』奧秘的數學家—Srinivasa Ramanujan (上)、(下)」。

interface) 來作爲使用者與核心之間的溝通橋樑。當你啓動 MATHEMATICA 之後, 它會自動開啓一個工作視窗[2] 和一個常用的基本輸入面板 (Basic Input Palette)。我們稱此工作視窗爲記事本 (notebook), 而 MATHEMATICA 則用一種特殊的檔案格式來儲存工作視窗的內容, 其附加檔名爲 .nb。這些記事本有如一般的文字處理軟體一樣, 你可在上面加註解[3]、做結論還可匯入或匯出多種不同格式的圖形檔。要發表的論文資料或上課講義可在此預備, 也可在此下達指令將這些文件列印出來。其中的資料可在記事本之內或之間互相剪貼, 目的就是希望能再使用或是經過修改其文字、圖形或計算式子後成爲我們所需要的文件。

前端記事本檔案將裡面的資料分類安排放在所謂的「細胞 (cells)」當中, 所以這些細胞就是構成此記事本檔案最基本的單元。在這些文字、圖形或計算式的細胞之最右端都有「右中括弧」, 而這些「右中括弧」就代表整個文字、圖形或計算式單元。輸入細胞 (input cells) 含有 MATHEMATICA 的指令者可按下 `SHIFT-RETURN` 鍵 (先按住 `SHIFT` 鍵然後再按 `RETURN` 鍵) 來執行這些指令。文字細胞僅包含有文字信息, 所以不需經由核心來計算。圖細胞則包含有描繪圖及曲線圖。

我們可以把細胞格式化使其具備各式各樣的屬性, 如展示的字體之大小與顏色等。這些細胞也可用摘要的形式來組織一文件, 其方法如下: 先點取此細胞最右端的「右中括弧」, 再從工作視窗上點取 Format 選單中的 Style 然後選取 Title, Subtitle, Section, Subsection, $\cdots$ 即可。

接著我們介紹一下 MATHEMATICA 所用的慣用語法, 你可透過這些規則來了解它是怎麼樣發號司令的。

- 基本四則運算及指數所用的語法跟其他的程式語言完全一樣。
- 變數通常用小寫字母表示, 但也可以是一個字串如 `yvalue=` $\cdots$。
- 函數的變數 `x,y,...` 以中括弧 `[x,y,...]` 括起來, 而小括弧則用來達到分組的效果。
- 串列 (list) 是 MATHEMATICA 最原始的資料結構, 以大括弧 `{}` 括起來, 其中的元素則用逗點分開。如一維串列 `{1,2,3}` 爲一向量, 而二維串列 `{{1,2,3},{4,5,6}}` 則表矩陣其第一列就是第一個元素 `{1,2,3}`。
- 所有的指令 (除了符號之外), 包括內建函數與內建常數都是以大寫字母起頭, 如 `Sin[Pi/2]` 指的就是 $\sin(\pi/2)$。

---

[2]這是一個未命名的空白檔案, 名稱爲 untitled-1, 亦即未命名的第一個檔案。

[3]若程式過於龐大, 使用者得發很多時間從頭看起; 所以一旦加上註解, 使用者便可以清楚的知道程式的內容。一般的註解可直接寫在畫面上, 但如此一來電腦在執行計算時便容易產生混亂。因此我們可先點取最右端的「右中括弧」, 再從工作視窗上點取 Format 選單中的 Style 然後選取 Text 即可。

- 乘號是用 * 或是空格來代表, 如 a*b 或 a b。變數與整個括號相乘可不用留空格, 如 a(b-1)。數字與變數應注意前後次序, 如 7x 表示 7*x, 但 x7 表示一變數名爲 x7, 而 x 7 則表示乘積 x*7。

- 符號 = 意指代換, 如 t=1 , 而相等則以符號 == 表示之,
  如 Equal[x,t] 或 x==t 只有當 x 與 t 有相同的值才會爲眞。

- 否定指令 Not 可用 ! 表示之, 如 x!=t 爲眞若 x 與 t 有不同的值時。

- 上一個輸入以 % 表示之, 而 %n 指的是第 n 個輸入即 In[n]。所以 %% 意指上上一個輸入$\cdots$ 等等。

- 每一個細胞都是以 In[n]:= 開始的, 但你絕不可鍵入這些, 而只需鍵入你要的文字或指令, 因爲 MATHEMATICA 會自動在你執行 (即按下 SHIFT-RETURN 鍵) 之後將之冠在前頭的位置。

若想做進一步的探討, 可透過 Wolfram 的書 [23] 或其他入門書 [1, 11, 13, 24] 來達成你的美夢。或者也可以從 Help 視窗當中輸入你所要了解的指令, 詳細的介紹就會出現在你的眼前。注意 MATHEMATICA 對大小寫是敏感的 (case sensitive), 字母是大寫就必須大寫, 否則會出現錯誤信息。

## 2. 數論基本概念

- 整除 (Divides): $a$ 整除 $b$, 以符號 $a \mid b$ 表示之, 意即 $b$ 爲 $a$ 的整數倍。換句話說, $a$ 除 $b$ 之後沒有餘數。例如: $6 \mid 12$, $-14 \mid 98$, 然而 $7 \nmid 11$ (不整除) 。在理論性的論證當中, 通常我們把 $a \mid b$ 改寫成爲 $b = ak$ ($k$ 爲一整數) 較爲方便。在數學運算大師 MATHEMATICA 中與此有關的指令如下:

  – Divisors[$n$] 列舉整數 $n$ 所有的正因數。

- 質數(Primes):你知道他們是什麼玩意兒? $2, 3, 5, 7, 11, 13, 17, \cdots$ 而且你也知道有無窮多個質數。但你可知道

  「目前已知最大的質數是多少嗎?」

  答案當然不是「要有多大就有多大」, 爲什麼呢? 想想看, 隨便給你一個整數, 你能馬上回答此數就是質數嗎? 也許你說: 給我些許時間我就可以給你答案。好吧! 那你就隨便挑一個一百位數, 再試試看如何? 其實, 你的知道只是理論上的知道, 而不是實作上的知道。

判斷一整數是否爲質數乃是一個大挑戰, 但已被證實存在有演算法, 可在多項式時間內完成[3]。在此先介紹兩類有趣的質數。

1. 梅仙尼質數(Mersenne Primes):形如 $M_p = 2^p - 1$ 的質數稱之爲梅仙尼質數。此類質數與完全數有關, 目前已知最大的質數就是這一類的質數爲

$$2^{25964951} - 1。$$

這是已被發現的第四十二個梅仙尼質數, 由 GIMPS[4] 團隊於 2005 年 2 月 26 日發現。請進入 Chris K. Caldwell 所精心設計的質數網頁[5] 遨遊一番。

2. 費馬質數(Fermat Primes): 形如 $F_n = 2^{2^n} + 1$ 的質數稱之爲費馬質數。前面五個費馬質數爲

$$F_0 = 3, \quad F_1 = 5, \quad F_2 = 17, \quad F_3 = 257, \quad 與 \quad F_4 = 65537。$$

當年費馬因此推斷所有這一類型的數都是質數, 但奇怪的是, 這五個數是目前僅有的費馬質數。於是有人猜測說: 僅存在有限多個費馬質數。你認爲呢?

在數學運算大師 MATHEMATICA 中與質數有關的一些指令如下:

- `Prime[`$k$`]` 顯示第 $k$ 個質數。
- `Prime/@Range[`$k$`]` 列舉前 $k$ 個質數。
- `PrimePi[`$x$`]` 數論函數 $\pi(x)$, 就是 $\leq x$ 質數的個數。
- `PrimeQ[`$n$`]` 判斷整數 $n$ 是否爲質數。是則輸出 True, 否則輸出 False。此判別法實際上是一擬質數性 (pseudoprimality) 的判別法。爲一非常可信賴的方法, 而且已被證明至少正確到十位數。若要保證絕對正確的結果, 則用指令 `ProvablePrimeQ[`$n$`]`, 描述如下。
- `ProvablePrimeQ[`$n$`]`[6] 判斷整數 $n$ 是否爲質數。是則輸出 True, 否則輸出 False。這個判別法是一眞正質數性 (primality) 的判別法, 故其結果保證絕對正確。

例題01: 我們現在就用指令 `PrimeQ` 一同來確認上面所提的第五個費馬數眞是質數, 但第六則否。首先在 MATHEMATICA 中定義函數 $F_n = F[n]$, 然後列出 $F[n]$, $n = 4, 5$ 再判斷之。

---

[4]乃 The Great Internet Mersenne Prime Search 的縮寫。這是此大搜索團隊所發現的第八個, 之前的兩個爲 $2^{24036583} - 1$ 與 $2^{20996011} - 1$ 分別在 2004 年 5 月 15 日與 2003 年 11 月 17 日發現。詳情請進入其網站, 網址爲 http://www.mersenne.org/prime.htm。

[5]The Prime Pages, 網址爲 http://www.utm.edu/research/primes/largest.html。

[6]注意: 在下達這個指令之前, 得先載入「Number Theory」的套裝軟體, 指令如下:

```
<< NumberTheory`PrimeQ'
```

```
In[1]:= F[n_]:=2^(2^n)+1;{F[4],F[5]}
Out[1]= {65537, 4294967297}
In[2]:= PrimeQ[{F[4],F[5]}]
Out[2]= {True, False}
```

例題02: 由上 Out[1] 可知第五個費馬數是 5 位數, 但第六則變成 10 位數。這並不稀奇, 因爲從定義就可看出來後一個大約是前一個的平方。試列出前八個一起觀察並判斷是否爲質數。

```
In[3]:=Table[F[n],{n,0,7}]
Out[3]={3,5,17,257,65537,4294967297,18446744073709551617,
        340282366920938463463374607431768211457}
In[4]:=PrimeQ[Table[F[n],{n,0,7}]]
Out[4]={True, True, True, True, True, False, False, False}
```

- 最大公因數(Greatest Common Divisor): 兩整數 $a$ 與 $b$ 的最大公因數, 符號爲 $\gcd(a, b)$, 就是可以同時整除 $a$ 與 $b$ 的那個最大的正整數。例如: $\gcd(64, 14) = 2$, $\gcd(7, 11) = 1$, $\gcd(48, 60) = 12$。若 $\gcd(a, b) = 1$, 我們就稱 $a$ 與 $b$ 爲互質。衆所週知, 有兩個標準的方法可以用來求最大公因數。

  1. 因數分解法: 若你有辦法將 $a$ 與 $b$ 這兩個數分解成質因數的乘積, 那就做吧! 對每一個質因數, 看看它出現在 $a$ 與 $b$ 的分解式中的次冪。取兩者中較小的那個次冪, 然後將這些較小的質數次冪放在一起即可得到最大公因數。透過實例最容易瞭解這一切, 且看: $\gcd(5184, 189) = \gcd(2^6 3^4, 3^3 7) = 3^3 = 27$,

  $$\gcd(3528, 700) = \gcd(2^3 3^2 7^2, 2^2 5^2 7) = 2^2 7 = 28 (= 2^2 3^0 5^0 7^1)。$$

  注意: 若一質數不出現在任何一個分解式中, 則此質數當然就不會出現在這個最大公因數當中。

  2. 輾轉相除法: 若 $a$ 與 $b$ 是兩個大數, 所以有可能不容易分解因數。此時我們可透過所謂的歐幾里德演算法, 即輾轉相除法來計算最大公因數。這可追溯到每個人在小學時所學過的帶餘數的除法。舉個例子遠勝過千言萬語的解說或僅僅列出公式, 且看:

     例題03: 試計算 $\gcd(482, 1180)$。

解: 將 482 除 1180, 其商爲 2 而餘數爲 216 。然後將此餘數 216 除 482, 其商爲 2 而餘數爲 50 。再將此餘數 50 除前一個餘數 216, 其商爲 4 而餘數爲 16 。如此這般的進行下去, 即將最新的餘數除前一個餘數。最後那個非零的餘數就是所要求的最大公因數, 在此例爲 2:

$$\begin{aligned} 1180 &= 2 \cdot 482 + 216 \\ 482 &= 2 \cdot 216 + 50 \\ 216 &= 4 \cdot 50 + 16 \\ 50 &= 3 \cdot 16 + 2 \\ 16 &= 8 \cdot 2 + 0 \end{aligned}$$

注意: 觀察一下這些數的角色是怎麼個變動法,

餘數 $\longrightarrow$ 除數 $\longrightarrow$ 被除數 $\longrightarrow$ 退隱山林或下臺一鞠躬。

例題04: 試計算 $\gcd(1234567, 1111111)$。

解: 輾轉相除之得

$$\begin{aligned} 1234567 &= 1 \cdot 1111111 + 123456 \\ 1111111 &= 9 \cdot 123456 + 7 \\ 123456 &= 17636 \cdot 7 + 4 \\ 7 &= 1 \cdot 4 + 3 \\ 4 &= 1 \cdot 3 + 1 \\ 3 &= 3 \cdot 1 + 0 \end{aligned}$$

因此得到 $\gcd(1234567, 1111111) = 1$。

注意: 關於輾轉相除這個演算法, 值得一提的有

- 不需要將此兩數 $a, b$ 加以分解, 所以演算的速度相當快。
- 但是還有比這個方法更快的演算法, 稱之爲二進 GCD 演算法 (Binary GCD Algorithm), 由 Josef Stein 於 1961 年所提出的, 下面會特別介紹。

在數學運算大師 Mathematica 中與此有關的指令如下:

- `FactorInteger[n]` 將 $n$ 分解成質因數乘積之標準分解式。例如, `FactorInteger[90]`輸出爲`{{2,1},{3,2},{5,1}}` , 意表 $90 = 2 \cdot 3^2 \cdot 5$。
- `FactorInteger[n,FactorComplete->False]` 找出小於 $2^{16}$ 之 $n$ 的質因子。
- `GCD[`$a$, $b$`]` $a$, $b$ 的最大公因數, 即 $\gcd(a, b)$ 。

例題05: 試分解 $F_5, F_6, F_7$ 三個費馬數。

解: 執行 FactorInteger 指令如下:

```
In[5]:=Table[F[n], {n, 5, 7}] // FactorInteger
Out[5]={{{641,1},{6700417,1}},
        {{274177,1},{67280421310721,1}},
        {{59649589127497217,1},
         {5704689200685129054721,1}}}
```

所以得到此三個費馬數的分解式如下:

$$F_5 = 641 \times 6700417$$
$$F_6 = 274177 \times 67280421310721$$
$$F_7 = 59649589127497217 \times 5704689200685129054721$$

例題06: 試計算 $\gcd(123456789, 987654321)$。

解: 我們用上面的指令得到 $\gcd(123456789, 987654321) = 9$。

```
In[6]:= GCD[123456789,987654321]
Out[6]= 9
```

3. 二進GCD 演算法: 假設 $1 < a < b$ 爲二正整數。記得嗎? 在上面的輾轉相除法中, 我們所用到的基本原理是:

$$\gcd(a, b) = \gcd(a, b - ma),$$

其中所取的 $m$ 乃最大的正整數使得 $b - ma \geq 0$。此 $m$ 是透過長除法完成的, 但長除法之運算相對地說來是挺花時間的。若取 $m = 1$, 則速度當然會很快。另外除以 2 之運算也是極端快速的, 特別是在機器語言中。若 $a$ 與 $b$ 都是偶數, 則我們有

$$\gcd(a, b) = 2 \times \gcd(a/2, b/2)。$$

若 $a$ 爲奇數, $b$ 爲偶數, 則我們有

$$\gcd(a,b) = \gcd(a,b/2)。$$

所以得到這個二進GCD 演算法的步驟如下:

(a) 若 $a$ 與 $b$ 都是偶數, 則將二數同時除以 2 ; 亦即將 $a$ 與 $b$ 分別用 $a/2$ 與 $b/2$ 來取代。重複此步驟直到有奇數出現爲止。令 $r$ 爲執行此步驟的次數。

(b) 若有偶數, 將此偶數擺在 $b$ 的位置並將 $b$ 用 $b/2$ 來取代。重複此步驟直到變成奇數爲止。

(c) 若 $b = 1$, 則 $\gcd(a,b) = 2^r$。

(d) 若都是大1的相異奇數, 則將大數擺在 $b$ 的位置並將 $b$ 取代爲 $b-a$ 並回到步驟(b)。

(e) 若 $b = a$, 則 $\gcd(a,b) = 2^r \times a$。

爲方便起見, 我們用 $(a,b)$ 來表示 $\gcd(a,b)$。且看下面的例子:

- $(30,42) = 2(15,21) = 2(15,6) = 2(15,3) = 2(3,15) = 2(3,12) = 2(3,6) = 2(3,3) = 6$。
- $(56,72) = 2(28,36) = 2^2(14,18) = 2^3(7,9) = 2^3(7,2) = 2^3(7,1) = 2^3$。
- $(77,133) = (77,56) = (77,28) = (77,14) = (77,7) = (7,77) = (7,70) = (7,35) = (7,28) = (7,14) = (7,7) = 7$。

## 3. 整係數二元一次方程之整數解

在上面輾轉相除法中, 我們並沒有去驚動到商 $q_1, q_2, q_3, \ldots$, 甚至連碰都沒碰一下。感覺上, 他們應該很有用才對, 且看下面的分析。通常談到最大公因數時, 我們都會提到一個非常基本的事實: 給予二整數 $a$ 與 $b$, 必存在有整數 $x$ 與 $y$ 使得

$$ax + by = \gcd(a,b)。$$

換句話說, $a$ 與 $b$ 的最大公因數可以用這兩個數的一個線性組合來表示。在理論上, 這個事實有著很重要的應用, 特別是當 $a$ 與 $b$ 互質時。但怎麼證明呢? 可簡單描述如下: 考慮形如 $ax+by$ 之整數所成的集合。透過正整數之良序性, 我們知道這個集合有一最小的正整數, 姑且稱之爲 $d$。然後再利用長除法可證明這個集合中的每一個元素都是 $d$ 的一個倍數。由此很容易就可推論得到我們所要的結果 (試將完整的證明寫下來)。這樣的證明, 雖說是簡潔漂亮; 但美中不足的是, 在整個證明的過程當中, 我們尋不到怎麼樣去找 $x$ 與 $y$ 的蛛絲馬跡。我們要的乃是一個建構式

的證明 (constructive proof), 亦即在證明當中, 同時也能告訴我們怎麼樣去找 $x$ 與 $y$。在目前的例子中, 其演算法實際上是隱藏於輾轉相除法中。

所以讓我們重頭再來一遍: 令 $a = r_0, \quad b = r_{-1}$ 。輾轉相除之, 得到

$$r_{j-2} = q_j r_{j-1} + r_j, \quad j = 1, 2, \ldots, n+1。 \tag{1}$$

這意味著 $r_{n+1} = 0$ 且 $r_n = \gcd(a, b)$。若從第 $n$ 個除式開始逆推回去, 不難看出最後的餘數可用前面兩個餘數的一個線性組合來表示; 如此這般一直往前推, 到末了就得到我們所要的結果 (試著將完整的證明寫下來)。然而不同的是, 這一次我們實實在在的算出了 $x$ 與 $y$。

再看一次上面的例題03: $\gcd(482, 1180) = 2$, 我們會用到那兒計算時的一些數。透過其中的那些餘數 $216, 50, 16, 2$ 逆流而回到原先的兩個數 1180 與 482, 而最後得到最大公因數 2 爲 1180 與 482 的一個組合。由第四行開始, 我們有

$$2 = 50 - 3 \cdot 16,$$

所以我們可以把 2 表示成前兩個餘數的組合。往上移動一行, 我們可將餘數 16 寫成 216 與 50 的組合, 然後代入上式得

$$16 = 216 - 4 \cdot 50 \Longrightarrow 2 = 50 - 3 \cdot 16 = 50 - 3 \cdot (216 - 4 \cdot 50) = 13 \cdot 50 - 3 \cdot 216。$$

現在我們已經用過 gcd 演算過程中後面兩個非零餘數。將上一個未用到過的餘數 50 表示成 482 與 216 的組合, 然後代入上式得

$$2 = 13 \cdot (482 - 2 \cdot 216) - 3 \cdot 216 = 13 \cdot 482 - 29 \cdot 216。$$

最後將 216 代換成 $1180 - 2 \cdot 482$ 得到

$$2 = 13 \cdot 482 - 29 \cdot (1180 - 2 \cdot 482) = 71 \cdot 482 - 29 \cdot 1180。$$

此即最大公因數 2 表示成 1180 與 482 之組合的式子。只要輾轉相除時的計算式不太多的話, 那麼即使是用手算也不是太費工夫。但一般而言, 此法不太適用於搭配電腦來使用。算的時候似乎有點費工夫, 因爲沒有遞迴法則可言。

怎麼樣才能找到那一個更美、更好且更適合電腦來使用的方法呢? 我們知道每一個 $r$ 都是前兩個 $r$ 的組合, 因此可得到如同 $r_n$ 一樣的結論; 也就是說我們有

每一個 $r$ 都是 $a$ 與 $b$ 的線性組合。

這句話孕育著呼之欲出的 $x$ 與 $y$ 之遞迴公式。怎麼說呢? 且看: 令

$$r_j = x_j a + y_j b, \quad j = 1, 2, \ldots, n。 \tag{2}$$

比較 (1) 式及(2) 式, 得到 $j = 1$ 時我們有

$$x_1a + y_1b = r_1 = -q_1a + b \Rightarrow x_1 = -q_1, \quad y_1 = 1。$$

而 $j = 2$ 時, 我們卻有

$$x_2a + y_2b = r_2 = -q_2(x_1a + y_1b) + a \Rightarrow x_2 = -q_2x_1 + 1, \quad y_2 = -q_2y_1。$$

當 $j \geq 3$ 時我們終於有

$$\begin{aligned} x_ja + y_jb = r_j &= -q_j(x_{j-1}a + y_{j-1}b) + (x_{j-2}a + y_{j-2}b) \\ &= (-q_jx_{j-1} + x_{j-2})a + (-q_jy_{j-1} + y_{j-2})b, \end{aligned}$$

因而得到 $j = 3, 4, \ldots, n$ 時的遞迴公式就是:

$$x_j = -q_jx_{j-1} + x_{j-2}, \qquad y_j = -q_jy_{j-1} + y_{j-2}。$$

雖然這遞迴公式在 $j = 1, 2$ 時沒有意義, 但很容易可以看出只要我們定義

$$x_{-1} = 0, \quad x_0 = 1, \quad y_{-1} = 1, \quad y_0 = 0,$$

則所有的問題就煙消雲散。因而此二數列 $\{x_j\}$ 與 $\{y_j\}$ 可定義如下:

$$x_{-1} = 0, x_0 = 1, x_j = -q_jx_{j-1} + x_{j-2}, j = 3, 4, \ldots, n; \tag{3}$$

$$y_{-1} = 1, y_0 = 0, y_j = -q_jy_{j-1} + y_{j-2}, j = 3, 4, \ldots, n。 \tag{4}$$

由 (3) 及(4) 式, 很快就可以算出 $x_n$ 與 $y_n$ 因此得到 $a$ 與 $b$ 的最大公因數爲

$$\gcd(a, b) = x_na + y_nb。 \tag{5}$$

我們再一次看上面的例題03: $\gcd(482, 1180) = 2$, 這次用到公式 (5)。在例題03當中, 其商依序爲 $q_1 = 2$, $q_2 = 2$, $q_3 = 4$, $q_4 = 3$, $q_5 = 8$ 。我們可以很快算出 $x_j$ 及 $y_j$ 如下:

| $j$ | $-1$ | $0$ | $1$ | $2$ | $3$ | $4$ | $5$ |
|---|---|---|---|---|---|---|---|
| $q_j$ | | | $2$ | $2$ | $4$ | $3$ | $8$ |
| $x_j$ | $0$ | $1$ | $-2$ | $5$ | $-22$ | $71$ | |
| $y_j$ | $1$ | $0$ | $1$ | $-2$ | $9$ | $-29$ | |

所以我們得到

$$\gcd(a, b) = 71 \cdot 482 - 29 \cdot 1180。$$

注意: 上述的方法通常稱之爲延伸 (extended) 輾轉相除法, 在下面解某些同餘式時會用到。

在數學運算大師 MATHEMATICA 中與此有關的指令如下:

– `ExtendedGCD[` $a$, $b$ `]` 算出 $\gcd(a,b)$ 且將其表成 $a$ 與 $b$ 的組合。

例題07: 試求 $\gcd(482, 1180)$ 並表示成此二數的一個線性組合。

解: 我們用上面的指令得到 $\gcd(482, 1180) = 2$ 且

$$2 = 71 \cdot 482 - 29 \cdot 1180。$$

```
In[7]:= ExtendedGCD[482, 1180]
Out[7]= {2,{71,-29}}
```

例題08: 試求方程式 $123456789x + 987654321y = 9$ 的整數解。

解: 我們用上面的指令得到 $x = -8$ 與 $y = 1$。

```
In[8]:= ExtendedGCD[123456789, 987654321]
Out[8]= {9,{-8,1}}
```

## 4. 模算術

今有物不知其數, 三三數之剩二, 五五數之
剩三, 七七數之剩二, 問物幾何?[7] —孫子算經 [21]

在數論當中, 最基本又最有用的觀念之一就是模算術或同餘的觀念。我們就從孫子算經[8]說起。首先「三三數之剩二」是什麼意思呢? 那不過說某一個數 $x$ 被 3 除剩餘 2; 換句話說, $x - 2$ 被 3 整除。我們用下式表示「三三數之剩二」

$$x \equiv 2 \pmod 3。$$

一般而言, 令 $n$ 爲一正整數。對任意二整數 $a$ 與 $b$, 若 $n \mid a - b$ 則我們稱之爲在模 $n$ 之下, $a$ 同餘 $b$, 用符號

$$a \equiv b \pmod n$$

表示之。反之, $a \not\equiv b \pmod n$ 表示 $a, b$ 在模 $n$ 之下不同餘。例如: $52 \equiv 27 \pmod 5$, $-19 \equiv 37 \pmod 7$, $11 \not\equiv 91 \pmod{13}$。在理論性的論證當中, $a \equiv b \pmod n$ 通常改寫

[7]孫子算經下卷第 26 題, 就是這個孫子問題 [4] (第58頁)。俗稱韓信點兵, 也叫秦王暗點兵、鬼谷算、隔牆算、剪管術、神奇妙算、大衍求一術等等。

[8]約成書於第四世紀, 作者生平和編寫年代都不清楚。現今傳本共有三卷。卷上敍述記數的縱橫相間制度和乘除法則, 卷中舉例說明分數算法和開平方法。卷下第 31 題, 可以看作後世雞兎同籠題的始祖, 後來傳到日本, 變成鶴龜算。請參考下面網頁 http://www.edp.ust.hk/math/history/6/

爲 $b = a + nk$ ($k$ 爲一整數) 較爲方便。你可以將同餘式看成好像是等式一樣, 在式子的兩邊執行你所要的四則運算 (試證明之!) , 唯一需要小心的當然是除的時候。且看下面幾個例子, 即可明白是怎麼一回事。

例題09: 試求同餘方程式 $x + 11 \equiv 7 \pmod{17}$ 的解。

解: $x \equiv 7 - 11 \equiv -4 \equiv 13 \pmod{17}$ 。答案若寫成負的並沒什麼錯誤。但當我們在模 $n$ 之下工作時, 通常將最後的答案表示爲介於 0 與 $n - 1$ 之間的整數。

除法原理: 在模 $n$ 下, 只要除數與 $n$ 互質, 你就可以將兩數相除。

例題10: 試求同餘方程式 $4x + 11 \equiv 7 \pmod{17}$ 的解。

解: $4x \equiv 7 - 11 \equiv -4 \pmod{17}$, 所以 $x \equiv -1 \equiv 16 \pmod{17}$。被 4 除沒問題, 因爲 $\gcd(4, 17) = 1$。

例題11: 試求同餘方程式 $5x + 7 \equiv 11 \pmod{17}$ 的解。

解: $5x \equiv 4 \pmod{17}$。現在怎麼辦呢? 兩邊同時除以 5 就好了嗎, 但 $4/5$ 在模 17 之下是什麼意思呢? 我們知道 $4 \equiv 21 \equiv 38 \equiv 55 \equiv \cdots \pmod{17}$, 所以 $5x \equiv 4 \pmod{17}$ 與 $5x \equiv 55 \pmod{17}$ 是一樣的。現在我們可除以 5 得到 $x \equiv 11 \pmod{17}$ 爲其答案。注意 $4 \equiv 11 \cdot 5 \pmod{17}$, 所以在模 17 之下 11 就如同是 $4/5$ 一樣。

另解: 我們也可從另一角度來解此同餘方程式。因爲 $5 \cdot 7 \equiv 1 \pmod{17}$, 我們看出在模 17 之下 7 是 5 的乘法反元素。因此, 除以 5 可由乘以 7 來完成, 如下: $5x \equiv 4 \pmod{17} \Longrightarrow x \equiv 35x \equiv 28 \equiv 11 \pmod{17}$。

除法原理證明: 從上一節我們知道, 若 $\gcd(a, n) = 1$ 則存在有整數 $x$ 與 $y$ 使得 $ax + ny = 1$, 亦即 $ax \equiv 1 \pmod{n}$。因此在模 $n$ 之下, $x$ 是 $a$ 的乘法反元素, 故得證。

注意: 用上一節的延伸輾轉相除法即(3) 式來計算 $x$ 是相當省時的。$y$ 則不需要算, 因爲在模 $n$ 之下, 不管是多少都會被磨掉。

例題12: 試求同餘方程式 $1111111x \equiv 4 \pmod{1234567}$ 的解。

解: 從例題04得到 $q_1 = 1$, $q_2 = 9$, $q_3 = 17636$, $q_4 = 1$, $q_5 = 1$, $q_6 = 31$。因此可得 $x_{-1} = 0$, $x_0 = 1$, $x_1 = -1$, $x_2 = 10$, $x_3 = -176361$, $x_4 = 176371$, $x_5 = -352732$, 這告訴我們 $1111111 \cdot (-352732) + 1234567 \cdot y_5 = 1$, 由此得知 $1111111 \cdot (-352732) \equiv 1 \pmod{1234567}$。將原先的同餘式兩邊同時乘上 $-352732$ 得到 $x \equiv -1410928 \equiv 1058206 \pmod{1234567}$。在實作上, 這意味著如果我們在模 1234567 之下工作而且碰到分數 $4/1111111$, 那麼就以 1058206 取代之。這似乎是有點奇怪, 但不妨想

一想 4/1111111 指的是什麼呢? 這只不過是一個符號代表乘以 1111111 之後會得到 4 的那個數。當我們在模 1234567 之下時, 1058206 也具有這個性質, 因 $1111111 \times 1058206 \equiv 4 \pmod{1234567}$。

在數學運算大師 MATHEMATICA 中與此有關的指令如下:

– `Mod[`$k$, $n$`]` 將 $k$ 被 $n$ 除所得的餘數, 亦即 $k \pmod{n}$。

例題13: 試計算 $123 \cdot 456 \pmod{789}$。

解: 使用上面的指令, 得到答案爲 $69 \pmod{789}$。

```
In[9]:= Mod[123*456,789]
Out[9]= 69
```

總結上述討論過的一些東西, 摘要如下:

- 在模 $n$ 之下求乘法反元素 $a^{-1} \pmod{n}$
  1. 使用延伸輾轉相除法[9], 求出整數 $s$ 與 $t$ 使得 $as + nt = 1$。
  2. $a^{-1} \equiv s \pmod{n}$。
- 當 $\gcd(a, n) = 1$ 時, 解同餘方程式 $ax \equiv c \pmod{n}$

  ($\Leftrightarrow$ 當 $\gcd(a, n) = 1$ 時, 在模 $n$ 之下, 計算分數 $c/a$ 之值)
  1. 使用延伸輾轉相除法[10], 求出整數 $s$ 與 $t$ 使得 $as + nt = 1$。
  2. 答案爲 $x \equiv cs \pmod{n}$($\Leftrightarrow$ 將分數 $c/a$ 取代爲 $cs \pmod{n}$)。
- 當 $\gcd(a, n) = d > 1$ 時, 解同餘方程式 $ax \equiv b \pmod{n}$
  1. 若 $d \nmid b$, 則此同餘方程式無解。
  2. 若 $d \mid b$。考慮新的同餘方程式
  $$\frac{a}{d}\, x \equiv \frac{b}{d} \pmod{\frac{n}{d}}。$$
  因 $\frac{a}{d}$, $\frac{b}{d}$, $\frac{n}{d}$ 都是整數且 $\gcd(\frac{a}{d}, \frac{n}{d}) = 1$。按上述步驟得一解 $x_0$。
  3. 原同餘方程式 $ax \equiv b \pmod{n}$ 之解爲
  $$x_0,\ x_0 + \frac{n}{d},\ x_0 + 2\,\frac{n}{d}, \ldots,\ x_0 + (d-1)\frac{n}{d} \pmod{n}。$$

---

[9]僅求 $s$ 即可, 不需要算 $t$, 理由已經在上面的注意提過。
[10] 同上。

例題14: 試求同餘方程式 $12x \equiv 21 \pmod{39}$ 的解。

解: 因爲 $\gcd(12, 39) = 3$, 此最大公因數可以整除 21 。除以 3 得到一新的同餘方程式 $4x \equiv 7 \pmod{13}$ 。試幾個值, 或透過延伸輾轉相除法可得一解 $x_0 = 5$ 。所以原同餘方程式 $12x \equiv 21 \pmod{39}$ 之解爲

$$x \equiv 5,\ 18,\ 31 \pmod{39}。$$

## 5. 模次冪與連續平方法

在序曲 [19]中, 我們知道解密的過程裡需要執行如 $848^{187} \pmod{1189}$ 之計算。如果我們先算出848的187次方, 然後簡化至模1189之下; 那麼我們就得跟非常大的數打交道。將848自乘187次, 這是一個很大的數。表面上你得執行186次的乘法, 這相當花時間。所以在那邊, 我們提出了連續平方法來解決 $848^{187} \pmod{1189}$ 的計算問題。如此一來, 原先需執行186次乘法的工作, 現在只要12次就了結; 而且在整個計算過程當中絕對不會碰到一個比 $1189^2$ 還大的數。

輕而易舉的, 我們可將此一算法推廣用來計算 $a^b \pmod{n}$, 其中執行模 $n$ 下的乘法次數頂多是 $2\log_2(b)$, 而在整個計算過程當中絕對不會碰到一個比 $n^2$ 還大的數。這意味著次冪的運算可很快完成, 而且不需要用到太多的記憶體。

當 $a, b, n$ 都是 100 位數時這個方法是非常有用的。如果我們算出 $a^b$, 然後簡化至模 $n$ 之下, 那麼電腦的記憶體就有可能會溢位 (overflow): 想想看 $a^b$ 有多大呢? 其位數超過 $10^{100}$, 而此數又比宇宙中所有粒子的總數還要多。然而, 計算 $a^b \pmod{n}$ 依目前的演算法來估計不會超過 700 個步驟即可完成, 而且在整個計算過程當中絕對不會碰到一個超過 200 位的數。

令 $b_1b_2b_3\cdots b_w$ 爲整數次冪 $b$ 的二進位表示法 (如 $b = 1011$ 表示 $b_1 = 1, b_2 = 0, b_3 = 1, b_4 = 1$)。上述之演算法步驟如下: (注意 $r_w \equiv a^b \pmod{n}$)

1. 從 $k = 1$ 開始, 且令 $s_1 = 1$。
2. 若 $b_k = 1$, 令 $r_k \equiv s_k a \pmod{n}$; 否則令 $r_k = s_k \pmod{n}$。
3. 令 $s_{k+1} \equiv r_k^2 \pmod{n}$。
4. 若 $k = w$, 停止; 若 $k < w$, 則將 $k$ 增加 1 並回到步驟 2。

在數學運算大師 MATHEMATICA 中與此有關的指令如下:

– `PowerMod[`$a$, $b$, $n$`]` 將 $a^b$ 被 $n$ 除所得的餘數, 亦即 $a^b \pmod{n}$。

– `PolynomialMod[`$p$, $n$`]` 將多項式 $p$ 的係數化簡至 $\pmod n$ 下。

例題15: 試計算 $234567^{876543} \pmod{565656565}$。

解: 使用上面的指令, 得到答案爲 $473011223 \pmod{565656565}$。

```
In[10]:= PowerMod[234567,876543,565656565]
Out[10]= 473011223
```

例題16: 試計算 87878787 在模 9191919191 之下的乘法反元素。

解: 取 $-1$ 次方即乘法反元素, 故得 $7079995354 \pmod{9191919191}$。

```
In[11]:= PowerMod[87878787,-1,9191919191]
Out[11]= 7079995354
```

例題17: 試求同餘方程式 $7654x \equiv 2389 \pmod{65537}$ 的解。

解: 兩種解法如下:

```
In[12]:= Solve[{7654*x==2389,Modulus==65537},x,Mode->Modular]
Out[12]= {{Modulus->65537,x->43626}}
In[13]:= Mod[PowerMod[7654,-1,65537]*2389,65537]
Out[13]= 43626
```

## 6. 孫子定理 (又名中國剩餘定理)

現在回到孫子算經中「問物幾何?」的問題。「物不知其數, 三三數之剩二, 五五數之剩三, 七七數之剩二」可用同餘式表示如下:

$$\begin{cases} x \equiv 2 & \pmod 3 \\ x \equiv 3 & \pmod 5 \\ x \equiv 2 & \pmod 7 \end{cases}$$

「韓信點兵問題」就是求一組同餘式的公解。孫子算經不僅僅是提供了答案並給出了解決的方法。德國數學家高斯於 1801 年出版算術探究中明確寫出下述定理。1852 年英國基督教士 Alexander Wylie (1815-1887) 將孫子算經對此一問題的解法傳到歐洲,1874 年 L. Mathiesen 指出來孫子算經之解法符合高斯的定理, 從而在西方的數學史裏將這一個定理稱爲中國剩餘定理 (Chinese remainder theorem)。我們先敍述最簡單的版本, 然後再作一般的推廣。

孫子定理: 若 $m, n$ 爲互質的兩個正整數, 則對任意給予的正整數 $a, b$ 必存在唯一的正整數 $x \pmod{mn}$ 滿足聯立同餘方程組:

$$\begin{cases} x \equiv a & (\text{mod } m) \\ x \equiv b & (\text{mod } n) \end{cases}$$

例題18: 試求下列聯立同餘方程組的解:

$$\begin{cases} x \equiv 3 & (\text{mod } 7) \\ x \equiv 5 & (\text{mod } 15) \end{cases}$$

解: $x \equiv 80 \pmod{105}$ (注意: $105 = 7 \cdot 15$)。因爲 $80 \equiv 3 \pmod 7$ 且 $80 \equiv 5 \pmod{15}$, 所以 80 是一個解。中國餘數定理保證存在有解, 而且說到在模 $mn = 105$ 之下, 此解是唯一的。

## 怎麼找到這個答案的呢?

對小數 $m$ 與 $n$ 而言, 方法之一是按大小列出 在模 $n$ 之下與 $b$ 同餘的數, 直到你挑選到其中一個被 $m$ 除餘 $a$ 的數爲止。例如, 在模 15 之下與 5 同餘的數爲

$$5,\ 20,\ 35,\ 50,\ 65,\ 80,\ 95,\ \cdots$$

模 7 之後, 變爲 $5, 6, 0, 1, 2, 3, 4, \cdots$。因爲我們要的是 $3 \pmod 7$, 所以選取 80。對大數 $m$ 與 $n$ 而言, 列表比對的話有可能效率非常低。然而, 同樣的想法還是行得通的。在模 $n$ 之下與 $b$ 同餘的數爲 $b + nk$ ($k$爲一整數), 所以我們必須解同餘式 $b + nk \equiv a \pmod m$, 亦即

$$nk \equiv a - b \pmod{m}。$$

因爲我們假設 $\gcd(m, n) = 1$, 所以在模 $m$ 之下 $n$ 有乘法反元素 $n^{-1} \pmod m$。兩邊同乘此乘法反元素可得

$$k \equiv n^{-1}(a - b) \pmod{m}。$$

代回 $x = b + nk$, 然後簡化至模 $mn$ 之下, 即得其解。

例題19: 解下列聯立同餘方程組:

$$\begin{cases} x \equiv 7 & (\text{mod } 1234567) \\ x \equiv 11 & (\text{mod } 1111111) \end{cases}$$

解: 首先, 例題09的演算告訴我們

$$1111111^{-1} \pmod{1234567} \equiv -352732。$$

因此可得

$$k \equiv 1111111^{-1}(7-11) \equiv -352732 \cdot -4 \equiv 176361 \pmod{1234567}。$$

所以其解爲

$$x \equiv 11 + 1111111 \cdot 176361 \equiv 195956647082 \pmod{1371740973937}。$$

## 如何使用孫子定理?

如果你所面對的同餘方程式是在一合成數模 $n = \prod_{p|n} p^a$ 之下, 則解題的關鍵想法是, 按 $n$ 的分解式將方程式拆開成幾個在模 $p^a$ 之下的同餘方程式。解決完這些同餘方程式後, 再透過孫子定理重組得到在模 $n$ 之下的答案。優點是化繁爲簡, 通常分析在質數模或質數次冪模之下的同餘式遠比直接去面對合成數模之下的同餘式容易。

假設你要解同餘方程式 $x^2 \equiv 1 \pmod{35}$。因爲 $35 = 5 \cdot 7$, 所以拆開而成一對同餘方程式

$$\begin{cases} x^2 \equiv 1 \pmod{5} \\ x^2 \equiv 1 \pmod{7} \end{cases}$$

現在觀察得知 $x^2 \equiv 1 \pmod{5}$ 有兩個解: $x \equiv \pm 1 \pmod{5}$, 而 $x^2 \equiv 1 \pmod{7}$ 也有兩個解:$x \equiv \pm 1 \pmod{7}$ 。所以有四種不同的組合方式分別得到解如下:

$$\begin{aligned}
&x \equiv \ \ 1 \pmod{5}, \quad x \equiv \ \ 1 \pmod{7} \implies x \equiv \ 1 \pmod{35}, \\
&x \equiv \ \ 1 \pmod{5}, \quad x \equiv -1 \pmod{7} \implies x \equiv \ 6 \pmod{35}, \\
&x \equiv -1 \pmod{5}, \quad x \equiv \ \ 1 \pmod{7} \implies x \equiv 29 \pmod{35}, \\
&x \equiv -1 \pmod{5}, \quad x \equiv -1 \pmod{7} \implies x \equiv 34 \pmod{35}.
\end{aligned}$$

所以我們得知, 同餘方程式 $x^2 \equiv 1 \pmod{35}$ 的解爲 $x \equiv 1,\ 6,\ 29,\ 34 \pmod{35}$。

注意: 一般而言, 若 $n = p_1 p_2 \cdots p_r$ 爲 $r$ 個相異奇質數的乘積, 則同餘方程式 $x^2 \equiv 1 \pmod{n}$ 有 $2^r$ 個解。此乃下述更一般形式化之孫子定理所結出來的果實。

孫子定理(一般形式): 若 $m_1, m_2, \ldots, m_k$ 爲兩兩互質的 $k$ 個正整數, 則對任意給予的 $k$ 個整數 $a_1, a_2, \ldots, a_k$ 必存在唯一的整數 $x \pmod{m}$, 此處 $m = m_1 m_2 \cdots m_k$ 滿足聯立同餘方程組:

$$\begin{cases} x \equiv a_1 \pmod{m_1} \\ x \equiv a_2 \pmod{m_2} \\ \quad \vdots \qquad\qquad \vdots \\ x \equiv a_k \pmod{m_k} \end{cases}$$

例如, 此定理保證韓信點兵問題的聯立同餘方程組:

$$\begin{cases} x \equiv 2 & (\mathrm{mod}\ 3) \\ x \equiv 3 & (\mathrm{mod}\ 5) \\ x \equiv 2 & (\mathrm{mod}\ 7) \end{cases}$$

恰有一個解 $x \pmod{105}$。其實, 孫子給出答案爲「答曰: 二十三。」事實上, 這是最小的正整數解。他又說出其演算技巧爲「術曰: 三三數之剩二, 置一百四十; 五五數之剩三, 置六十三; 七七之數剩二, 置三十。幷之得二百三十三。以二百一十減之, 即得。凡三三數之剩一, 則置七十; 五五數之剩一, 則置二十一; 七七數之剩一, 則置十五。一百六以上, 以一百五減之, 即得。」這段話寫成數學式子就是:

$$\begin{aligned} x &= 2 \times 70 + 3 \times 21 + 2 \times 15 - 2 \times 105 \\ &= 140 + 63 + 30 - 210 \\ &= 23 \end{aligned}$$

如何找到這個解的呢? 其演算法如下 (其實這就是孫子定理的一個證明):

1. 算出 $m = m_1 m_2 \cdots m_k$。
2. 對 $j = 1, 2, \ldots, k$, 算出 $z_j = m/m_j$。
3. 對 $i = 1, 2, \ldots, k$, 利用延伸輾轉相除法算出 $y_i = z_j^{-1} \pmod{m_j}$。
4. 令 $x = a_1 y_1 z_1 + \cdots + a_k y_k z_k \pmod{m}$, 則 $x \equiv a_j \pmod{m_j}\ \forall j$。

在數學運算大師 MATHEMATICA 中與此有關的指令如下:

– `ChineseRemainderTheorem[ {a1, ..., ak}, {m1, ..., mk} ]`[11]

滿足下列聯立同餘方程組的唯一解 $x \pmod{m_1 m_2 \cdots m_k}$ :

$$\begin{cases} x \equiv a_1 & (\mathrm{mod}\ m_1) \\ x \equiv a_2 & (\mathrm{mod}\ m_2) \\ \quad \vdots & \qquad \vdots \\ x \equiv a_k & (\mathrm{mod}\ m_k) \end{cases}$$

[11]注意: 在下達這個指令之前, 得先載入「Number Theory」的套裝軟體, 指令如下:
`<<NumberTheory‘NumberTheoryFunctions‘`

例題20: 試解韓信點兵問題:

$$\begin{cases} x \equiv 2 & (\mathrm{mod}\ 3) \\ x \equiv 3 & (\mathrm{mod}\ 5) \\ x \equiv 2 & (\mathrm{mod}\ 7) \end{cases}$$

解:使用上面的指令得到答案爲 23。

例題21: 試解聯立同餘方程組:

$$\begin{cases} x \equiv 2 & (\mathrm{mod}\ 78) \\ x \equiv 5 & (\mathrm{mod}\ 97) \\ x \equiv 1 & (\mathrm{mod}\ 119) \end{cases}$$

解: 使用上面的指令得到答案爲 647480。

```
In[14]:= <<NumberTheory‘NumberTheoryFunctions‘
In[15]:= {ChineseRemainderTheorem[{2, 3, 2}, {3, 5, 7}],
  ChineseRemainderTheorem[{2, 5, 1}, {78, 97, 119}]}
Out[15]= {23, 647480}
```

驗算如下:

```
In[16]:= Mod[{23, 647480}, {{3, 5, 7}, {78, 97, 119}}]
Out[16]= {{2, 3, 2}, {2, 5, 1}}
```

## 7. 費馬小定理 (Fermat Little Theorem)

在序曲中, 我們用到了費馬小定理。此定理看似艱深, 其實只需用到大家耳熟能詳的數學歸納法及二項式定理, 而所得到的結果還更一般化。說明如下: 令 $p$ 爲一質數。對所有的正整數 $k$, 二項式定理告訴我們

$$(k+1)^p \overset{?_1}{\equiv} k^p + 1 \pmod{p}。$$

藉著數學歸納法馬上得證 $k^p \equiv k \pmod{p}$。另一方面, 若 $a$ 爲一負整數則 $a = -k$, 此 $k$ 爲一正整數。所以我們有 (請在下標問號處提供理由!)

$$a^p = (-k)^p \overset{?_2}{\equiv} -k^p \overset{?_3}{\equiv} -k = a \pmod{p}。$$

然而, $a = 0$ 此同餘式顯而易見, 故得證下述之定理。

定理: 若 $p$ 爲一質數, 則對所有的整數 $a$ 我們恆有

$$a^{p-1} \equiv 1 \pmod{p}。$$

換句話說, $p \mid (a^p - a) = a(a^{p-1} - 1)$。如果 $p \nmid a$, 那麼我們馬上得證 ($?_4$) 下述之費馬小定理。

費馬小定理: 若 $p$ 爲一質數且 $p \nmid a$, 則

$$a^{p-1} \equiv 1 \pmod{p}。$$

例如 $2^{10} = 1024 \equiv 1 \pmod{11}$。由此可算出 $2^{53} \pmod{11}$ 如下:

$$2^{53} = (2^{10})^5 2^3 \equiv 1^5 2^3 \equiv 8 \pmod{11}。$$

費馬小定理另一證明: 考慮集合 $F = \{1, 2, b, \ldots, p-1\}$ 並考慮將 $a$ 乘上 $F$ 中每一個元素之後所形成的集合

$$aF = \{a, 2a, 3a, \ldots, (p-1)a\}。$$

不難看出, 在模 $p$ 之下集合 $aF$ 中每一個元素會跟集合 $F$ 中的一個而且是唯一的一個元素同餘 ($?_5$)。所以得到

$$(p-1)!a^{p-1} = a(2a)(3a)\cdots(p-1)a \equiv (p-1)!,$$

故得證 ($?_6$)。

注意: 當我們在模11之下工作時。對次冪而言, 本質上我們是在模 10 而非模 11 之下工作。換句話說, 從 $53 \equiv 3 \pmod{10}$ 我們可推論而得到 $2^{53} \equiv 2^3 \pmod{11}$。

## 費馬小定理逆敘述成立嗎?

一般而言, 如果 $2^{n-1} \equiv 1 \pmod{n}$, 那麼 $n$ 就是質數。然而, 有例外: $561 = 3 \cdot 11 \cdot 17$ 爲一合成數, 但 $2^{560} \equiv 1 \pmod{561}$。怎麼知道的呢? 且看下面的分析: 因 $560 \equiv 0 \pmod{2}$, 故 $2^{560} \equiv 2^0 \equiv 1 \pmod{3}$。同樣地, 因 $560 \equiv 0 \pmod{10}$ 且 $560 \equiv 0 \pmod{16}$, 所以 $2^{560} \equiv 1 \pmod{11}$ 且 $2^{560} \equiv 1 \pmod{17}$。將此三同餘式合併可得 (爲什麼?)

$$2^{560} \equiv 1 \pmod{561}。$$

另一個例外是 $1105 = 5 \cdot 13 \cdot 17$。滿足 $a^n \equiv a \pmod{n} \forall a$ 的合成數 $n$ 稱之爲 Carmichael 數。這種數極其稀少, 但卻有無限多個 [2]。雖然如此, 這些例外在實作上是相當罕見的。因此, 如果 $2^{n-1} \equiv 1 \pmod{n}$ 那麼非常有可能 $n$ 是一個質數。

前面十個 Carmichael 數爲

$$561, 1105, 1729, 2465, 2821, 6601, 8911, 10585, 15841, 29341, \cdots。$$

將這十個 Carmichael 數分解後如下:

```
In[17]:=FactorInteger[{561,  1105,  1729, 2465, 2821, 6601,
                       8911, 10585, 15841, 29341}]
Out[17]={{{3,1},{11,1},{17,1}},{{ 5,1},{13,1},{17,1}},
         {{7,1},{13,1},{19,1}},{{ 5,1},{17,1},{29,1}},
         {{7,1},{13,1},{31,1}},{{ 7,1},{23,1},{41,1}},
         {{7,1},{19,1},{67,1}},{{ 5,1},{29,1},{73,1}},
         {{7,1},{31,1},{73,1}},{{13,1},{37,1},{61,1}}}
```

觀察得到所有這十個 Carmichael 數都是三個相異質數的乘積, 但不要誤以爲這是 Carmichael 數的特性, 因爲

$$62745 = 3 \times 5 \times 47 \times 89$$

是一個具有四個質因子的 Carmichael 數。然而不難證明所有的 Carmichael 數至少有下列兩個特性:

- 每一個 Carmichael 數都是奇數。
- 每一個 Carmichael 數都是相異質數的乘積。

### 費馬小定理反逆敘述

當然, 如果 $2^{n-1} \not\equiv 1 \pmod{n}$, 則 $n$ 不可能是質數。因爲計算 $2^{n-1} \pmod{n}$ 的速度奇快無比 (見第 9 節) , 這提供了我們一個尋找質數的方法。即, 選取一起始點 $n_0$ 並連續測試每一個大於 $n_0$ 的奇數 $n$, 看看是否 $2^{n-1} \equiv 1 \pmod{n}$ ? 若 $n$ 無法通過此測試, 那麼就丟掉此數並進行下一個 $n$。當一個 $n$ 通過此測試時, 再使用更細膩的技巧來測試 $n$ 的不可分解性。此法之優點在於其整個演算過程遠比去分解每個 $n$ 要快許多, 尤其是能很快的將許多的 $n$ 刪除。當然, 還有辦法可用來加速整個尋找的過程, 譬如說可先將包含有小質數因子的 $n$ 刪除, 然後再進行上述的方法。

## 8. 歐拉定理 (Euler's Theorem)

對合成數$n$我們亦需要類似費馬小定理的結果。令$\phi(n)$ 爲比$n$小且與$n$互質之正整數的個數。例如,$n = 10$ 則有四個這種整數, 即 1, 3, 7, 9。因此,$\phi(10) = 4$。若 $p$ 爲一質數而 $n = p^r$, 則除了 $p$ 的倍數 (也就是 $p, 2p, 3p, \cdots p^{r-1} \cdot p$) 之外, 其餘的數皆與 $n$ 互質。因此,

$$\phi(p^r) = p^r - p^{r-1} = \left(1 - \frac{1}{p}\right) p^n。$$

特別而言,

$$\phi(p) = p - 1。$$

更一般地, 可由孫子定理推論得知(試證明之!) 對任何正整數 $n$,

$$\phi(n) = n \prod_{p|n} \left(1 - \frac{1}{p}\right),$$

此處的乘積個數乃整除 $n$ 之相異質數的個數。當 $n = pq$ 爲兩相異質數的乘積時, 則 $\phi(pq) = (p-1)(q-1)$。

在數學運算大師 MATHEMATICA中與此有關的指令如下:

– `EulerPhi[` $n$ `]` 比 $n$ 小且與 $n$ 互質之正整數的個數, 亦即 $\phi(n)$。

```
In[18]:= EulerPhi[1000]
Out[18]= 400
```

根據定義我們知道, 總共有 $\phi(n)$ 個比 $n$ 小且與 $n$ 互質的正整數, 說是 $\Phi = \{b_1, b_2, b_3, \ldots, b_{\phi(n)}\}$ 好了。假設 $a$ 是任意與 $n$ 互質的一個整數並考慮將 $a$ 乘上 $\Phi$ 中每一個元素之後所形成的集合

$$a\Phi = \{ab_1, ab_2, ab_3, \ldots, ab_{\phi(n)}\}。$$

不難看出, 在模 $n$ 之下集合 $a\Phi$ 中每一個元素會跟集合 $\Phi$ 中的一個而且是唯一的一個元素同餘 (試說明原因!)。所以得到

$$a^{\phi(n)}(b_1 b_2 b_3 \cdots b_{\phi(n)}) = (ab_1)(ab_2)(ab_3)\cdots(ab_{\phi(n)}) \equiv b_1 b_2 b_3 \cdots b_{\phi(n)},$$

因此得證 (試說明原因!) 歐拉定理如下:

歐拉定理: 若 $\gcd(a, n) = 1$, 則 $a^{\phi(n)} \equiv 1 \pmod{n}$。

注意: 當 $n = p$ 爲一質數, 則歐拉定理 $=$ 費馬小定理。

例題22: 試問 $7^{803}$ 的最後三位數爲何?

解: 所要求的乃是被 1000 除的餘數, 因此我們必須在模 1000 之下工作。因爲 $\phi(1000) = 1000\left(1-\frac{1}{2}\right)\left(1-\frac{1}{5}\right) = 400$, 所以

$$7^{803} = (7^{400})^2 \cdot 7^3 \equiv 1^2 \cdot 343 = 343 \pmod{1000}。$$

因此 $7^{803}$ 的最後三位數爲 343。

注意: 在此我們可將次冪由 803 改爲 3, 因 $803 \equiv 3 \pmod{\phi(1000)}$。

基本原理: 在模 $n$ 下工作時, 次冪的地方必須在模 $\phi(n)$ 之下運作。

這個極其重要的原理我們會一再的使用。所以要好好的思考上面的那些例子, 直到你自己確信在次冪的地方一定要模 $400 = \phi(1000)$ 才算完成正確 (也就是說, 不要去當那許許多多的大傻瓜, 錯誤地試著將次冪放在模 1000 之下來運作)。

## 9. 原根 (Primitive Roots)

先觀察一下, 在模 11 之下 7的次冪:

| $j$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| $7^j$ | 7 | 5 | 2 | 3 | 10 | 4 | 6 | 9 | 8 | 1 |

一眼即可看出, 在模 11 之下所有非零元素都是 7 的次冪。此種元素 7 稱之爲模 11 的一個原根 (primitive root)[12]。同樣地, 每一個模 23 的非零元素都是 5 的次冪, 所以 5 是模 23 的一個原根。然而, $2^{11} \equiv 1 \pmod{23}$, 所以只有 1, 2, 4, 8, 16, 9, 18, 13, 3, 6, 12 是 2 的次冪。因此 2 不是模 23 的一個原根。模 23 的原根有 5, 7, 10, 11, 14, 15, 17, 19, 20, 21。

一般而言, 當 $p$ 爲一質數, 模 $p$ 的一個原根就是其中一個非零元素 $g$ 使得每一個模 $p$ 的非零元素都是 $g$ 的一個次冪。每一個質數 $p$ 都有 $\phi(p-1)$ 個原根。特別而言, 至少有一個原根。在實作上, 如果知道 $p-1$ 的分解式, 那麼要找到一個原根並不難。見一般習題 8。

在數學運算大師 MATHEMATICA 中與此有關的指令如下:

- `PrimitiveRoot[`$p$`]`[13] 給予質數模 $p$ 之下的最小原根。( $p$ 也可能是一奇質數的次冪或是一奇質數次冪的兩倍)

---

[12]「乘法生成元素 (multiplicative generator)」可能更傳神, 但不普遍爲人採用。
[13]下達這個指令前, 需先載入「`Number Theory`」的套裝軟體如前面`In[14]`所示。

```
In[19]:= PrimitiveRoot[{65537,11,23}]
Out[19]= {3,2,5}
```

我們將密碼學當中所要用到原根方面的主要結果摘要在此。

定理: 令 $g$ 爲質數 $p$ 的一個原根。

- 若 $n$ 爲一整數, 則 $g^n \equiv 1 \pmod{p} \iff n \equiv 0 \pmod{p-1}$。
- 若 $j$ 與 $k$ 爲二整數, 則 $g^j \equiv g^k \pmod{p} \iff j \equiv k \pmod{p-1}$。

證明: 若 $n \equiv 0 \pmod{p-1}$, 則存在一整數 $m$ 使得 $n = (p-1)m$。
因此透過費馬小定理, 我們有

$$g^n \equiv (g^m)^{p-1} \equiv 1 \pmod{p}。$$

相反地, 假設 $g^n \equiv 1 \pmod{p}$。我們要證明 $p-1$ 整除 $n$。很自然地, 我們將 $n$ 除以 $p-1$ 得到 $n = (p-1)q + r$, 其中 $0 \le r < p-1$。我們有

$$1 \equiv g^n \equiv (g^q)^{p-1} g^r \equiv 1 \cdot g^r \equiv g^r \pmod{p}。$$

假設 $r > 0$。考慮所有 $g \pmod{p}$ 的次冪即可察覺, 我們頂多就只有 $r$ 個元素。但 $r < p-1$, 因此並非所有模 $p$ 之下的非零元素都是 $g$ 的一個次冪。這與 $g$ 是原根的假設矛盾。剩下來唯一的可能性是 $r = 0$。所以 $n = (p-1)q$, 也就是 $p-1$ 整除 $n$。這證明了第一部份。

對第二部份, 假設 $j \ge k$ (否則將 $j$ 與 $k$ 對調)。我們有

$$g^j \equiv g^k \iff g^{j-k} \equiv 1 \iff j - k \equiv 0 \iff j \equiv k \pmod{p-1}。$$

## 10. 模 $n$ 之下的逆方陣

根據模 $n$ 之下的除法原理, 我們知道求模 $n$ 之下的逆方陣可用一般求逆方陣的方法來完成。所需要的基本事實如下:

基本事實: 在模 $n$ 之下, 方陣 $M$ 是可逆的 $\iff \gcd(\det(M), n) = 1$。

在此我們僅處理小方陣 (2或3階) , 此乃因爲在我們所要介紹的密碼學當中這已綽綽有餘。在這種情況之下, 求模 $n$ 之下的逆方陣最容易的方式就是使用有理數, 然後再轉換回模 $n$ 之中的數。衆所週知, 一個佈於整數的方陣其逆方陣可寫成另一佈於整數的方陣除以其行列式

值。因爲我們假設其行列式值與 $n$ 互質, 所以可求出此行列式值的乘法反元素。例如在 $2\times 2$ 的情況中, 一般的公式爲

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}^{-1} = \frac{1}{ad-bc}\begin{pmatrix} d & -b \\ -c & a \end{pmatrix},$$

所以我們需要求出 $ad-bc$在模 $n$之下的乘法反元素。

例題23: 試求 $\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \pmod{11}$ 的逆方陣。

解: 因爲 $ad-bc=-2$ 在模 11 之下的乘法反元素爲 5 , 所以我們可將 $-\frac{1}{2}$ 用 5 來代換得到

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}^{-1} \equiv -\frac{1}{2}\begin{pmatrix} 4 & -2 \\ -3 & 1 \end{pmatrix} \equiv 5\begin{pmatrix} 4 & -2 \\ -3 & 1 \end{pmatrix} \equiv \begin{pmatrix} 9 & 1 \\ 7 & 5 \end{pmatrix} \pmod{11}。$$

驗算如下, 得知無誤:

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}\begin{pmatrix} 9 & 1 \\ 7 & 5 \end{pmatrix} \equiv \begin{pmatrix} 23 & 11 \\ 55 & 23 \end{pmatrix} \equiv \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \pmod{11}。$$

例題24: 試求 $M=\begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 3 \\ 1 & 4 & 9 \end{pmatrix} \pmod{11}$ 的逆方陣。

解: 因爲 $\det(M)=2$ 在模 11 之下的乘法反元素爲 6, 所以我們可將 $\frac{1}{2}$ 用 6 來代換得到

$$M^{-1}\equiv\frac{1}{2}\begin{pmatrix} 6 & -5 & 1 \\ -6 & 8 & -2 \\ 2 & -3 & 1 \end{pmatrix}\equiv 6\begin{pmatrix} 6 & -5 & 1 \\ -6 & 8 & -2 \\ 2 & -3 & 1 \end{pmatrix}\equiv\begin{pmatrix} 3 & 3 & 6 \\ 8 & 4 & 10 \\ 1 & 4 & 6 \end{pmatrix} \pmod{11}。$$

## 為何我們需要行列式值與 $n$ 互質呢?

假設 $MN\equiv I \pmod{n}$, 此處 $I$ 爲單位方陣。則

$$\det(M)\det(N)\equiv\det(MN)\equiv\det(I)=1 \pmod{n}。$$

因此 $\det(M)$ 在模 $n$ 中有乘法反元素, 亦即 $\det(M)$ 與 $n$ 必須互質。

例題25: 試求 $M=\begin{pmatrix} 13 & 12 & 35 \\ 41 & 53 & 62 \\ 71 & 68 & 10 \end{pmatrix} \pmod{999}$ 的逆方陣。

解: 若使用 `Mod`在 `Inverse` 上, 得不到眞正的答案:

```
In[20]:=Mod[Inverse[{{13,12,35},{41,53,62}, {71,68,10}}],999]
Out[20]={{3686/34139, 34102601/34139, 1111/34139},
         {34100869/34139, 2355/34139, 34104232/34139},
         {975/34139, 32/34139, 34104664/34139}}
```

所以改用 PolynomialMod 在 Inverse 上, 則得到答案如下:

```
In[21]:= PolynomialMod[Inverse[{{13,12,35},{41,53,62},
          {71,68,10}}] ,999]
Out[21]= {{772,472,965},{641,516,851},{150,133,149}}
```

## 參考文獻


1. Abell, Martha L./Braselton, James P.: *Mathematica by Example*, Second Edition, Academic Press, San Diego, 1997.
2. Alford, W. R./Granville, A./Pomerance, C.: There are Infinitely Many Carmichael Numbers, *Ann. Math.* 139, 703-722, 1994. 全文見網頁 http://www.dms.umontreal.ca/~andrew/agpapers.html
3. Agrawal, Manindra/Kayal, Neeraj/Saxena, Nitin: PRIMES is in P. 見網頁 http://www.cse.iitk.ac.in/news/primality.html
4. Andrews, George E.: *Number Theory*, W. B. Saunders Co., Philadelphia, PA, 1971, Reissued, Dover, New York, 1995.
5. Apostol, Tom M.: *Introduction to Analytic Number Theory*, Undergraduate Texts of Mathematics, Springer-Verlag, New York, First Edition, 1976, Corr. Fifth Printing, 1998.
6. Bressoud, D. M.: *Factorization and Primality Testing*, Undergraduate Texts of Mathematics, Springer-Verlag, New York, 1989.
7. Buchmann, Johannes A.: *Introduction to Cryptography*, Springer-Verlag, Second Edition, 2004.
8. CAS網頁 http://www.rbjones.com/rbjpub/cs/ai031.htm#introduction
9. Clock算術網頁 http://en.wikipedia.org/wiki/Modular_arithmetic 或 http://www.math.csusb.edu/faculty/susan/modular/modular.html
10. Cohen, Henri: *A Course in Computational Algebraic Number Theory*, Volume 138 of *Graduate Texts in Mathematics*, Springer-Verlag, New York, First Edition, 1993, Fourth Printing, 2000.
11. Gaylord, Richard J./Kamin, Samuel N./Wellin, Paul R.: *Introduction to Programming with Mathematica*, Springer-Verlag, New York, Second Edition, 1996.
12. Hardy, G.H.: *A Mathematician's Apology*, Cambridge University Press, London, 1940. 摘要見網頁 http://en.wikipedia.org/wiki/A_Mathematician%27s_Apology
13. 洪維恩, 數學運算大師 Mathematica 4, 碁峰資訊,2001年5月。
14. 華羅庚, 數論導引, 北京科學出版社,1957年。
15. Ireland, Kenneth F./Rosen, Michael I.: *A Classical Introduction to Modern Number Theory*, Volume 84 of *Graduate Texts in Mathematics*, Springer-Verlag, New York, Second Edition, 1990, Corr. Fifth Printing, 1998.
16. 莫宗堅, 韓信點兵, 科學月刊第一卷第一期,1970年1月。全文見網頁 http://episte.math.ntu.edu.tw/articles/sm/sm_01_01_2/
17. Rosen, Kenneth H.: *Elementary Number Theory and Its Applications*, Addison-Wesley, Fifth Edition, 2005.

18. Schroeder, M.R.: *Number Theory in Science and Communication*, Springer-Verlag, Third Edition, 1997, Corr. Second Pprinting, 1999.
19. 沈淵源, 近代密碼學序曲, 數學傳播第二十七卷第一期 (105), 92年3月, 第68–74頁。全文見網頁 `http://www.math.sinica.edu.tw/math_media/vol.phtml?voln=271`
20. Silverman, Joseph H.: *A Friendly Introduction to Number Theory*, Prentice Hall, Third Edition, 2006.
21. Sun-Tsu(孫子): 孫子算經, 收入《宋刻算經六種》, 上海文物出版社,1980 年。
22. Wolfram 介紹網頁 http://www.wolfram.com/products/mathematica/introduction.html
23. Wolfram, Stephen: *The Mathematica Book*, Fifth Edition, Version 5, Wolfram Media, 2003.
24. 余家銘, Mathematica 程式設計風格與應用, 文魁資訊,2002年7月。

—本文作者任教於私立東海大學數學系—