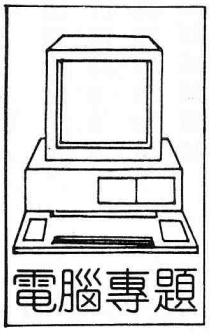

未來數學家的挑戰——

計算量問題

楊照崑　楊重駿

一、前言

有數學家說過「一個好的問題勝過十個好解答」。因爲解答一出，此問題已是到了終點，對不斷求創新的人們而言，已不構成挑戰。而新的問題是源頭活水，能開拓新的境界。多數人都不願沉醉在好的解答中不斷的玩味，而希望找到新的問題，不斷的思考，摸索。

大家在「數播」上已看見了不少好的問題，尤其最近康明昌教授談到的費馬定理，幾何三大難題，都是極有趣的問題。有的已有了解答，有的尚待解決。除了上面的題目外，像四色問題（即任何一個地圖只要用四種顏色就可以把國界分開），五次以上方程式的公式解，及數論上質數分佈問題，都曾在職業及業餘數學家的心目中佔有相當的地位。本文所要介紹的是一個最近（1970年代開始）一種許多數學家及電子計算機學家所關心的大問題——NP問題。NP所代表的意思，你看完本文之後自然會明白，現在你不妨記住“NP-hard”這個偉大的字。將來如果你對某人說你的問題是“NP-hard”，他也許就要對你刮目相看了，NP-hard不但表示 hard（難），而且是NP的難！

NP問題的代表問題之一是售貨員旅行問題（traveling salesman problem）。有一個售貨員要開汽車到 n 個指定的城市去推銷貨物，他必須經過全部的 n 個城。現在他有一個有此 n 城的地圖及各城之間的公路距離，試問他應如何取最短的行程從家中出發再回到家中？

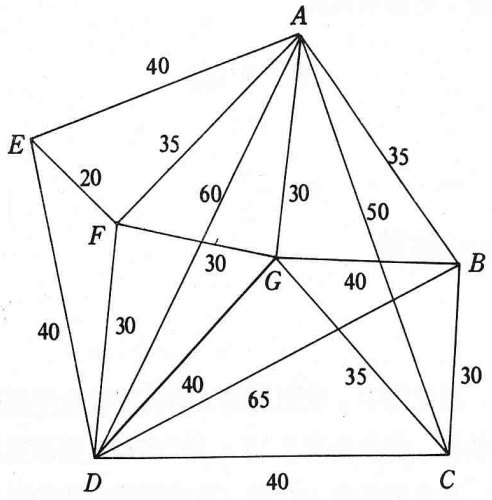

圖1　售貨員之地圖，A，B，C，…表城名，數字表兩城之間之里數。

如圖1中，A，B，C，…，G表示7個城市，而售貨員要從A城出發再回到A城並訪問B，C，…G，所有的城，一個可行的方法是

$$A\to B\to C\to G\to D\to E\to F\to A$$

問題是：這是否是最短的途徑？也許

$$A\to G\to B\to C\to D\to F\to E\to A$$

更近呢？加起來的結果第一路徑總長235里，而第二路徑總長爲230里，故第二路徑較短，但是否存在一個更短的路徑呢？目前的方法接近一個一個的排著試，還沒有找到更好可以尋得最短路徑的方法。對七個城而言，共有6！＝720個排法，尚不算難，但若有20個城，則排法就有19！種。因

$$n!\simeq\sqrt{2\pi n}\,(n/e)^n$$

故

$$19!\simeq 1.21\times 10^{17}$$

在排列組合裡$n!$寫起來輕鬆，但1.21×10^{17}是一個大得不得了的數字，若每秒鐘排一次，要排3.84×10^9年（一年約爲3.15×10^7秒），即使使用計算機，每秒排一百萬次（不容易做到）也得重做三千年才能找到答案。「生也有涯，知也無涯」，想不到區區二十個城，要三十個世紀才能找到答案。

由於電子計算機的發展，有許多以前認爲枉費時的計算，像行列式之值，反矩陣，高次方程式的解，都可以在極短的時間內解決。但也突然出現了一些新問題，連大型計算機也望之興嘆。像售貨員問題，因爲找不到比硬排好得很多的做法，使得數學家們開始想要證明，根本找不到比硬排好得很多的做法。這個證明至今尚未找到。就像以前一角三等分問題一樣，既然找了幾千年找不到用圓規直尺三等分任一角的方法，也許我們可以證明絕對不可能用圓規直尺三等分一角。現在我們要證明絕不可能寫一個計算機程式大大的簡化售貨員旅行問題。與三等分一角問題不同的是，前者是一種數學上的好奇，而當今的問題與實際用途却有密切的關連。

在此我們一直強調一個好得很多的方法。原因是對這類的問題，你若能計算快一倍或十倍，千倍，往往起不了什麼大作用，好像剛才的二十城旅行問題，即使快了千倍，仍需三年的計算時間，而再加三城立刻就把這個計算法的效果抵消了，因此我們所要的是計算量基本層次的減少，這就是我們在下一節所要討論的。

二、計算量

計算量，顧名思意，是指解決某問題所需要計算的時間，但因每個複雜問題的計算往往都要經過許多不同的運算，除加減乘除四則外，還要包含比較，取數據，存數據等等，若仔細計算起來，十分困難，一般都只繪出一兩個主要的量，加以統計，以上節中售貨員旅行問題爲例，其主要的工作是對每一個排法加起總路徑之長，因對n城而言，有$(n-1)!$的排法，我們就定其計算量爲$O(n!)$，即在$n!$之層次（order即O縮寫之來源）之內。

舉二個例子，我們若要求n個數的和或平均值，則其計算量爲$O(n)$。但若我們要把n個數字依次排列，則其計算量會因做法的不同而有相當的差別，一個直接了當的方法是，先求出最大的（比$(n-1)$次），再從不是最大的中間求次大的（比$(n-2)$次），再求第三大的（比$(n-3)$次），…如此一共比了

$$(n-1)+(n-2)+\cdots+1=\frac{n(n-1)}{2}$$

次就可以完成此工作。因此我們以$O(n^2)$，即在n^2之層次來表此方法的計算量。另外一種快排法，先把n個數分成若干小塊，每塊排好之後再合起來，則可以證明此種方法之計算

表一 以計算機每秒做一百萬次時完成各層次計算量所約需的時間（若無單位，均以秒爲單位）

n	$\log n$	n	$n\log n$	n^2	n^3	2^n	3^n	$n!$
10	10^{-6}	10^{-5}	10^{-5}	10^{-4}	10^{-3}	10^{-3}	0.059	0.45
20	10^{-6}	10^{-5}	10^{-5}	10^{-4}	10^{-2}	1（秒）	58（分）	1年
50	10^{-5}	10^{-4}	10^{-4}	0.0025	0.125	36年	2×10^{10}年	10^{57}年
1000	10^{-5}	10^{-3}	10^{-3}	1	16小時	10^{333} 年	極 大	
10^6	10^{-5}	1	6	1月	10^5 年	極 大		
10^9	10^{-5}	16小時	6天	3年	3×10^9年	極 大		

量爲$O(n\log_2 n)$[1] 因排數字與排名字，電話號碼相同，這種排法很有實用價值，例如某大城有一百萬戶，則$n^2=10^{12}$，而$n\log_2 n$ 只有2×10^7，其差別三個月與一分鐘之比。

一般計算量的層次多以下表來區分，

$$O(\log n)<O(n)<O(n\log n)<O(n^2)<O(n^k)<O(2^n)<O(k^n)<O(n)$$

在上表中，k爲某一大於2的正整數，它們中間都有一道鴻溝，有基本層之不同，在計算機理論上，若某人能發現一個新的方法，降低一個層次的計算量，那麼他的新方法有資格稱之爲一個突破，可以不朽矣。表1有一個對上項各量之比較，是以計算機每秒作一百萬次(10^6)計算爲原則。

在這個表中，特別注意n^3與2^n中之差異，一般稱2^n爲計算量呈指數上升，而n^3或n^k之計算量呈n的方次上升[2]，對目前及未來的計算機而言，一個呈方次上升的計算量應可以應付，但對一個呈指數上升的計算量在n相當大時則毫無希望。因此計算機學家所集中精力的方向在如何將一個呈指數上升的計算量問題，簡化成一個方次上升的計算量問題。我們對定義凡對一個問題中最重要的參數n而言，若能找到一個方法可以以方次上升的計算量完成，我們稱此問題爲一P－問題（P爲英文多項式Polynomial之第一字母），包含所有此問題之集合以P表示之。

三、P之外？

本節之題目有點不平常，我們的目的是提醒讀者本文中常用之英文大寫的P是一個凡能用$O(n^k)$計算量解決之問題之集合。而P之外加一個問號係指到目前爲止，我們尚不知道P之外是否是一個空集合。

到目前爲止，除了售貨員旅行問題之外，已經有上百有趣或有用的問題，無法用$O(n^k)$的計算量來解決，我們在此列舉幾個例子。

問題1：售貨員旅行問題(甲)，即第一節所述之問題，不再重複，不過假定所有距離均爲正整數[3]。

問題2 售貨員旅行問題(乙)，與第一題之條件相同，但現在有一個給定之正整數B，問題是是否存在一條路徑其總距離不大於B。（問題1與問題2在表面上相似，但在以後的理論上有很大的不同）

問題3 背袋問題(甲)，有物體n個，各重$w_1, w_2, \cdots, w_n$，今欲將它們分爲二袋，試問如何分法可使兩袋之重量最爲接近。（不妨

假定 w_i 皆爲正整數，這並未失去一般性。）

問題4：背袋問題(乙)，如上題，並給定一正整數 B，試問可否選出若干 w_i，使其和

$$S \leq \sum_{i=1}^{n} w_i / 2 \quad \text{且} \quad S \geq B$$

問題5：包裝問題：有 n 個各別重量小於 1 公斤的物品及足夠可以裝 1 公斤東西的盒子，今將物品裝於盒子之中，多個物品可裝於一盒，但任何一盒不得重於 1 公斤，試求最小的盒子數。

問題6：舞伴問題：今有 n 個男孩子與 n 個女孩子參加舞會，每個男孩與女孩均交給主持一個名單，寫上他（她）中意的舞伴（至少一人，但可以多於一人）。試問主持人在收到名單後，是否可以分成 n 對，使每人均得到他（她）所喜歡的舞伴。

問題7：庫存問題：某倉庫有 D 個存倉，排成一列，今有 n 批貨物，各可佔有一個或多個存倉，並已知各批物品存入與提出之日期。試問可否將各貨物存入庫裏不發生存倉不夠的困難且同一批貨物若需一個以上存倉時，其存倉必須相鄰。

問題8：已知 a，b，n 三正整數，問是否存在一小於 n 位之正整數使得

$$x^2 \equiv a \pmod{b}$$

問題9：(甲)：給定一 n 位正整數 a，試問其是否爲質數？

(乙)：給定一 n 位正整數 a，試問是否存在 m，$n > 1$ 且 $a = mn$？

問題10：分叢問題：已知空間 n 個點，並假定各點之間之距離爲正整數，又給定兩正整數 K 與 B，問是否可將此 n 點分成小於 K 個不重合的子集，使得在同一子集內之任意二點距離均不大於 B？

現在可以看出這類問題的一般結構了。很顯然的，有些是極有用的問題，而有些可以轉換成有用的問題。例如舞伴問題，若把男孩與女孩換成工人與工頭，或醫生與病人就有大用了。這些問題到目前沒有一個可以證明是屬於 P 的，大家都猜測它們可能在 P 之外，即其計算量是呈指數增加的。

在 60 年代，已有些人把某些問題歸於一類了，即是幾個問題是互依的，若其中之一若屬於 P，則其他幾個也屬於 P，其證明方法大都是證明兩個互依問題中間有一個只需要用 $O(n^k)$ 時間來完成的橋樑。直到 1971 年古克（Stephen A. Cook）發表了“The Complexity of Theorem Proving Procedures”才把 P 之外約問題歸成了三大類，即 NP，NP-complete 及 NP-hard[4]，現在談古克定律。

四、古克定律與NP-completeness

古克定律的證明很難，就是瞭解它也不容易，我們將從幾個角度來看這個問題，試著去瞭解它。它的主要結果是把前節那類問題大部歸於一個較易證明的集合，稱之爲 NP，而在 NP 中找到一批互依的問題稱之爲 NP-complete 類並得到下面的結果。

1. 若有一個 NP-complete 問題可以用 $O(n^k)$ 計算量來解決，則全體的 NP 問題都可以用 $O(n^k)$ 之計算量來解決，即

1′ 若有一個 $x \in$ NP-complete 且 $x \in P$，則 $P =$ NP。

又換句話說，NP-complete 是 NP 中的難題，NP-complete 解決了[5]，NP 就解決

了。但若有一個屬於 NP 而不屬於NP-complete 的問題解決了，則其他的 NP 問題不一定可以解決。

什麼叫做 NP ？NP 是英文 nondeterministic polynomial 的縮寫，意思就是非確定性的多項式時間。要瞭解這個字。我們先看一看普通計算機的作用。

現在已知用一個計算機，要解決售貨員旅行問題非常困難，但若我們有許多計算機同時用，是否可以快到把原問題在$O(n^k)$時間內解決？“許多”，不是一、二，多一二個是於事無補的，多百個千個仍是杯水車薪，不能有很大的作用，因爲就是一千個機子可以分開做，也最多只能快一千倍，在第一節內已說過，幫助不大。因此計算機學家先放眼望去，乾脆允許你可以無限的增加機器。現在我們要注意的是並不是有了無限多的機器所有的問題就可以立刻解決了，因有的問題有先後次序，例如在算下式的時候

$$〔(a_1+a_2)a_3+a_4〕a_5+a_6$$

除非換個形式，否則必須一步一步的解括弧，機子多了並不能加快計算的速度，而且機子多了，其間之聯絡千變萬化，一個機子要應付千千萬萬別的機子送來的信號也疲於奔命了。因此我們只假定所有的機子都只承上啓下，單線作業，不作任何橫向聯絡[6]，也就是說，機器1可以把它的結果傳給它下面的機子，像a_1，a_2，…，a_{n_1}而每一個機子又可以把它們的結果傳給自己的子機，但在a_1，a_2，…，a_n，之間不互相聯絡。以售貨員旅行問題爲例，若有20個城，第一個機子開始，叫下面19個機子各取一個不同的城及計算與A距離，而這個19個機子又將它所求得的距離交給自己的18個子機令它們取一個與自己不同的城加上距離，如此往下，在第十次時，第十階段的機子把它已取9城及總距離告訴下一個機子，叫他們再取一與已取之城不同之城加上距離，如此一直做到第19次，所有路線的距離都有了，在時間上求得所有的距離是$O(n)$（但用了19!個計算機），古克定義凡可以在$O(n^k)$時間內用無限多計算機解決的問題爲一 NP 問題。

現在要記住的是由於無橫向連絡，在所有路徑的距離都有了之後，並沒有解決售貨員問題㈠，因爲不知誰是最短（若加以比較以求最短距離，則要$O(n!)$個比較），因此我們不能說售貨員旅行問題㈠是一個 NP 問題。但上節問題2，售貨員旅行問題乙，任何一個單線都可以知道它的總距離是否不大於B，因此每單線都有一個“Yes”或“No”的答案。只要有一個“Yes”的答案，我們即知道本問題已解決，故問題二是一個 NP 問題。在單線作業中，每個機子可以作三件事。

1. 目前答案不明確，大家各自作業。
2. 某線已找到答案，立刻叫停，大家停止作業，解題完畢。
3. 此路不通，本線不再作業，但不叫停，別線仍然作業。

從上項作用，很容易看出找出答案的計算時間即某線叫停的時間，亦即任何一個有“Yes”答案線中計算量之總和。也就是說找到答案“捷徑”上所需的時間。易言之，在一非確定性計算機系統下，其子機像有“猜測”到捷徑的功能。若在任何計算步驟中，某人猜了一個答案，而計算機可以在$O(n^k)$時間內回答“Yes”或“No”，這個問題即是一個 NP 問題。再以售貨員旅行㈠及圖1爲例，若你猜一個路徑

$$A\to B\to C\to D\to E\to F\to G\to A$$

我們無法知道此路是否最短，但在(B)問題中，一個“Yes”或“No”的結果只要7個加法就可以回答了。因此根據新的定義，問題2是一個 NP 問題。

由這兩個定義，讀者不難看出問題2，4，6，7，8，9㈡與10皆爲 NP 問題，特別是問題9之㈠，㈡，其實是一樣的問題，但如果你

猜二個 m，n，立刻就可知道 a 是否是 mn。

古克定理的關鍵在證明若一種叫滿足問題（Satisfactability Problem）的例子屬於 P，則所有 NP 問題均屬於 P（即此問題屬於 NP-complete），令 $\cdot$，$+$，$-$（且，或，反）表三個基本的邏輯運算（即對 0 與 1 邏輯符號而言，$\bar{0}=1$，$\bar{1}=0$，除了 $1+1=1$ 之外，$+$，$\cdot$ 與一般代數之加乘相同）。令 f 爲一個含有 n 個邏輯變數（$u_1, u_2, \cdots, u_n$）的函數。假如我們可以找到一組 $u_1, u_2, \cdots, u_n$，使得 $f(u_1, u_2, \cdots, u_n)=1$，則 $f(u_1, u_2, \cdots, u_n)$ 稱爲可滿足。例如

$$f(u_1, u_2, u_3) = (u_1 \cdot (\bar{u}_1 + u_2) \cdot u_2 + u_3) \cdot u_3 \tag{1}$$

爲一可滿足函數，取 $u_1=u_2=u_3=1$ 即可，但

$$f(u_1, u_2, u_3) = (\bar{u}_1 \cdot u_1 + u_2 \cdot \bar{u}_2)(u_1+u_2+u_3) \tag{2}$$

永爲 0，故此 f 爲不可滿足。

直覺上，這類問題除了將 $u_1, u_2, \cdots, u_n$ 一個一個以 0，1 代入檢查 2^n 次之外，顯無捷徑可循，古克 1971 年之論文即證明這是一切 NP 問題之母。

古克定理：滿足問題爲NP-complete。

現在已可證明在前節中之問題，除了問題 1, 3, 5, 9, 之外，全是NP-complete 問題。

五、NP問題之近似解

NP 問題既找不到可行的解法，而很大部分的 NP 問題都在計算機語言，程式，電路設計，統計學，程式作業上有大用，因此只好退而求其次找一個可行的近似解。很可惜的是，所有的 NP-complete 問題雖在 NP 的層次上相聯，在近似解上往往各需不同的解法，這些解法多從直觀而來，我們在此舉二個例子。

例1 在第三節問題 5，包裝問題中，若採取“能裝就裝”法，即現有的盒子若可以裝得下，就不用新盒子，則此法所需用之盒子數 k_1 與最可能少的盒子數 k_0 滿足 $k_1 \leq 2k_0+1$。

證明 今令 n 個物品之重爲 $w_1, w_2, \cdots, w_n$ 公斤，因每個盒子只可以裝 1 公斤，故

$$k_0 \geqslant \sum_{i=1}^{n} w_i$$

另一方面，“能裝就裝”法不可能有兩個以上的盒子同時少於 1/2 公斤，故

$$k_1 \leqslant 2\sum_{i=1}^{n} w_i + 1$$

本例得證。

這個問題的結果是說，我們大約可以用“能裝就裝”法做得最好情形的一半好。經過較複雜的證明，Johnson 在 1974 年證得，當 n 很大時，

(i) $k_1 \leqslant \frac{17}{10} k_0 + 2$，且存在一種情形能產生。

(ii) $k_1 \geqslant \frac{17}{10}(k_0 - 1)$

也就是用“能裝就裝”法不會壞到 70% 以上，但可以壞到多用了 70% 的盒子。

售貨員旅行問題的一個直觀走法是先訪問最近那個尚未訪問過的城，稱爲“先訪近城”法，以圖 1 爲例，其走法爲

$$A \to G \to C \to B \to D \to F \to E \to A$$

Rosenkrantz 等在 1977 年證明這並不是一個很理想的走法，他們證出若各城間的距離

滿足三角不等式[7]，則“先訪近城”法所走之總程 D_1 與最短路徑 D_0 之關係爲

$$D_1 \leqslant \frac{1}{2}(\ \lceil \log_2 n \rceil + 1\ )\ D_0$$

且當 n 很大時，可以有一種情形使得

$$D_1 \geqslant \frac{1}{3}(\ \log_2 n + \frac{4}{3}\ )\ D_0$$

上式中之 $\lceil x \rceil$ 表示大於 x 之最小整數，例如 $\lceil 5 \rceil = 5$，$\lceil 2.5 \rceil = 3$。因 $\log_2 n$ 當 n 大時可以很大，故 D_1 可與 D_0 相差非常之大，但在同一篇論文之中，Rosenkrantz 等證明另一種複雜的“直觀”走法可以達到 $D_1 \leqslant 2D_0$ 之地步。

在上面的定理中，三角不等式的條件很重要，若城之距離無此關係存在時，Sahni 與 Gonzalez 在 1976 年證得：若 $P \neq$ NP，則不可能存在一個有限的 m，及一個 $O(n^k)$ 計算量的走法，能使其全程長 D_1 在任何 n 時滿足

$$D_1 \leqslant m\, D_0$$

即上式中 m 非等於無限大不可，亦即所有 $O(n^k)$ 的做法都不很好。

六、NP-hardness與圍棋

不是所有的難題都可歸結爲 NP 問題，像下得一手絕對好的圍棋現在目前的推測是比所有 NP 問題還要難的計算問題，即NP-hard 問題，NP-hard 問題的定義如下：

若 x 爲一NP-hard 問題，則若 NP $\neq P$，則 $x \notin P$

也就是說，即使 $P =$ NP，x 還不一定屬於 P，但 $P \neq$ NP，則 x 絕不比 NP 的問題容易。

在第三節中的問題 1，3 不一定是 NP 問題，但若能以 $O(n^k)$ 的計算量解決它們，則比較容易的問題 2 與 4 也可以 $O(n^k)$ 解決，故若問題 1，$3 \in P$ 則問題 2，$4 \in P$，又因 2，4 是 NP-complete，即推出 NP $= P$。這與 NP-hard 之定義相合，故問題 1，3 均爲 NP-hard 問題。同理問題 5 也屬於NP-hard，不過這些 NP-hard 似乎比 NP 難不了多少，但下棋問題可能比 NP 問題要難得多，圍棋問題可以作如下觀。

問題 11：（圍棋問題）；以平常的圍棋規則在一個 $n \times n$ 的棋盤上下，給定一個殘局（下了二個子就可以算殘局），首先，是否可以確定黑子在最好的下法之下，一定會贏？

這個問題不能用一般的方法證明它是不是爲 NP。因爲目前沒有人能猜一個必勝的下法且在 $O(n^k)$ 時內證明它是對的，因爲它與對方如何應付有關，而敵方的應付又與他對你以後的下法的推測有關，如此往下走，首先發生困難的是記憶上亮了紅燈，即所需要的記憶可能呈方次以上的進展。

因每一個記憶至少要用（來計算）一次，否則這個記憶就不如不要，因此一個問題的記憶若呈指數上升，則其計算量亦非呈指數似的上升不可，但若某問題只需要方次上升的記憶，即不能保證它只需要方次上升的計算量。

因此計算機學家定義三個新的集合：

PSPACE = {x：x 只需要方次上升的記憶}[8]

PSPACE-complete：

若 $x \in$ PSPACE，

又 $x \in$ PSPACE-complete

且 $x \in P$，

則 $P =$ PSPACE。

PSPACE-hard：

若 $x \in$ PSPACE-hard

且　$x \in P$

則　$P = \mathrm{PSPACE}$

注意在上式中PSPACE-complete $\subset$ PSPACE，即PSPACE-complete是PSPACE中的難題，但PSPACE-hard 不一定屬於PSPACE Stockmeyer and Meyer在 1973 年證明了一個與古克相似的定理。

若令 $\exists x$ 表示存在一個 x，$\forall x$ 表對所有的 x，Q 表 $\exists$，$\forall$ 中的一個，x 爲布氏變數 0 與 1，則我們稱 $f(Q_1 x_1, Q_2 x_2, \cdots, Q_n x_n)$ 爲一量化布氏公式。若 f 有可能爲 1，則 f 稱之爲可滿足，例如把第四節中之(1)式改寫成

$$f(\forall u_1, \exists u_2, \forall u_3)$$
$$= ((\forall u_1) \cdot (\overline{\forall u_1} + \exists u_2)$$
$$\cdot (\exists u_2) + \forall u_3) \cdot (\forall u_3))$$

則上式不可能滿足，因對 $\forall u_3$（u_3 爲 0 或 1）而言，f 不全是 1。

Stockmeyer 與Meyer之定理爲：

定理：檢定一個量化布氏公式爲可滿足是一個PSPACE-complete 問題。

當我們下棋面對着一盤殘局沉思的時候，我們的要求是

對我是否存在一着必勝棋可以對付
敵人任何一着應付棋
此後我是否存在一着必勝棋可以對付
敵人任何一着應付棋
⋮
我是否存在一着必勝棋可以對付
敵人任何一着棋
我贏了

因此這完全是 $\exists$，$\forall$，$\exists$，$\forall$，…之交替作用與Stockmeyer 與Meyer定理之關係至爲密切，Robertson 與 Munro 在 1918 年證得圍棋是一種PSPACE-herd 的問題，目前有人計算到圍棋[9]必勝法之記憶計算量在 10^{600} 以上，不論人腦或電腦的記憶絕少不了一個原子，而現今所知的宇宙原子數約只有 10^{75} 個。棋之道，大矣哉！要做一個下圍棋必勝的機器人是談何容易！

七、結論

現在你明白二十世紀的大難題了，$P = NP$ ？用簡單的語言說，就是是否能找到一個只呈方次增加的方法去解決旅行、包裝、舞會等問題。平凡的問題，期待您不平凡的解答。

小　註

1. 依次排（Sorting）的方法很多，但都不能低於 $O(n \log n)$，讀者可在一般 Database 的書中找到有關Sorting 的法則。
2. 又稱呈多項式上升，但因一個 n 的多須式之大小，在 n 很大時都爲第一項所支配，故可寫成 $O(n^k)$。
3. 並不失去一般性，即若距離不是正整數也可以把它們化成正整數。
4. 在古克的原文中，並沒有NP-complete, NP-hard 之明確定義。但是由於他的論文，使這種分法顯得很自然。不過 NP-hard 之定義仍因人而異，不一定同於本文。
5. 本文中之解決，均指一個 $O(n^k)$ 計量算的解法。

6. 與情報人員之單線作用相同。一個諜報人員只知道他的頂頭上司及他的第一線下層下屬，其餘的人他都不知道。
7. A，B，C表任三城，而$d(A,B)$，$d(B,C)$，$d(A,C)$分別表示A，B；B，C；C，A城之距離，則
$$d(A,B) \leqslant d(A,C) + d(B,C)$$
稱爲三角不等式。
8. x均指問題。
9. 指19 × 19之棋盤，許多計算機學家都是圍棋高手，中國的算盤與圍棋，好像包含了計算機的開始與終極。

參考資料與引用論文

Gorey, M. R. and Johnson, D. S. "Computers and Intractability — A Guide to Theory of NP-Completeness", 1979, Freeman and Company.

Pearl, J. "Hearistics—Intelligent Search Strategies for Computer Problem Solving", 1984, Addsion-Wesley.

Cook, S. A. "The Complexity of theorm-proving procedure" Proc. 3rd Ann. ACM Symp. on Thery of Computing, 1971, 151～158.

Johnson, D. S. et. al. "Worst case performance bounds for simple one-dimensional packing algorithms," SIAM J. Comp. 1974, 299～325.

Rosenkrantz, D. J. et. tl. "An analysis of several heurishics for the traveling salesman problem," SIAM J. Comp. 1977, 563～581.

Sahin, S. and Gonzalez, "P—complete approximation problems" J. ACM 1976, 555～565.

Stockmeyer, L. J. and Meyer, P.R. "Word problems requiring exponential time," Proc. 5th. Ann. ACM Symp. on Theory of Computing, 1973, 1～9.

Robertson, E. and Munro, I. "NP-completeness, puzzles, and games" Utilifas Math. 1978, 99～116.

～本文作者分別任職於
美國佛羅里達大學，統計系及美國海軍研究實驗所～