數學傳播　42卷2期, pp. 3-24

電腦與數學：問題與展望

演講者：Ronald Graham

Ronald Graham

時　間：民國 104 年 10 月 5 日
地　點：天文數學館 202 演講廳
整　理：編輯室

李瑩英教授：我們非常高興能夠邀請 Ronald Graham 教授訪問台灣大學, 並在今年度的數學沙龍演講。Graham 教授現爲聖地牙哥 California Institute for Telecommunications and Information Technology 首席科學家, 加州大學聖地亞哥分校計算機科學與工程系 Irwin 和 Joan Jacobs 教授。他是美國國家科學院院士和美國藝術與科學院院士, 並曾獲頒美國數學會 Pólya 獎和 Steele 獎等多項榮譽和獎項。他是電腦科學教授, 也是數學家, 被美國數學學會譽爲『近年來讓離散數學在全球迅速發展的主要建築師之一』。他在調度理論 (scheduling theory)、計算幾何、Ramsey 理論和準隨機性 (quasi-randomness) 方面曾做出重要的貢獻。他也爲數學界服務良多。1993 年至 1995 年, 他擔任美國數學學會會長兼 MAA 的主席。特別的是, 他還曾是國際雜耍協會的前任主席, 之後赴聖地亞哥, 在貝爾實驗室和 AT&T 實驗室工作了 30 多年。今天他要講電腦和數學：問題和展望。讓我們歡迎 Graham 教授。

Graham: 謝謝李教授。各位來賓, 各位同學, 很高興有這個機會和大家一起討論。我本來希望能用中文來演講, 可是我的中文不太好, 希望下一次我來的時候可以用中文來說。謝謝大家!

我今天要做的, 是講一些電腦和數學互動的例子。我想大家都知道, 電腦越來越強大, 演算法越來越好, 問題越來越困難。這個演講的首要主題將是：受到電腦極大幫助的問題、電腦可能幫得上忙的問題, 以及電腦似乎從來沒幫上忙的問題。

這是上個世紀最偉大的數學家 David Hilbert 的照片。除了在 1900 年數學家大會上提出的 23 個問題, 他還寫了很多文章討論『問題對數學的功用』。他說:「只要科學的某個分支能提供大量的問題, 它就會生機盎然。」一旦問題逐漸竭盡, 也許它就邁向終結。而經由解決這些新問題, 你開發了新的工具來破解更進一步的問題。

什麼是好問題? 那應該是難題, 而不應該微不足道, 但也不應該不可能解決。往往事先很難判斷: 一個問題是否會在下星期內解決, 或者是否需要一百年。我將關注電腦對數學的影響, 討論一些問題。

David Hilbert

1. 質數及數論

如果正整數 p 的因數僅有 1 和 p, 則 p 是質數。例如, 2, 3, 5, 7 和 11 等都是所謂的質數, 14, 15 和 100 不是質數, 因爲它們有介於 1 和數本身之間的因數。

爲什麼質數很重要? 它們是建造整數的積木。算術基本定理說: 你可以用唯一的方式把每個整數寫成質數的乘積 (不計質數順序)。

有幾個質數? 無限多。我想大家都看過下述證明 (據說它是由 Euler 提出, 但我認爲可能出現在更早之前)。假設只存在有限個質數 $p_1, \ldots, p_m$, 將它們的乘積加 1 得到 $p_1 \cdot p_2 \cdots p_m + 1$, 這個數不能被任何這些質數整除, 所以必然有一些新的質數。

目前爲人所知的最大質數是: $2^{57,885,161} - 1$, 有 17,425,170 個十進制數字, 幾年前被發現, 是 "Great Mersenne Prime Internet Search Project" 的一部分[1]。這 project 是一套演算法, 用以測試特定的指數 (比如 5,700 萬等等) 是否產生質數。大約有五十萬個處理器正在運作這套演算法。爲什麼人們關心這件事? 有人提供 $150,000 給超過一億個十進制數字的質數。他們對一千萬個十進制數字的質數也提供類似的獎金。

爲什麼人們關心質數? 當今在資訊安全和密碼學中, 質數尤爲重要。現在觀察一下, $4 = 2 + 2$, $6 = 3 + 3$, $8 = 3 + 5$, $10 = 3 + 7 = 5 + 5$, $12 = 5 + 7$, $14 = 3 + 11 = 7 + 7$, $19 = 3 + 13 = 5 + 11$ 等等, 事實上, Goldback 於 250 年提出下述

Goldback 猜想 (1742): 每一個偶數都是兩個質數的總和。

它已被驗證於小於 10^{18} 的整數。看起來很不錯。事實上, 數字越大, 你可以用更多方式把它寫成兩個質數的總和。幾年前有一家出版社提供 100 萬美元獎金給這個問題的解決方案, 但有一些限制, 必須至少 18 歲才能獲頒這個獎金。如果 18 歲以下的人可以提出證明, 他們不是應該

[1]編註. 至 2018 年 5 月, 已知之最大質數爲 $2^{77,232,917} - 1$, 有 23,249,425 個十進制數字, 於 2016 年一月由 Great Internet Mersenne Prime Search (GIMPS) 發現。

得到更多獎金嗎? 四十年前, 陳景潤證明:

定理 (**陳景潤**, 1973). *每個夠大的偶數都可寫爲 $A+B$, 其中 A 是質數, 而 B 的質因數不超過兩個。*

非常接近了。幾年前又有一次進展; Helfrott 證明:

定理 (Helfgott, 2013). *大於 5 的每一個奇數都是三個質數的總和。*

之前的下界約 $10^{1,346}$, 他下修了下界, 填補了缺口。

另外, 你可以輕易找到相差 2 的質數對, 這些對質數被稱爲 孿生質數。譬如 3 和 5, 11 和 13, 41 和 43 都是孿生質數。有個迄今尚未解決的

問題:*有無限多對孿生質數嗎?*

似乎是有的。這是目前爲人所知的最大一對孿生質數:

$$3,756,801,695,685 \cdot 2^{666,669} \pm 1\,。$$

它們有 200,700 個十進制數字[2]。我想大家都聽說過張益唐, 對吧? 他的故事很動人。他證明了

定理 (**張益堂**, 2013):*存在常數 $N \leq 7 \cdot 10^7$, 使間隙爲 N 的相鄰質數有無限多對。*

張益堂

但他並沒有試圖讓 N 盡可能地小。於是人們非常努力去縮小 N。目前的記錄是 $N \leq 246$。有無限多對相異的質數彼此差不到 246。當然, 你希望把 N 下修爲 2。你需要一些新的想法。

算術基本定理說： 每一個整數 N 都能以唯一的方式寫爲質數的乘積。真的做得到嗎? 例如, 11 是質數, $111 = 3 \cdot 37$ 不是質數。1,111 呢? $1,111 = 11 \cdot 101$。11,111 呢? 隨著數字越來越大, 逐漸有點難度。$11,111 = 41 \cdot 271$。那麼, 111,111 呢? 這是質數嗎? 不, 它不是質數, $111111 = 3 \cdot 7 \cdot 11 \cdot 13 \cdot 37$。好, 1,111,111 呢? 現在更難了吧? 你必須檢查它不能被 3 整除, 檢查 5, 7 等等。原來是 239。

$$1,111,111 = 239 \cdot 4639。$$

連續 71 個 1 呢? 事實上, 這是兩個質數的乘積:

11,111,111,111,111,111,111,111,111,111,111,111,111,111,111,111,111,111,111,111,111,111,111,111,111,111,111

$= 24157314239362767357695743904 9 \cdot 45994811347886846310221728895223034301839$

並不總是如此, 只是恰好發生。

[2]編註: 至 2018 年 5 月, 已知之最大孿生質數爲 $2996863034895 \cdot 21290000 \pm 1$, 有 388,342 個十進制數字。

The Assault on 114,381,625,757,888,867,669,235,779,976,146,612,010,218,296,721,242,362,562,561,842,935,706,935,245,733,897,830,597,123,563,958,705,058,989,075,147,599,290,026,879,543,541

By GINA KOLATA

Code was believed unbreakable a decade ago, but not anymore

Assault on Big Number Said to Be Near Success

NYTimes March 22, 1994

Ron Rivest

這是紐約時報的一篇文章, 有這129位數字:

114,381,625,757,888,867,669,235,779,976,146,612,010,218,296,721,242,362,562,561,842,935,706,935,245,733,897,830,597,123,563,958,705,058,980,075,147,599,290,026,879,543,541

上圖右下是 Ron Rivest 的照片。網路上最常用的加密數據方法, 援用所謂的 RSA 加密系統, 其安全性奠基於對大整數做質因數分解的明顯困難。RSA 的 R 代表 Rivest[3]。他向世人提出挑戰, 寫下上述的 129 位數字, 並宣稱沒有人會在 40 萬億 (quadrillion, 10^{16}) 年的時間內分解這個數。但實際上破解這問題只花 20 年。他的估計稍有偏差。

這是質數:

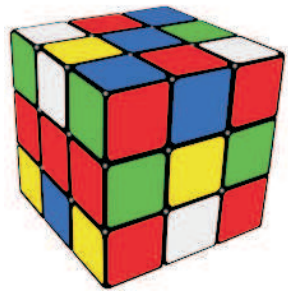
$$\overbrace{11, 111, 111, 111, 111, \ldots, 111, 111}^{1{,}031個}$$

這是最後一個這種形式的質數嗎? 不, 沒有人知道如何證明這件事。似乎有無限多這種形式的質數。而這似乎是一個質數:

$$\overbrace{1, 111, 111, 111, 111, \ldots, 111, 111}^{270{,}343個}$$

它至少通過了些測試, 表現得像一個質數, 但沒有人能夠證明它是質數。

[3]編註: RSA 是 1977 年由 Ron Rivest、Adi Shamir 和 Leonard Adleman 一起提出的。當時他們三人都在麻省理工學院工作。RSA 是他們三人姓氏開頭字母組成的。



這是一個魔術方塊。電腦爲它做了些事。我們的問題是: 回復原狀需要多少步驟? 所謂上帝的演算法 (God's algorithm) 是最好的演算法。可能的布局共有 43 quintillion (10^{18}) 種, 但 Google 有很多伺服器。Google 說最多需要多少步驟? 20; 做 20 個動作, 你就可以讓它回復原狀。你要怎麼做這 20 步驟? 我想大家並不知道怎麼做。他們把它放在所有的伺服器上, 檢查所有的位置, 從而得到答案。

右邊的傢伙呢? 它約有 10^{160} 種可能的布局。電腦能找到最好的演算法嗎? 我不認爲。它超出計算的範圍。如我們所知, 電腦能計算 10^{20} 筆資料, 但不能運算 10^{100} 筆資料。

現在有個數學和電腦的有趣互動。這裡有 14 個分數:

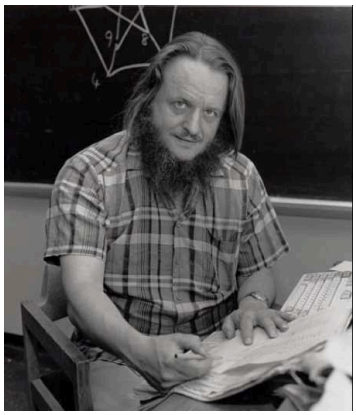
$$\frac{17}{91}, \frac{78}{85}, \frac{19}{51}, \frac{23}{38}, \frac{29}{33}, \frac{77}{29}, \frac{95}{23}, \frac{77}{19}, \frac{1}{17}, \frac{11}{13}, \frac{13}{11}, \frac{15}{14}, \frac{15}{2}, \frac{55}{1}$$

你應該如何運作做這些分數? 從數字 2 開始, 重複以下過程 : 乘以第一個使乘積爲整數的分數。如果你這樣做, 啊, $\frac{15}{2}$, 是的, 這是清單中的第一個使乘積爲整數的分數。好吧, 現在重複這過程, 爲 15 清查清單, 15 要乘上 $\frac{15}{1}$。這是第一個合用的分數。每次當你得到一個整數, 就繼續這樣做。你去清單找, 乘上第一個使乘積爲整數的分數, 得到一個整數, 然後繼續。

$$\begin{aligned} 2 \to 2 \cdot \frac{15}{2} = 15 \to 15 \cdot \frac{55}{1} = 825 \to 825 \cdot \frac{29}{33} = 725 \\ \to \cdots = 364 \to 364 \cdot \frac{17}{91} = 68 \\ \to 68 \cdot \frac{1}{17} = 4 = 2^2 \\ \to 4 \cdot \frac{15}{2} = 30 \to \cdots \to 8 = 2^3 \\ \to \cdots \to 2^5 \to \cdots \to 2^7 \to \cdots \to 2^{11} \to \cdots \end{aligned}$$

如此進行幾個步驟, 會得到 4, 這是 2^2。好, 把這個指數 2 寫下來, 繼續下去。接著你會得到 2^3, 繼而你會得到 2^5。你寫下所有你得到的2的指數: 2, 3, 5, 7, 11, 13, 19。如你所知, 這些是質數! 這個過程產生所有的質數, 且恰好產生一次。

我曾對解決了某 Hilbert 問題的數學家 (哈佛的傑出教授) 展示這件事, 他說 :「對, 這 14 個分數可能會給你頭 100 個或頭 200 個質數, 但是它們能給出所有質數?」好吧, 他是對的。



John Conway

這是 John Conway (左圖), 創作了 *the game of life*。他將這些分數編碼並執行試除 (trial division) 演算法。他使用一種名爲 Fractran 的語言, 是由 Fortran 轉變而得。這不是非常有效率的試驗方式, 但這就是 Conway。

現在回到電腦可能幫得上忙的事情。先看著名的畢氏定理 $3^2+4^2=5^2$, $20^2+21^2=29^2$ 等等。費馬最後定理 (Fermat's Last Theorem) 斷言: 如果 n 至少是 3, 那麼兩個 n 次方的和不可能是某數的 n 次方 (這裡沒有任何數是零)。

費馬最後定理: *若* $n \geq 3$ *且* $xyz \neq 0 \Rightarrow x^n + y^n \neq z^n$.

費馬自認可以證明這個敘述, 但大家並不相信。另外有此一說: 費馬寫下那註記的書, 空白處太小; 但我不認爲如此, 畢竟沒有人找到過那本書。但費馬確實證明了: 兩個四次方的總和不可能等於另一數的四次方 $(x^4 + y^4 \neq z^4)$。

Euler之後猜測: 三個四次方的總和不可能另一數的是第四方:

Euler's sum of powers 猜測[4]: $x^4 + y^4 + z^4 \neq t^4$.

人們努力嘗試證明, 但未能成功。它之所以無法證明, 是因爲事實並非如此。這是第一個例子[5];

$$2,682,440^4 + 15,365,639^4 + 18,796,760^4 = 20,615,673^4.$$

Noam Elkies

這數值通常會讓你放緩探索速度, 雖未制止你, 卻讓多數人望而卻步。上例是 Noam Elkies 在 1986 年找到的。故事是這樣的 : 兩位數論學家 Noam Elkies (左圖) 和 Don Zagier 都在研究這個問題, 想了解是否有反例。Noam Elkies 也是位電腦高手。他們在電腦上查看數據。某回 Zagier 沒帶電腦去旅行, Elkies 在那段期間捷足先登。現在 Zagier 總是帶著筆電旅行, 不想再被搶先。

[4]編註: Euler 完整的猜測 : 對任意大於 2 的整數 n, $n-1$ 個 n 次方的總和不可能是另一數的 n 次方。

[5]編註: 1988年, Roger Frye 發現最小的反例 $95,800^4 + 217,519^4 + 414,560^4 = 422,481^4$.

另一方面, Andrew Wiles 在 1995 年證明了費馬最後定理。右圖是 1993 年的照片, 他笑了; 雖然數學家之後發現了一個 gap, 但在 95 年被搞定, 接下來呢? 誠如 Hilbert 所言, 一旦你處理一個難題, 經常會開發新的工具, 而且這些工具可以讓你超越界限; 譬如, 現在你可以破解 谷山-志村-韋伊 (Taniyama-Shimura-Weil) 猜想。無論如何, 絕對有人會走更遠。

Andrew Wiles

現在來談談 Andrew Beal 的最後猜想。

Andrew Beal 的猜想: *若* $x^a + y^b = z^c$, *其中* $a, b, c \geq 3$ *且* $x, y, z > 0$, *則* x, y, z *必有大於* 1 *的公因數。*

例如: 我們有 $3^3 + 6^3 = 3^5$, $85,683^5 + 2,197^7 = 1,485,172^4$; 這些例子看起來很容易, 其中的 x, y, z 都有比 1 大的公因數。Andrew Beal 是德州富有的銀行家, 也是業餘數論學家。他是如此富有, 提供了百萬美元獎金。我在數學學會的某委員會, 該會已收到幾百宗解決方案, 但你該如何正式判斷對錯? 總之沒人解決它。

Andrew Beal

上左是 Andrew Beal 的照片。事實上, 他是非常出色的德州撲克玩家, 照片右方那本書描述他在拉斯維加斯的種種; 他試圖擊敗最好的職業德州撲克牌手。如果你正在與專業的牌手玩牌, 彩池 (pot, 所有玩家在該局已下注籌碼的總和) 會變得很大。或許有你的五萬美元, 但你不能虛張聲勢, 因爲這金額對你來說非同小可。專業牌手自己的錢不很多, 而 Andrew Beal 是如此富有, 他的底池會大到讓他們緊張。那是他的計謀, 而且有一段時間奏效, 但之後他們串謀, 讓他輸了 3000 萬美元。但他說：這是值得的。無論如何, 那就是 Andrew Beal。


Ben Green 陶哲軒

等差數列 (arithmetic progression) 是一組間隔相等的數字, 如

$$199, 409, 619, 829, 1039, 1249, 1459, 1669, 1879, 2089,$$

實際上, 這十個數恰好都是質數。有一個老問題：質數可形成任意長度的等差數列嗎? 這問題歷經 200 年未解決, 2003 年 Ben Green (上圖左) 及 (驚人的) 陶哲軒 (Terence Tao, 上圖右) 證明: 事實上, 你可以得到它們; 他們開發了新的工具, 因此可以做更多事。

定理 (Ben Green, Terence Tao, 2003): *質數可形成任意長度的等差數列。*

還有個更爲艱鉅的挑戰：*你能找到相繼的質數所形成的任意長度之等差數列嗎? 介於兩質數間, 沒有其他質數。*舉例來說, 這裡有十個這種質數 (1998 由 Manfred Toplic 發現):

$$P + 210 \cdot k, \quad k = 1, 2, \ldots, 9,$$

其中 P 非常大, 我幾乎不能用一行寫完:

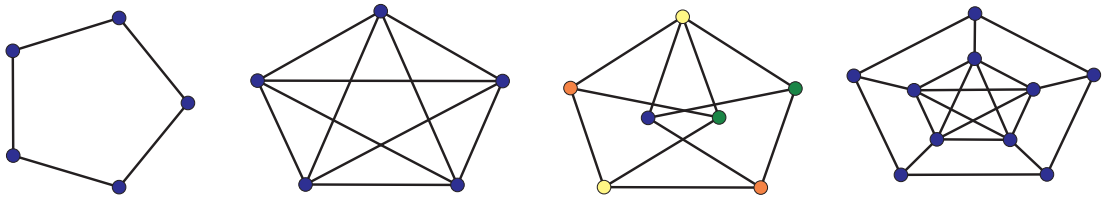
$$P = 100996972469714247637786655587969840329509324689190041803603417758904341703348882159067229719$$

我問過陶哲軒這個問題, 他說：「不, 這似乎是不可能的」。他們會找到 11 個相繼且等間隔的質數嗎? 一些人認爲, 無法用他們的方法找到, 但可能存在。好吧。

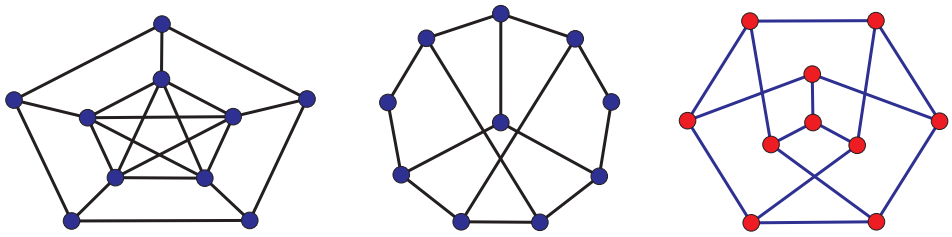
2. 圖論

來談談完全不同的東西, 看一些與電腦相關的組合學。圖 (graph, 通常我們只討論有限圖) $G = (V, E)$ 由頂點集合 V 和邊 (edges) 的集合 E 所組成, 其中所謂的邊是一對被配對

的頂點。如果 $(x, y) \in E$, 稱 x 與 y 相鄰 (adjacent), 記爲 $x \sim y$。下面是一些圖, 依序爲 5-cycle, 5 點的完全圖 (complete graph), Moser 圖, Petersen 圖。



這裡是 Petersen 圖的三個視圖。你如何繪製圖並不重要, 重要的是誰與誰有聯繫。如果我重新標記它並以不同的方式繪製它, 這三個圖形是一樣的:



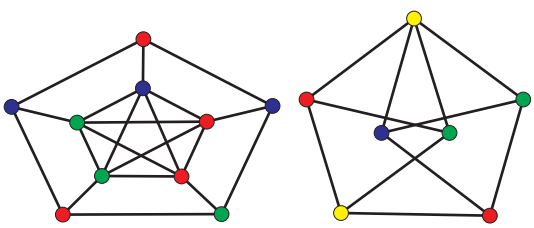
有些映射會一對一地映射頂點且保留所有邊。實際上, 這是一個未解決的問題: 兩個圖是否其實是同一個圖? 或者說, 它們同構嗎? 我們不知道如何回答。沒有多項式的演算法來判斷。

給定 r 個數, 分配其中某個數給各頂點, 亦即, 有個函數 $\lambda : V(G) \to \{1, \ldots, r\}$。譬如說, 有 r 種不同的顏色, 每個數代表一個顏色。如果相鄰的頂點不會有相同的顏色, 就是 r-著色 (r-coloring)。

定義: 稱 $\lambda : V(G) \to \{1, \ldots, r\}$ 爲 r-著色若其滿足 "$x \sim y \Rightarrow \lambda(x) \neq \lambda(y)$".

換言之, 當你分配頂點的顏色時, 如果有一個邊連接兩頂點, 則這兩頂點應該有不同的顏色。

定義: 著色數 (chromatic number) $\chi(G)$ 是: 爲圖形G以上述方式著色所需之最少顏色數量。

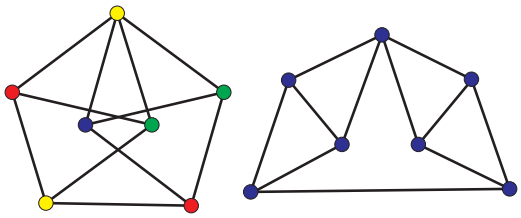


當然, 你可以給所有頂點互不相同的顏色, 這很容易; 但也許, 你想用較少量的顏色。例如, 我聲稱 Peterson 圖著色數是 3, 意即我可用 3 色而使相鄰的頂點不會有相同的顏色。然而, Moser 圖需要 4 色。你不可能用三色爲其頂點著色, 而不使相鄰頂點有相同的顏色。如果圖沒有任何

odd cycle (奇數頂點的迴路), 則著色數至多是 2。如果你拿個五角形, 嘗試使用兩種顏色: 紅色, 藍色, 紅色, 藍色、紅色, 哦, 哦, 行不通, 需要三種顏色。

事實: $\chi(G) \leq 2 \Leftrightarrow G$ 沒有 odd cycle。

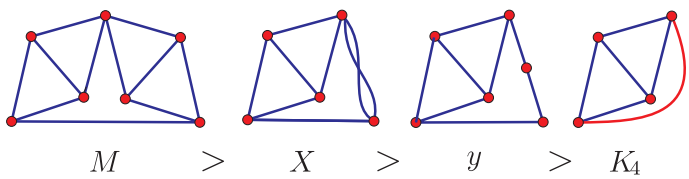
著色數是 3 呢? 這類問題計算上難以處理, 是 NP 難題。沒有好的演算法。如果我給你一個很大的圖, 你可以用三種顏色來著色嗎? 我不知道。



如果一個圖可繪製於平面上且邊不互相交叉, 則稱該圖爲平面圖 (planar graph)。左圖是 Moser 圖; 我只稍微伸展一下底邊, 就能像這樣畫一個平面圖; 好的, 這是一個平面圖。而 Petersen 圖不是平面圖; 如果你試圖在平面或球體表面繪製它, 無論你如何畫, 總有些邊會互相交叉。1853 年 Guthrie 提出

四色猜想 (Guthrie, 1853): 任意平面圖 G 滿足 $\chi(G) \leq 4$.

這猜想源自英國製圖人製作地圖所需, 他們希望確保相鄰的郡不會有相同的顏色, 否則很難區分各郡。他們發現用色從來沒有超過 4 種。曾有大約 12 年時間衆人以爲解決了猜想, 之後有人指出解中有錯。於是有一段時間此猜想懸而未決。Appel 與 Haken 在 1976 年宣稱 : 四色猜想是對的。他們需要做大量的計算, 處理數十億件案例, 使用最好的電腦。而當時人們並沒有意識到他們的兩篇文章破解了四色猜想。整個運算在大型主機進行 1500 小時。文章的第三作者 Koch 是一名程式設計師; 沒有他不可能進行這項工作。所以, 對這個證明來說, 電腦至關緊要。1997 年, Robertson、Sanders、Seymour 及 Thomas 宣稱這猜想仍然是對的。他們花了三個半小時在 Sparc-20 進行數十億筆計算。但迄今尚無任何凡人用紙筆得以完成的證明。



四色定理是 Hadwiger 猜想的特例, 那猜想講些什麼? 首先我們得先談談 graph minor。我們說 G 有一個 minor 爲 H, 記爲 $G > H$, 若且唯若你可以經由去除 G 的某些頂點、邊或縮小一些邊來得到 H。譬如看這 Moser 圖。我縮小一個邊, 於是有了新頂點, 我除去這一點, 接著用一個邊替換原來的兩個邊, 得到四點的完全圖 K_4。所以 K_4 是 M 的 minor。

事實上 graph minor 是圖論中一塊很大的領域。我們現在可以陳述:

Hadwiger 猜想 $H(k)$: $\chi(G) \geq k \Rightarrow G > K_k$.

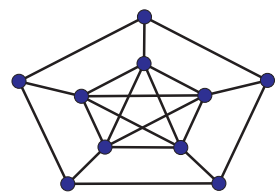
亦即, 若一個圖的著色數至少是 k, 則 k 點的完全圖必定是其 minor, 而這也就是導致這種情況發生的原因。事實上 $H(4)$ 相當於四色定理。Seymour、Robertson 及 Thomas 證明了 $H(6)$, 之後卡住了, 不能證明其他 $H(k)$; 如你我所知, 這意味著事情很困難。我希望有生之年看到對所有 k 的 $H(k)$ 證明, 但既然 Seymour 卡住了, 你必須等待有更多靈感的人。

接著談談電腦在這方面的趨勢。圖 G 中對頂點 x 與 y 定義

$$d_G(x,y) = x \text{ 與 } y \text{ 之間的距離, 亦即它們之間最短路徑的邊數。}$$

另一方面, 你可以定義 G 中的平均距離爲

$$\bar{d}(G) = \frac{1}{n^2} \sum_{x,y \in G} d_G(x,y).$$



我把各頂點間的距離全加給來後除以 n^2 (這是頂點的總數量), 得到圖的平均距離。例如, Peterson 圖的平均距離是1.5, $\bar{d}(P) = 1.5$。現在我給了你一個度量, 可測量兩點平均的接近程度。

還有另外一個圖形不變量名爲獨立數 (independence number)。圖 G 之某些頂點構成所謂的獨立集 S, 亦即 S 中之任二頂點都不相鄰; 我們稱最大獨立集的頂點數目爲獨立數 $\alpha(G)$。例如, 在 Peterson 圖, 有四個點沒有任何邊相連, 這四點是獨立的; 你找不到五個這樣的點, 但是有四個點如此, 因此 $\alpha(P) = 4$. 有一個很好的猜想:

猜想 (Graffiti, 1985): 對於每個連通的圖, $\alpha(G) \geq \bar{d}(G)$。

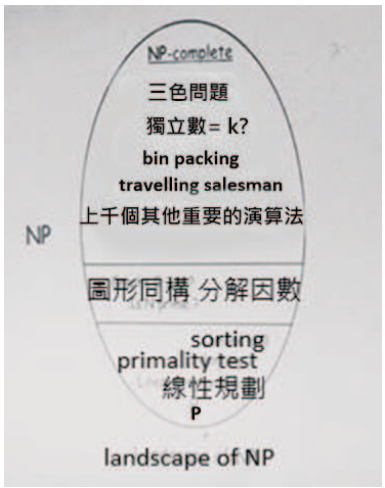
金芳蓉發現第一個證明, 非常複雜。

Graffiti 是個電腦程式, 已爲圖論產生大約六千個猜想, 由休士頓大學的 Siemion Fajtlowicz 研發, 著眼於圖的 50 至 75 個不變量, 譬如: 鄰接矩陣 (adjacency matrices) 的第二大之特徵值、著色數等, 並試圖找到彼此之間的關聯。他嘗試了很多類的圖, 譬如隨機圖 (random graphs) 等困難的圖, 進行所有測試, 並提出猜想。但大多數人不想處理那些猜想; 畢竟電腦還不是很擅長解決問題, 而這裡又生產大量問題。我們追得上嗎? 我不知道。

3. P, NP, NP-complete

接下來談複雜性 (complexity) 理論, 它涉及電腦及數學。P 的全名是 polynomial time, 表示可以在多項式時間找到解的問題。你想排序 (sort) N 個數字, 這有多難? 有個演算法用 $N \log N$ 個步驟能完成。

NP 的全名是 non-deterministic polynomial; NP 問題不確定有沒有多項式時間的解, 但是一旦提供一個答案, 就可以用多項式的時間「驗證答案」。如果有人聲稱有問題的解, 你至少可以檢查對錯。譬如, 我給你一個很大的數字, 說它是兩個質數的乘積; 如果要找這兩個質數, 沒有人知道好方法, 但如果我說 :「這是質數, 請檢查它是不是那乘積的質因數。」你就可以在多項式的時間檢查好。



最難的 NP 問題名爲 NP-complete。NP 問題都是任何 NP-complete 問題的特例, 而所有 NP-complete 問題在計算上都是等價的。這裡是一個地景, 往上看到 NP-complete 問題, 包括 : 三色問題, 圖的獨立數等於 k 的問題等上千個問題。底端有排序、線性規劃。還有個關乎質數的演算法[6], 落腳此處之前大家都不認識它; 當時有一位尚不知名的印度教授和他的兩名研究生, 發現多項式時間的演算法。它被發現後, 在網路上一天之內傳開, 聽聞者說:「哇, 這太神奇了, 我們怎麼錯過了?」至於判斷兩個圖 G, H 是否同構的演算法, 大家不知道它是在中間 NP 那塊或在下面 P 那塊。因數分解屬 NP, 你可以用之測試某數是否是爲質數, 或是找到未知的因數。

有一個紙箱包裝 (bin packing) 的問題。想像你有 100 個物體, 各具重量; 另有 10 個紙箱, 每個紙箱能承受 50 億的重量。你試圖把這 100 個物體放到 10 個紙箱裡, 但不能超過 500 億的重量總額。設這 100 個重量的總和是 5000 億, 那麼它們能成功裝箱嗎? 沒人知道。這個問題剛好超出範圍。我想很快會有演算法來解決這個問題, 但是我可以提出更大的難題。

我再講一次 : P 是我們在多項式時間內能解決的問題, 而最困難的是 NP-complete 問題。每個 NP-complete 問題都是其他 NP-complete 問題的特例。如果你找到一個多項式時間算法來解決三色問題、或者獨立數字 k 問題、或者業務員出差 (travelling salesman) 問題[7], 那麼你就有一個多項式時間的演算法來解決其他問題。但普遍的感覺是 : 不能如此, 這些 NP-complete 問題真的很難, 沒有多項式時間的演算法; 但不是所有人都這樣相信。

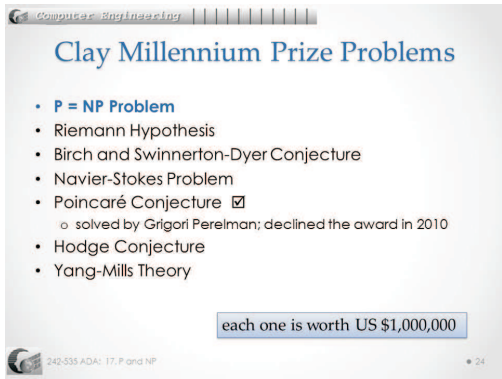
P=NP 嗎? 也許 NP-hard[8] 問題真的有多項式時間演算法, 只是我們還沒發現。這是一個大問題, 在懸賞一個百萬美元的問題名單中列名第一。我剛收到史坦佛大學 Don Knuth 的電子郵件, 他認爲答案是肯定的。雖然這可能是答案, 但多項式時間演算法是如此複雜, 你永遠

[6]編註: Manindra Agrawal 是 Indian Institute of Technology, Kapur 的電腦科學暨工程系教授。他和 Neeraj Kayal 及 Nitin Saxena 於 2002 年 8 月 6 日聯名發表 AKS 質數測試 (或稱 Agrawal-Kayal-Saxena 質數測試)。這個演算法可以在多項式時間判斷給定之整數是否爲質數。

[7]編註: 業務員出差問題 : 給定一系列城市和它們之間的距離, 要找路經每一座城市一次並回到起始城市的最短迴路。

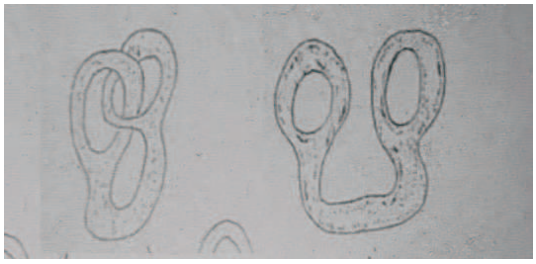
[8]編註: NP-Complete 問題滿足兩個條件 (1) 它是 NP 問題 (2) 所有的 NP 問題都是它的特例。NP-hard 問題滿足 (2) 但未必滿足 (1)。

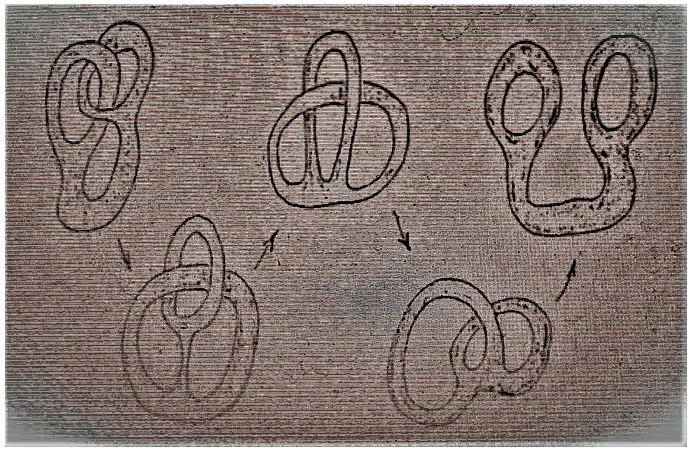
不能夠理解; 我的意思是, 我們只是凡人, 我們能做些什麼? N=NP又能如何?



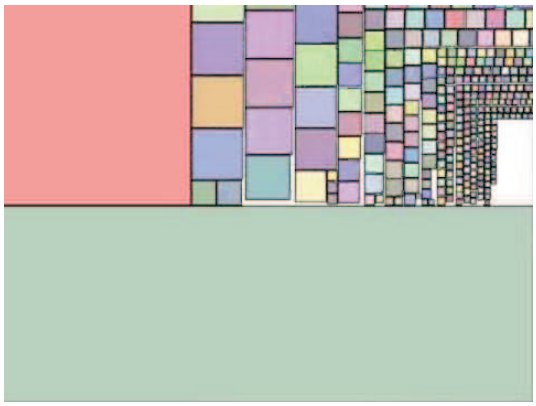
Clay 數學學院在西元 2000 年列出七個問題, 懸賞各一百萬美元。名單中列名第一的問題就是 P = NP。數學家說： 我們會解決這個問題, 電腦科學家說：不, 這是我們的問題。其實這是數學家和電腦科學家共同的問題。你可能看過這個問題: Poincaré 猜想, 已由 Grigori Perelman 解決。他非常不尋常, 高中時拿過國際數學奧林匹亞金牌獎。他獲頒一百萬美元, 但他說:「不, 我不要那筆錢, 我只想留在俄羅斯」。他和母親住在公寓裡, 拒絕領獎。所以 Clay 數學學院省下一筆錢, 現在還有六個一百萬美元的獎。畢竟這是他的選擇權。我們把它從列表中刪除, 現在表上還有六項。

現在來說個電腦比凡人占優勢的地方。這裡是一張圖片, 這是雙圓環, 想像它是充氣的橡皮雙圓環, 可伸展。我把玩它, 試圖解開結而不撕破它, 這可以做到。你認爲你能做到嗎? 你不這麼認爲, 但電腦說當然可以。你要怎麼做? 伸展它, 帶它到這個形式, 然後拉伸這個迴圈, 並把它塞進去, 然後盤繞在這裡。這些東西看上去好像糾結在一起, 但並非如此。這方面電腦占優勢, 它們沒有人類的直覺所構成的障礙。





現在有個幾何問題 square packing 給電腦做：「是否可以把邊長爲 1, 1/2, 1/3, 1/4, ..., 等越來越小的正方形, 都裝進 $1 \times \frac{\pi^2}{6}$ 的矩形?」爲什麼選擇 $\frac{\pi^2}{6}$? 這些正方形的面積總和是



$1+\frac{1}{4}+\frac{1}{9}+\frac{1}{16}+\cdots$, 而 Euler 說平方的倒數之總和 $\sum_{n=1}^{\infty}\frac{1}{n^2}=\frac{\pi^2}{6}$。所以, 這個矩形足以容納所有的正方形。你排列它們, 先把較大的正方形一個接著一個排進去; 你能完全排好嗎? 我不認爲。事實上, 電腦科學家 Clive Tooth 排進了前十萬個方形, 呈鋸齒狀, 能繼續不停地排。他用一個演算法, 可以證明它會一直繼續運作下去。那麼, 你可以一路跟著排嗎? 我付錢給任何恰當的證明或反證。你做不到的。我只是想知道這點。

4. 埃及的分數

來談談埃及的分數 (Egyptian fractions)。它們載於軍隊的萊因德數學紙草書 (Rhind papyrus)。公元前 1850年左右, 一些埃及人將分數寫成分母相異之單位分數的總和:

$$\frac{a}{b}=\frac{1}{n_1}+\frac{1}{n_2}+\cdots+\frac{1}{n_r},\qquad n_1<n_2<\cdots<n_r,$$
$$\frac{4}{7}=\frac{1}{2}+\frac{1}{14},\qquad \frac{3}{13}=\frac{1}{5}+\frac{1}{33}+\frac{1}{2145}.$$

我問過著名的數學家兼歷史學家 André Weil: 他們爲什麼要這樣做。他想了一下, 說道 : 這很容易解釋。他們誤入歧途, 他們錯了, 那是類似羅馬數字的錯誤。他們不應該這樣做, 但已經做了。Fibonacci (或名爲 Leonardo Pisano) 在 1202 年證明 : 你總是可以用貪婪演算法來達成任務。也就是說, 對 $\frac{a}{b}$, 取最大單位分數 $\frac{1}{\lceil b/a\rceil}$ 爲下一項, 後項的分子遲早會開始下降, 然後就搞定了。

現在做個微小的變化。假設你有一個奇數分母, 你想寫成奇數單位分數的總和,

$$\frac{a}{2b+1}=\frac{1}{2n_1+1}+\frac{1}{2n_2+1}+\cdots+\frac{1}{2n_r+1},\qquad n_1<n_2<\cdots<n_r,$$

貪婪算法會一直奏效嗎? 不知道, 沒有人知道它是否總奏效。例如, 用貪婪演算法分解 $\frac{5}{1444613}$, 會在第 37 個步驟完成, 最後一步的最大分母有 384 億位數。電腦得到它們, 但也耗盡力氣了 $\cdots$ 沒有一個不能完成分解的例子, 但大多數例子只是使數字變得更巨大, 情況似乎讓人無法樂觀。電腦好像在這裡筋疲力盡了。這是想告訴我們什麼? 我不知道。所有真正有趣的東西都在那裡發生, 但我們看不到。

我很久以前曾證明:

$$\frac{p}{q}=\frac{1}{n_1^2}+\frac{1}{n_2^2}+\cdots+\frac{1}{n_r^2},\ 1\le n_1<n_2<\cdots<n_r \Leftrightarrow \frac{p}{q}\in\left[0,\frac{\pi^2}{6}-1\right)\bigcup\left[1,\frac{\pi^2}{6}\right).$$

如果 $\frac{p}{q}$ 不大於 $\frac{\pi^2}{6}-1$, 你可以將 $\frac{p}{q}$ 寫爲完美平方的總和。如果你取 $n_1=1$, 則 $\frac{p}{q}$ 必須至少是 1。而 $\frac{\pi^2}{6}-1$ 大約是 .645, 與 1 之間有個間隙, 對間隙外的數你可以做分解。例如,

$$\frac{1}{2}=\frac{1}{2^2}+\frac{1}{3^2}+\frac{1}{4^2}+\frac{1}{5^2}+\frac{1}{6^2}+\frac{1}{15^2}+\frac{1}{18^2}+\frac{1}{36^2}+\frac{1}{60^2}+\frac{1}{180^2}.$$

貪婪算法總能完成分解? 不, 我相信它會逕自繼續做下去, 但非常低調。

5. 遞迴

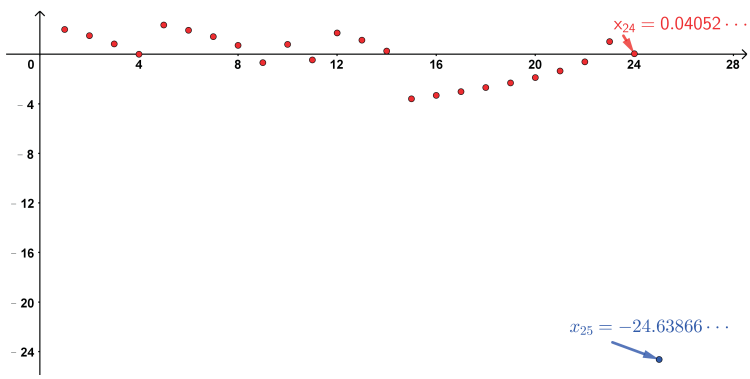
這裡又有個似乎完全沒有希望計算的遞迴, intractable recurrence。定義

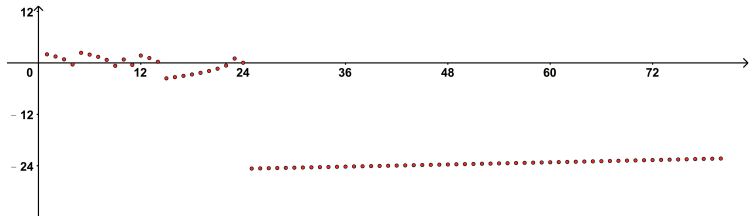
$$x_{n+1}=x_n-\frac{1}{x_n},\qquad x_1=2,$$

則有

n	1	2	3	4	5	6
x_n	2	3/2	5/6	-11/30	779/330	497941/257070

繼續做下去, 發現 x_{24} 真的很小, 是 x_{23} 減去比自身小的數。x_{24} 非常接近零。當 x_{24} 減去很小的數的倒數, 突然之間 x_{25} 數值變得非常大。之後, 數值呈現緩慢的變化, 問題是：這序列是無界的嗎? 如果你讓它繼續進行, 會不會出現任意大的值, 還是會滯留某處? 計算對此完全沒用; 任何小錯誤都會被放大, 無法證明任何事, 沒人知道答案。我們僅知, 在任何開區間, 有些地方有界, 有些地方無界。但是, 進一步的事, 沒有人知道; 我們沒有線索, 計算在此幫不上忙。





接著來看一個未公開的排序演算法。再次, 一旦你知道 x_n, 我會告訴你如何得到 x_{n+1}:

$$x_{n+1} = \left[\sqrt{2x_n(x_n+1)}\right], \quad x_1 = 1;$$

你把 x_n 乘上 $x_n + 1$, 再加倍, 取平方根並捨去小數, 如是, 你得到

n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
x_n	1	2	3	4	6	9	13	19	27	38	54	77	109	154	218
$x_{2n+1} \pmod 2$			1		0		1		1		0		1		0

你如何處理這些出現的數字, 用排序演算法進行分析? 你能談一下這個序列嗎? 它的表現良好嗎? 你再進行另一個運算, mod 2, 於是你得到

$$1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 1, 1, \ldots$$

你怎麼看這個序列? 它最終會有周期性嗎? 事實上, 把小數點放進來, 這是 $\sqrt{2}$ 的二進制展開。

$$1.01101000001001111\cdots \approx \sqrt{2} \quad \text{二進位制 (in binary)}$$

一旦你知道這件事, 就不那麼難證明。這些數字看起來有點隨機, 但它不是隨機的。它至少給了你定義明確的序列。

現在, 你把3放進來, 做同樣的事情, 令

$$y_{n+1} = \left[\sqrt[3]{3y_n(y_n+1)}\right], \quad y_1 = 1,$$

並類似地運作 mod 3, 得到

n	1	2	3	4	5	6	7	8	9	10	11	12	13
y_n	1	2	4	7	12	21	37	64	111	193	335	581	1007
$y_{2n+1} \pmod 3$			1		0		1		0		2		2

你看這個數字

$$1.01022200110\cdots(\text{基底 } 3),$$

這是什麼呢? 有些程式曾探尋過這可能是什麼, 但它們找不到任何東西。肯定地說, 這什麼也不是, 只是一堆數字, 就是這樣; 很難識別一個實數, 即使要識別零也很難。

6. 超基展開 (super-base expansion), Goodstein's Theorem

好吧, 來看一個很大的數字。你可以把 41 寫成 2 的冪次方的和,

$$41 = 2^5 + 2^3 + 1.$$

但是我不想就此停下。我想要把指數也寫成 2 的冪次方的和。

$$41 = 2^5 + 2^3 + 1 = 2^{2^2+1} + 2^{2+1} + 1.$$

於是, 出現的數字都是 2 的冪次方的和。這被稱爲超基-2 展開 (super-base-2 expansion)。這是第一步。現在以 3 取代 2, 減去 1, 寫下超基-3 展開 (super-base-3 expansion)。這是什麼意思? 我把 2 一個一個地替換成 3, 然後減去 1; 這是超基-3:

$$41 \to 3^{3^3+1} + 3^{3+1} + 1 - 1$$

全都是 3 或 3 的冪次方, 係數可能是 1 或 2。我們有

$$41 \to 3^{3^3+1} + 3^{3+1} + 1 - 1 = 22,876,792,455,042,$$

這是很大的數字啊! 下一步, 用 4 替換 3, 減去 1, 確保它是超基-4 展開 (super base-4 expansion)。如果你用 4 來代替所有的 3, 得到的還不是超基-4 展開; 你要再減 1, 才把它轉換成超基-4。這是超基-4 展開:

$$\begin{aligned} 3^{3^3+1} + 3^3 + 1 &\to 4^{4^4+1} + 4^{4+1} + 1 - 1 \\ &= 4^{4^4+1} + 3 \cdot 4^4 + 3 \cdot 4^3 + 3 \cdot 4^2 + 3 \cdot 4 + 3 \end{aligned}$$

一些係數是 3; 我們有這個數字:

$4^{4^4+1} + 4^{4+1} + 1 - 1$

$= 5363123171977038839829609999282338450991746328236957351089424577488705612029418790720$
$749719266761371076012743274594420341501553124778627978573459602433740 7$

一般的步驟是, 無論你以超基-b 展開成什麼, 用 $b+1$ 代替它, 之後減去 1, 就確保你得到了超基-$(b+1)$ 展開 (super-base-$(b+1)$ expansion)。這個數字啊,

$$5^{5^{5+1}}+3\cdot 5^5+3\cdot 5^3+3\cdot 5^2+3\cdot 5+2=$$

$$95550629897273876017820227985198229959904052449504716856975639462326026512130790150602969325986992513279322007789723111767960639433690348614420507345799330104398094837859785091964083016902380561298776681305050074132556170657388412620574654722358848264137814259836875719767877123954660960332094150589358456127621053502535453233719143542572497512829309723077159175568992456684588996406371692021577461842776339179818705105266577301567686266287431845457988934516413322959149190761514346828643684571132406564587188106816286516082264148974343128812268110900883661247028382140968003936035691853617765272317807697320059267424689635975729725275411637461080292445645547259497997434309977157383346900651858808179629723987308211002544253973490224356660256658036956711527009943628501916490062302509850673369858795451369474696190865789349842294989739053402141121804689197316763271140785215141622119275754115824548364285608585406161639524090863416375505637339115870549294434185426100035586674612695666115037807359021450376383889667615310030914300622762712153050344740272329235241032549133215968048019436812925537353717031814348828835134962932497677898815908695127544566561164737196517197808066416703641583174912907261343100215389954234405190209368416240045193679810645981680129156039083683687126666143964845360274529781070344441299562229092118979893173824215783688046181254518575589947071213113511003314324343393435509149043640128034655097464041541252209921239839602945440855616359615072779145837339759871527401320232342700136699693039929723298075087629348290572378425502078434386545185624126767191964269879937472924852501911250624464200091329502812564309381496902220367007117353102789265266251745909479485359965283109425648159375087176798014110051910580802427256051965665612816613038321811834414842510441974807141524236955699583481132497428184261735643664739834044225470294697555232547206895475113827282656650933531676066151423025971719069990528070032629765036589538635553289174708732134236047806732663874292119137449834377526252197109116095678611527033357686687124271822831891022850827296609077026774196807125332249292701653733234270945074067173857325157518977087889311405888292938470840454102546 7

這就是當你把所有的 4 換成 5 得到的數字。數字越來越大, 對嗎? 但難以置信的 Goodstein 的定理說：如果你重複這個操作, 最終會歸零。

Goodstein 的定理: 對每一個整數 n, 如果我們套用前述過程, 由 n 的超基 2 展開起步, 最終必定會到達 0。

這個敘述一開始看起來很有希望是對的, 歸零的速度確實非常快。這是來自邏輯學的定理。邏

輯學家看它的方式是： 讓各個出現的基底被取代爲第一無窮序數 (the first infinite ordinal number) ω, 當你減去 1, 你會得到更小的序數, 繼而依序得到嚴格遞減的序數鏈。無論如何, 令 $G(n)$ 是 n 降爲零所需的步驟數, 則 $G(3) = 6$： 如果你從 3 開始,

$$3 = 2^1 + 1 \to 3^1 + 1 - 1 \to 4^1 - 1 = 3 \to 3 - 1 = 2 \to 2 - 1 \to 1 - 1 = 0.$$

它會下降。好的, $G(4)$ 是什麼? 從 $4 = 2^2$ 起步, 第二步是 $3^3 - 1 = 2 \cdot 3^2 + 2 \cdot 3^1 + 3 - 1 \cdots, \cdots$, 你可能會說, 既然這是電腦演算法, 就讓電腦跑, 吃完午餐再回來檢查。$G(4)$ 可是相當大的。

$$G(4) = 3 \cdot 2^{27}(2^{3 \cdot 2^{27}} - 1) + 4 > 10^{1,000}.$$

它大於 10^{1000}, 需要做 10 萬步以上。金芳蓉的學生 Paul Horn 計算了 $G(5)$, 我需要付他 25 美元。沒有人計算過 $G(6)$, 它超出計算範圍了, 你根本無法計算。

7. 黎曼猜想

接下來, 讓人最想談的問題就是所謂的黎曼猜想 (Riemann hypothesis)。我們考慮 Riemann-Zeta 函數

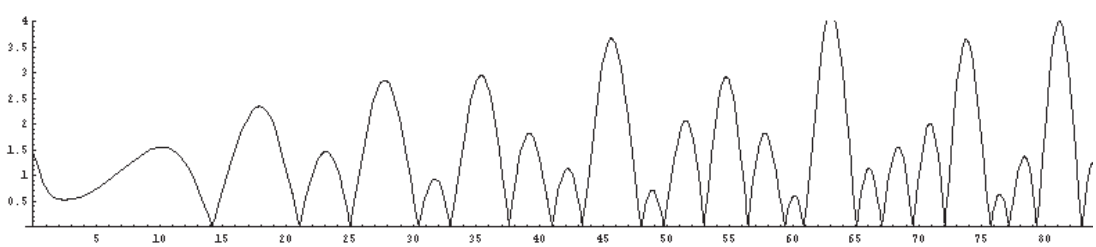
$$\zeta(s) = \sum_{n=1}^{\infty} \frac{1}{n^s}, \qquad s = \sigma + i\tau.$$

$$\zeta(2) = \sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{\pi^2}{6} = 1.6449 \cdots$$

我們再次看到整數平方的倒數總和約爲 1,644。n 是整數, s 是複數冪, 例如,

$$\zeta(2 + i) = 1.15036 - 0.43753i$$
$$\zeta(\frac{1}{2} + 14.134725 \cdots i) = 0$$

而 Riemann-Zeta 函數的零點與質數的性質密切相關, 因此人們很想了解它。這裡有一張圖片, 如果你取實部爲 1/2, 另有一個虛部, 你可以看到函數表現得很好。



$|\zeta(1/2 + bI)|$ for $b = 0$ to 85.

黎曼猜想: 所有非平凡零點的實部都爲 $1/2$。

前十萬億個零點確實如此, 所以看來很不錯。這也是一百萬美元的問題, 迄今沒人知道答案。

我來提最後一個問題。調合數 $H(n)$ 是前 n 個整數之倒數的總和。

$$H(n) = \sum_{k=1}^{n} \frac{1}{k} = 1 + \frac{1}{2} + \cdots + \frac{1}{n}.$$

我們已有大量關於調合數的分析及演算法。數論中另考慮 n 的因數總和,

$$\sum_{d|n} d = n \text{ 的因數總和。}$$

(例如, 如果 $n = 6$, 則其因子有 1, 2, 3 和 6)。另有一個問題:

$$\sum_{d|n} d \le H(n) + e^{H(n)} \log H(n) \qquad \forall\ n \ge 1? \tag{1}$$

是或否? 衆人爲何對這問題感興趣? 因爲它等價於於黎曼猜想 (J. Lagarias, 2000 及 G. Robin, 1984)。

如果你發現 (1) 的反例, 則將會有無限多個零點在臨界線 $1/2 + it$ 外。這與 Hilbert 的觀點一致; 亦即, 看到一個不認識的問題時, 你會問說 : 這很容易嗎? 還是很難? 你往往無法判斷; 很多困難的問題有等價的形式, 但並不明顯。當 $n < 10^{100}$ 時上述問題 (1) 是對的, 因此大多數人認爲黎曼猜想是對的, 但不是每個人都做如是想。我的前同事 Odlyzko 已經計算出第 10^{22} 個零點和其兩側 10^9 個鄰居 (因爲有更強的猜測, 不僅關乎零點的位置, 也關乎間隔的位置)。他的結論是 :「如果這是對的, 只是勉強是對的! 它幾乎失敗, 但並不完全失敗」。我的一位前同事 Richard Hamming 說:

「計算的目的不是數字, 而是洞察力 (insight, not numbers)」。

但有人指出, 我們可以用這種語法寫一個變體:

「計算的目的還看不到 (not yet in sight)」。

啊! 我們會知道黎曼猜想是否屬實嗎? 我贊同 Hilbert 的哲學「We must know, we will know」。他是德國人, 他當然不是真的這麼說。他說的是「Wir mussen wissen, wir warden wissen.」事實上, 他說過 :Wir durfen nicht denen glauben die heute mit philosophischer miene und uberlegenem Tone den Kulturuntergang prophezeien und sich in dem ignorabimus gefallen. Fur uns gibt es kein Ignorabimus und meiner Meinung nacht auch fur auch die Naturwissenschaft uberhaupt richt. Statt des$\cdots$. (我們不應該相信那些今天以哲學的陳述和優越的口吻預言文化的消亡, 而陷於無知 (Ignorabimus) 的人。對我們來說, 根本不存在無知, 甚至在科學上也是如此。相反地, 我們必須知道, 我們將會知道。).

他確實說過, 在 1930 年。我認爲這個演講振奮人心。問題在於 : 我們必須知道, 我們將會知道 ··· 是我們還是未來的一代? 事實上, 這就是爲什麼我們有這麼多聰明的學生, 可以解決我們解決不了的問題。希望我的演講讓你感受到電腦與數學的互動。謝謝你來聽演講。

8. 量子電腦

問: 質因數分解有多項式時間的演算法嗎?

答: 是的, 我認爲最好的演算法會像 n^6。演算法的選擇, 取決於所謂的廣義黎曼猜想 (generalized Riemann hypothesis)。如果我們相信廣義黎曼猜想是對的, 那麼演算法的速度會更快; 以這種方式嘗試似乎總能奏效, 但我們無法證明它總行得通。如果你想實際找到未爲人知的質因子, 這是多項式時間的演算法。事實上, 如果 P 不等於 NP, P 和 NP 確爲互相獨立的類別, 那麼兩者之間必然存在著一些問題; 介於兩者之間的不是多項式的演算法, 但也不像最難的 NP 那麼難; 人們懷疑因數分解或圖形同構問題可能介於兩者之間。

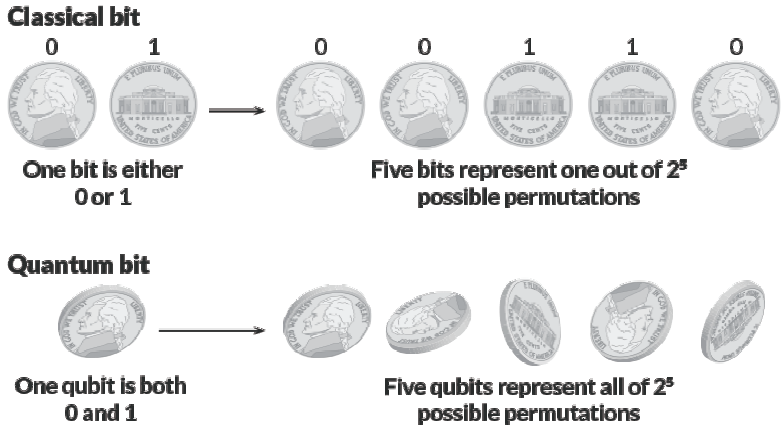
你可能聽說過量子電腦, 我們尚未建立它, 但沒有做不成的理由。我在貝爾實驗室 (現任職於麻省理工學院) 的同事 Peter Shor 證明 : 如能建立量子電腦, 則可於多項式時間分解因數。那麼 NP-hard 問題呢? 沒有人真正發現過 NP-hard 問題, 而量子電腦能以指數級加速。即使有了量子電腦, 也許它能做一些事, 但是它不會如人們曾認爲的那樣強大。而譬如能源融合, 這是工程問題, 建起來只需 50 或 100 年。理論上沒有建不成量子電腦的理由。起初人們認爲 : 不可能製造它, 因爲它太微妙了, 但目前的量子計算已可容錯, 能夠更正演算法中的錯誤。物理學家非常聰明, 正在建立量子電腦, 想要纏結 (entangle) 量子位元 (qubit); 他們持續進步, 但仍需要時間。

問: 可以介紹一下量子電腦嗎?

答: 可以。數位電腦能模擬任何可落實的計算設備, 並且能以有效的方式進行。任何實體的作業, 都可以用數位電腦執行。

費曼指出, 某些量子力學系統, 似乎不可能在標準 von Neumann 計算機上有效率地模擬, 他建議盡可能利用量子力學的行爲本質, 用量子力學本身特質來破解這些問題。我認爲對量子資訊理論來說, 姚期智是的極具開創性的人物。藉由 Benioff、Deutsch、Bennett 和姚期智等人的工作, 奠基於量子力學的計算模型開始發展。94 年有重大突破, Peter Shor 發現了一個演算法, 如果能有一台量子計算機來執行它, 就可用多項式時間 ($c(\log N)^2$ 步驟, c 爲常數) 分解因數。在數位電腦上因數分解的演算法最快以 $\exp(c(\log N)^{1/3}(\log\log N)^{2/3})$, ($c$ 爲常數), 運行。他也爲離散對數 (discrete logarithm) 發現相關的演算法。

經典的 n 位寄存器 (register) 和量子電腦裡的 n 量子位元寄存器有完全不同的組合。[9] 每一位元都在 0 和 1 系統; 但在量子力學, 有一些複 (complex) 線性組合 superposition, 不能孤立任何東西, 它們在 2^n 維的複 Hilbert 空間, 以張量積描述, 極其龐大, 在數學上一團混亂; 在大多數階段, 不能以個別量子位元的內積描述; 它們纏結在一起 (entangled)。譬如, 如果你把光子纏結在一起, 之後讓它們相距數公里, 一旦你測量其中之一, 會發生順時針或逆時針偏振, 因此另一個光子會立即到達它應該去的地方, 看似它們彼此相關。這是物理上的大串謀, 能操縱這些纏結態將賦予你威力。麻煩的是, 如果做一丁點干擾, 它們就可能會塌縮。



消退 (decoherence) 意味著：量子系統暴露於外在影響, 因而跳轉成固定狀態, 並從量子狀態退出。事實上, 你有量子糾錯碼 (quantum-error-correcting codes), 可以實際糾正這些事件及所有這些算法的小錯誤。所以, 如果能夠很大程度地仰賴物理, 且有容錯演算法, 而你可以學習容錯演算法, 那麼你就可以建立整個理論。

我相信量子電腦問世的時間會比你想的早。另一方面, Hofstadter 的定律說： 所有任務花費的時間總會比你所預期的長, 即使在你的預期中考慮了 Hofstadter 的定律。這是遞迴的自我引用 (recursive self-referential) 的論證, 所以, 誰知道? 你看看吧。

我用經典的 Erdös 猜想來結束演講, 我問過陶哲軒： 他們是否能解這問題, 他說不行, 他們無法解決它。這裡有一組整數, 其倒數的總和發散, 例如質數, 這組整數是否必包含任意長的等差數列? 特別是,

$$S \subset \mathbb{N}, \sum_{s\in S} \frac{1}{s} = \infty \overset{?}{\Rightarrow} \exists\, x, d \text{ 使得 } \{x, x+d, x+2d\} \subset S.$$

3,000 多美元已經懸賞了近 70 年, 有些人正試圖證明這種三項等差數列存在, 但仍做不到。

[9]編註: 下圖複製自https://www.sciencenews.org/article/quarter-century-ago-qubit-was-born.