C/C++ 進階班
演算法

P 與 NP 問題

李耕銘

這章只有學術/考研會用到，聽聽就好

課程大綱

- 問題的分類
- P 與 NP 問題
- NP-hard
- NP-complete (NPC)
- 總結

Warning ：A little math in this chapter ！

問題的分類

問題的分類

- 大抵上電腦科學中的「問題」可以分成兩種
  1. <u>決策問題 (Decision Problem)</u>
     - 只須回答是 (Yes) 或不是 (No) 兩種之一
     - 若回答是，須給出能滿足要求的解
     - 若回答不是，須給出相對應的證明
     - Ex：123 是不是質數？
  2. <u>最佳化問題 (Optimization Problem)</u>
     - 須從在有限的候選答案中選出最佳的解
     - 通常最佳化問題比決策問題難
     - Ex：最佳排班方式、最佳交通途徑

問題的分類

典型的決策問題 (Decision Problem)

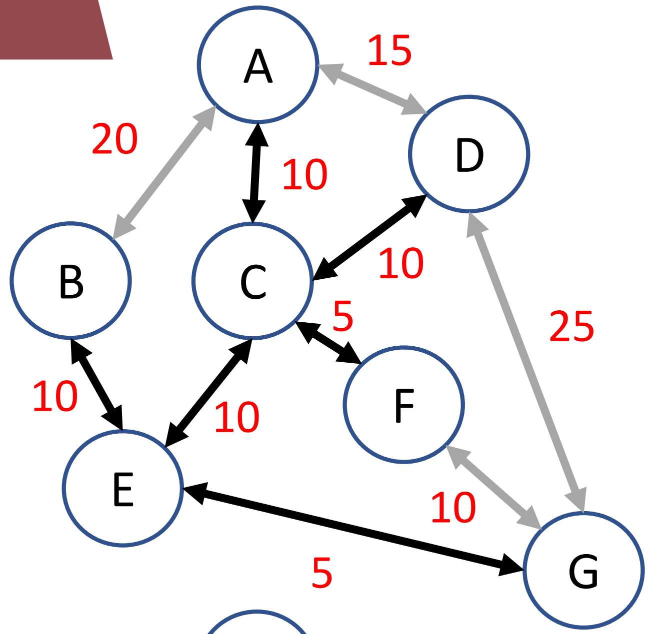
1. Partition Problem (分割問題)
   - 給一正整數的集合，是否可將其分成兩子集合，使這兩子集和數字總和相等。
   - Ex: S = {1, 2, 4, 5, 6}，答案是「可以」
     因為可以分成兩集合：$\boldsymbol{S_1 = \{1, 2, 6\}, S_2 = \{4, 5\}}$
2. Sum of Subset Problem (部份集合的和問題):
   - 給一正整數的集合，其中是否存在一子集合的和為特定常數 C
   - Ex: S = {1, 2, 4, 5, 6}, C = 15，答案是「是」
     因為可以找出一子集合：$\boldsymbol{S_1 = \{4, 5, 6\}}$

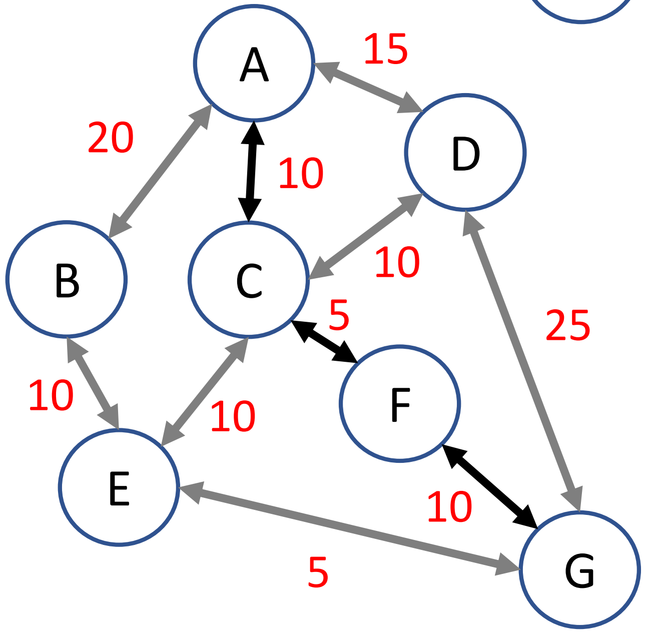
問題的分類

典型的最佳化問題 (Optimization Problem)

1. <u>Minimal Spanning Tree Problem (最小生成樹問題)</u>
   - ➢ 給定由頂點與邊構成的圖，試著從中取出部分邊與所有點讓其形成一顆樹，並且該樹的權重和最小
   - ➢ Ex: $\overline{BE}$、$\overline{EC}$、$\overline{AC}$、$\overline{CF}$、$\overline{EG}$、$\overline{CD}$

2. <u>Shortest Path Problem (最短路徑問題)</u>
   - ➢ 給定由頂點與邊構成的圖，試著找出兩點間的最短路徑
   - ➢ Ex: A、G 間的最短路徑為：A、C、F、G





問題的分類

- 最佳化問題均可轉換成與對應的決策問題
  - 稱為這個最佳化問題的決策版本
  - Ex：最短路徑問題
    - ✓ 最佳化問題：

      給定由頂點與邊構成的圖，試著找出兩點間的最短路徑
    - ✓ 決策問題：

      給定由頂點與邊構成的圖，以及一個常數 C，能否找出一路徑使兩點間的最短路徑 < C？
- 但決策問題不一定可以轉換成與對應的最佳化問題
  - 原最佳化問題會比該問題的決策版本難

問題的分類

如何描述問題對人的難度？

- 這問題好難喔，沒有計算機我不會。
- 這問題好難喔，不管有沒有計算機我都不會。
- 這問題好難喔，不只有我，大家都不會。

問題的分類

問題對電腦的難度呢？

- 究竟是難在哪裡？
  1. 問題已有近似解，想進一步找出最佳的解法卻很困難
  2. 問題本身就很難找出簡單的解決方式
- 用複雜度來描述難度？
- 精確點的說是把問題分成
  1. 易解的 (Tractable)
  2. 難解的 (Intractable)

問題的分類

難解的 (Intractable) 問題

- 在最壞狀況 (Worst-case)下，仍沒有辦法找到多項式時間的問題解法，此問題就被稱為難解 (Intractable)。
- 但若目前還找不到多項式時間的解法，也無法保證未來就一定找不到，並無法證明此問題是難解的。
- 現在找不到，不代表以後找不到。

問題的分類

根據問題的難度與複雜度可以區分成以下四種

1. P (Polynomial Time)：存在多項式時間複雜度的演算法來解決問題
2. NP (Nondeterministic Polynomial Time) ：存在多項式時間複雜度的演算法來驗證問題的解答是否正確
3. NP-hard：目前還沒有找到多項式時間複雜度的算法，也尚不確定每一組解能不能在多項式時間的算法內被驗證
4. NP-complete(NPC)：目前還沒有找到多項式時間複雜度的算法，但每一組解都可以被多項式時間的算法驗證，NPC = NP ∩ NP-hard

聽完不懂很正常，接下來我們會依依來解釋。

P 與 NP 問題

P 與 NP 問題

- P 問題 (Polynomial Time)
  - ➢ 該問題可以在最壞狀況 (Worst-Case) 下被多項式時間內的算法解決的問題
  - ➢ Easy to find.
- NP (Non-Deterministic Polynomial Time) 問題
  - ➢ 不確定(可能可以，也可能不可以)在多項式時間是否可被解決的問題
  - ➢ 但可以在最壞狀況 (Worst-Case) 下被多項式時間的算法驗證
  - ➢ 注意 NP 並不是 not polynomial time
  - ➢ Easy to check.
- P 問題一定是 NP 問題，$P \subseteq NP$
  - ➢ 可以在多項式時間內被解決，就可以在多項式時間內被驗證

P 與 NP 問題

何為多項式時間

$$O\left(a_0 + a_1 n + a_2 n^2 + a_3 n^3 \ldots\ldots + a_k n^k\right) = O(n^k)$$

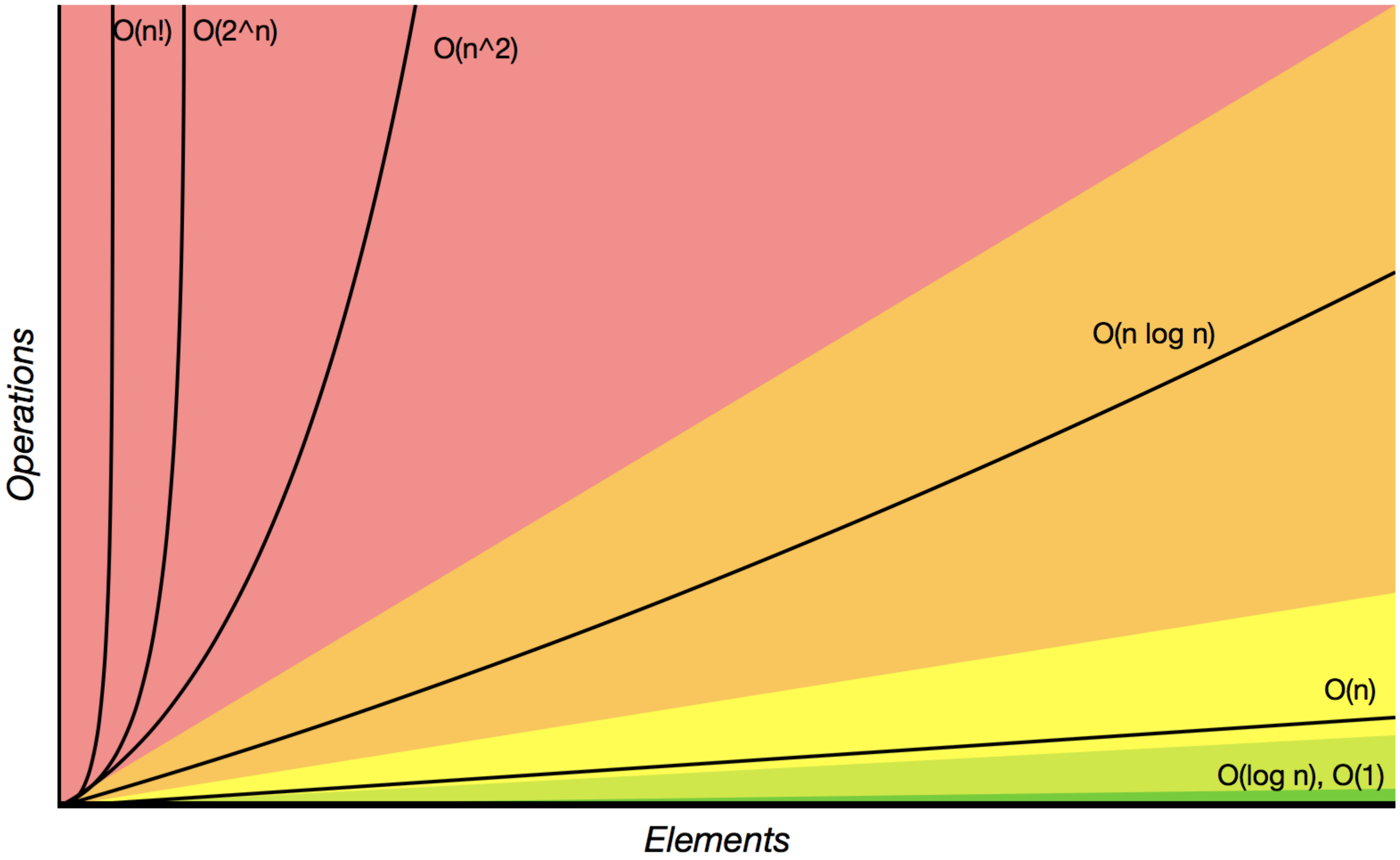
	$log_2 n$	n	$nlog_2 n$	n^2	n^3	2^n	$n!$
10	3.32	10	33.22	10^2	10^3	~10^3	3628800
10^2	6.64	10^2	664.39	10^4	10^6	~10^{30}	~10^{158}
10^3	9.97	10^3	9965.78	10^6	10^9	~10^{300}	X
10^4	13.29	10^4	132877.12	10^8	10^{12}	~10^{3000}	X

Polynomial

Non-Polynomial

P 與 NP 問題

Non-Polynomial Polynomial



P 與 NP 問題

Q：給定 400 位學生與一個有 100 個床位的宿舍，但同時有一份不相容的名單，名單上的學生兩兩成對，名單上的每一對學生都不能同時住宿在宿舍中，請給定一個分配宿舍的方式。

- 此為 NP 問題，因為給定一解，很容易驗證是否成立
  - 依序檢查該名單上的學生是否同時存在於解中
- 但無法給出必定能在多項式時間複雜度中解決的算法

Ref: Clay Mathematics Institute

P 與 NP 間的關係

- 基本上我們能夠解決大部分的 P 問題
- P ~ 可以解決的問題， NP ~ 無法被解決的難題
- P 問題又屬於 NP
- 那 NP 問題是否屬於 P 問題呢？
  - ➢ 如果能夠證明 P = NP
    - ✓ 我們可以解決所有 NP 難題，世紀大發現！
  - ➢ 如果能夠證明 P ≠ NP
    - ✓ 我們注定無法解決這些 NP 難題，世紀大發現！

千禧年世紀難題

- 千禧年世紀難題：100萬美金/題
  1. P/NP問題
  2. 霍奇猜想
  3. 黎曼猜想
  4. 楊-米爾斯存在性與質量間隙
  5. 納維-斯托克斯存在性與光滑性
  6. 貝赫和斯維訥通-戴爾猜想
  7. 龐加萊猜想

NP-hard

NP-hard

- 歸約 (reduction)
  - 把某個計算問題A 轉換為另一個計算問題 B
    - ✓ 寫成 A $\leq_P$ B
  - 有問題B 的多項式時間解法，那問題A 同樣有多項式時間解法
  - 歸約化具備傳遞性
    - ✓ 問題A 可歸約為問題B，問題B 可歸約為問題C
    - ✓ 問題A 便可歸約為問題C
    - ✓ 透過傳遞性連結眾多 NP 類問題，最終會出現 NP-hard 問題
    - ✓ 若此問題同時是 NP 問題，則為 NPC 問題

NP-hard

- 歸約 (reduction)
  - ➢ 問題B 至少跟問題A 一樣難
    - ✓ Ex：判斷**是否是質數**的問題(A)，轉成**找出所有因數**(B)
    - ✓ 寫成 A $\leq_P$ B
    - ✓ 如果可以解決問題B 就可以解決問題A
    - ✓ 如問題B 有多項式的時間解，問題A 亦有多項式的時間解
    - ✓ 如問題A 沒有多項式時間解，則問題B 亦沒有多項式時間解

Ref: CMU

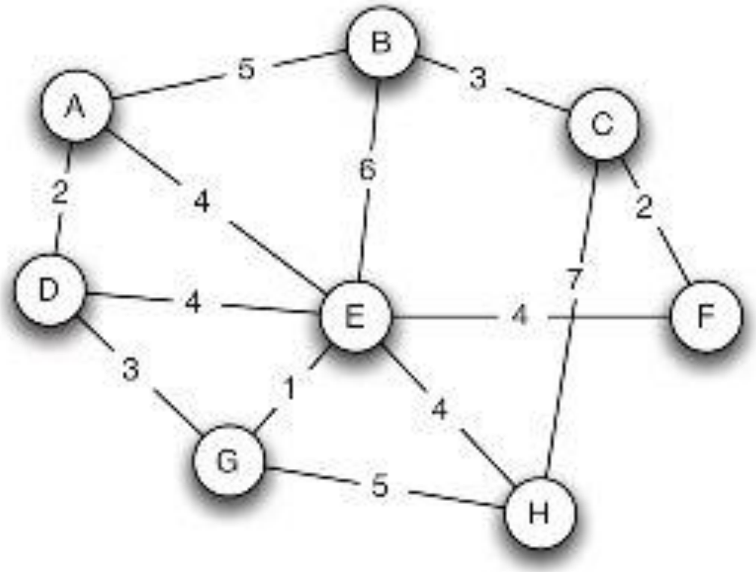
NP-hard

- NP-hard
  - 所有 NP 問題可以歸約化到 NP-hard 問題
    - ✓ NP $\le_P$ NP-hard
  - 但是 NP-hard 問題不一定是 NP 問題
- 為何要歸約成單一問題？
  - 因為演算法的問題實在太多，如果能把所有問題歸約成單一問題，並用 P 問題解決之，所有演算法問題都可以被解決！
  - Ex：世界上的問題都可以歸究給貧富差距，只要解決貧富差距那世界上就沒有問題了！

NP-hard

Q：旅行業務員問題 (Travelling Salesman Problem)

有一個業務員需要不斷地各個城市拜訪，在每次的出差中，每個城市都只能經過一次，在拜訪完每個城市後必須回到原本的城市，並且每一城市都有航班可以到其他所有城市，在給定所有城市與飛機航線的飛行時間後，請找出一個路徑可以讓這趟旅程所花的時間最短。



NP-hard

如果城市數目比較小：

- 3 個城市 ： 1 組可能解
- 5 個城市 ： 12 組可能解
- 10 個城市 ： 181440 組可能解

通解是：

- n 個城市： (n-1)! / 2

無法在多項式時間內解決

NP-hard

若有 a~z 共 26 個城市：

- 共有 25!/2 條路徑可以選擇，
- $\frac{25!}{2} \sim 1.55 \times 10^{25}$

假設每秒可以計算一百萬 (10^6)條路徑，一年有3.15×10^7 秒

- $\frac{1.55\times10^{25}}{10^6\times3.15\times10^7} \sim 5 \times 10^{11}$，約莫是 五千億年

NP-complete

NP-complete

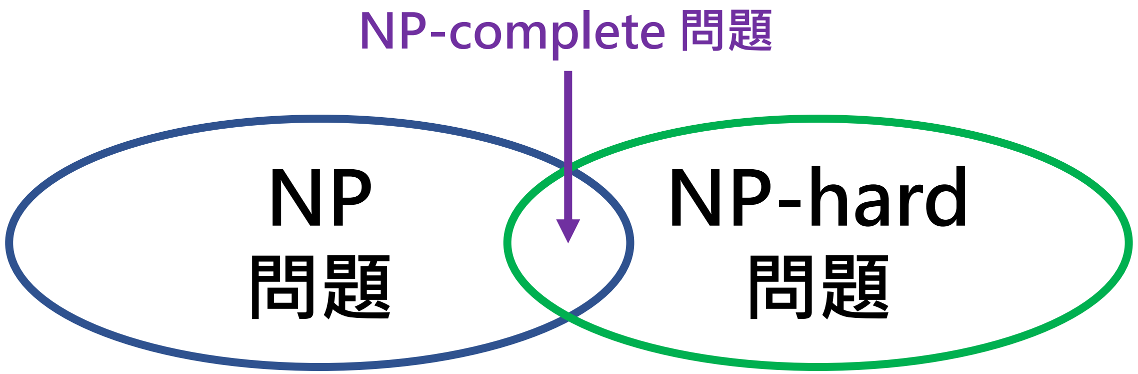
- 目前 P=NP 或 P ≠ NP 的問題還沒有被證明
  - ➢ 普遍的共識是 P=NP 問題應不成立
  - ➢ 至少有一個 NP 的問題是無法在多項式時間內解決
- 為什麼會有這個共識？
  - ➢ 因為 NPC 問題的發現
- NPC = Non-deterministic Polynomial Complete proble
  - ➢ 大量的NP問題經過歸約發現的終極 NP 問題
  - ➢ NPC 問題是 NP類中「最難」的問題

NP-complete

- 1971年 Stephen A. Cook提出了*Cook-Levin理論*
  - ➢ 任一 NP 決策問題都可在多項式時間內**轉成同一個問題**
    - ✓ 「布林方程式是否存在解」
  - ➢ 所有問題殊途同歸到「布林方程式是否存在解」
  - ➢ 只要能在多項式時間內解決「布林方程式是否存在解」
    - ✓ 就能在多項式時間內解決所有 NP 決策問題
    - ✓ P=NP，100萬鎂！

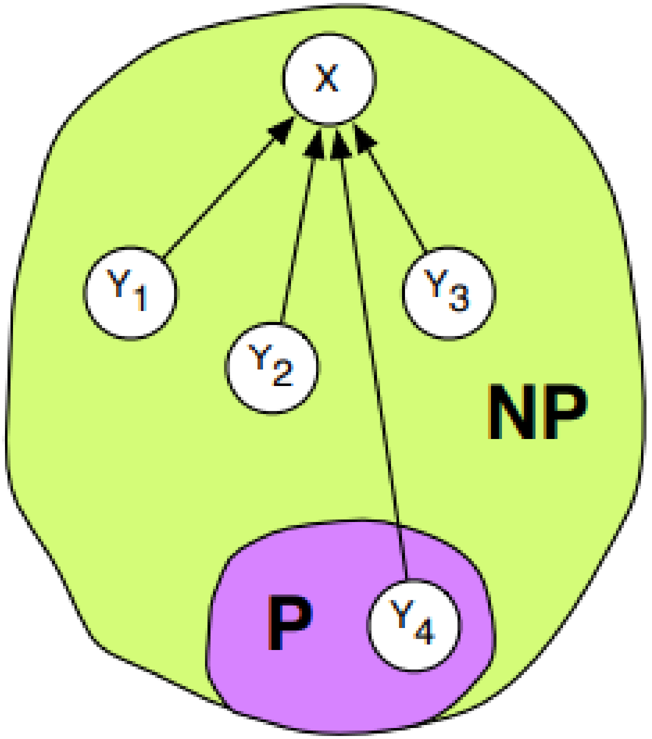
NP-complete

- NPC 問題一定是 NP問題
  - ✓ 跟 NP-hard 問題不同之處， NP-hard 問題不一定是 NP問題
  - ✓ NPC 問題是 NP-hard 問題的特例
- 所以 NP-hard 問題比 NP-complete 問題更難
  - ✓ 證明NP-hard 問題必須要證明任意一 NP 問題都能歸約到該問題
- NP-hard 不一定完全包含NP，兩者的交集為NPC



NP-complete

- 歸約 (reduction)
  - ➢ NP-complete 的定義
    - ✓ Y 是 NP 問題
    - ✓ 對所有的 NP問題Y ，都可以歸約成 X
      - ◻ $Y \leq_P X$
    - ✓ 則代表問題X 是NP問題中最難的 :(



Ref: CMU

NP-complete

- NPC 問題
  - ➢ NP-complete(NPC) = NP ∩ NP-hard
  - ➢ 所以要證明問題是 NPC 非常困難
    - ✓ 需同時證明屬於 NP 和N-Phard
  - ➢ 要證明問題屬於NP 問題不太困難
    - ✓ 瓶頸在證明屬於 NP-hard 問題

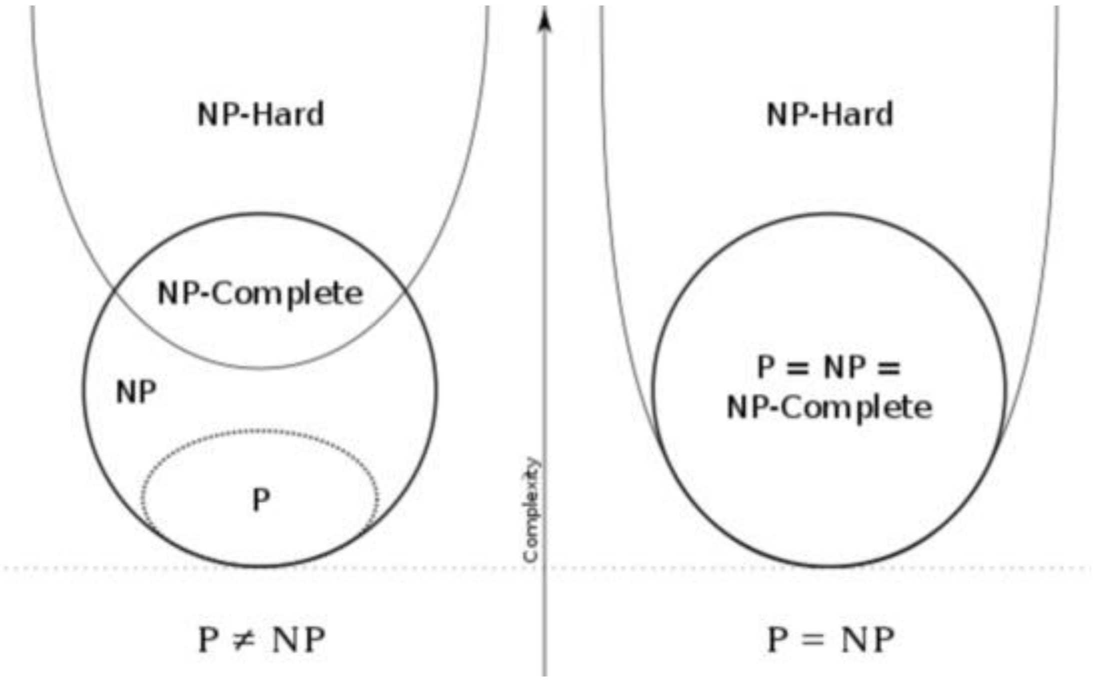
NP-complete

- NPC 問題
  1. NPC 問題屬於 NP 問題
  2. 所有 NP 問題都可以歸約化到該 NPC 問題
  3. 只要能解決 NPC 問題，就能解決所有 NP 問題
  4. NPC 問題是 NP 問題的大魔王
     - ✓ 目前已經找出上百個 NPC 問題，只要解決其中一個，P=NP

總結

總結

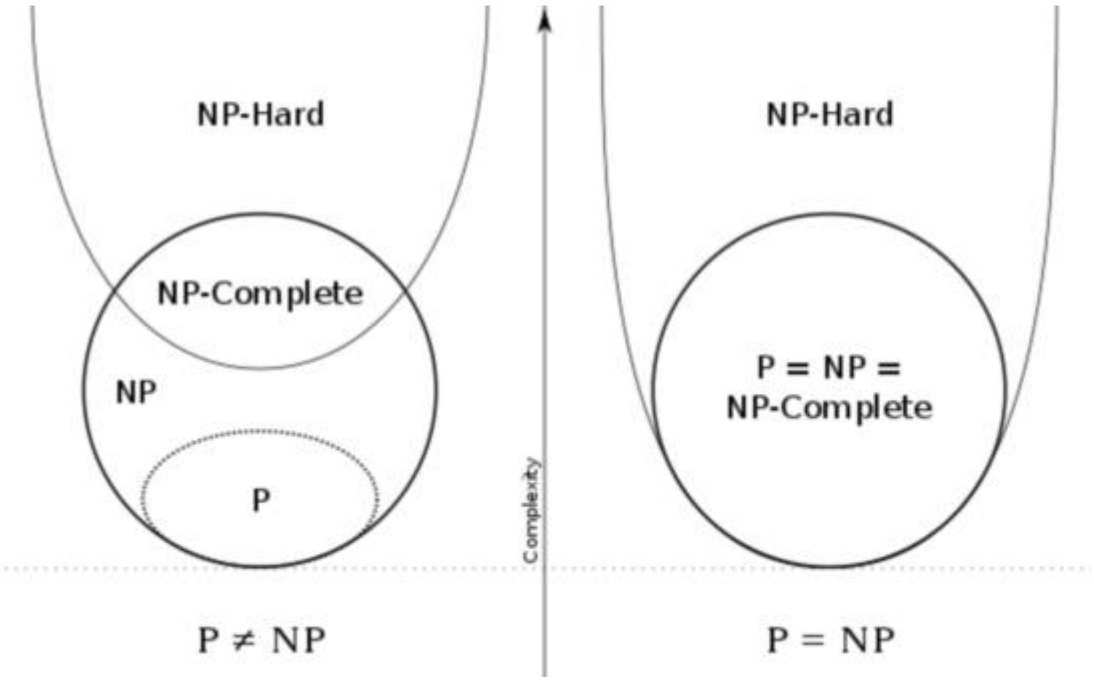
1. P (Polynomial Time)
   - 有簡單的解法
2. NP (Nondeterministic Polynomial Time)
   - 有簡單的驗證方法
3. NP-hard
   - 可以把所有 NP問題歸約成 NP-hard 問題
   - 還沒有找到簡單的解法與驗證法
4. NP-complete(NPC)
   - 可以把所有 NP問題歸約成具 NP 性質的 NPC 問題
   - 還沒有找到簡單的解法，但有找到簡單的驗證方法
   - 是 NP問題與 NP-hard 問題的交集



總結

1. 幾乎所有惱人的決策問題都是 NP問題
2. 這些問題中，可以找到多項式解的叫做 P問題
3. 我們試圖透過歸約把眾多問題歸約成單一問題
   - ✓ 只要要能解決這個問題，就能夠解決所有難題
4. 但還有一大堆問題目前還找不到多項式時間解

<u>~宇宙很大，人的所知真的很渺小~</u>



總結

作者 jackliao1990 (j) 看板 Gossiping
標題 [爆卦] 德國密碼學家宣稱自己摧毀了RSA加密法
時間 Mon Apr 12 21:49:06 2021

https://eprint.iacr.org/2021/232.pdf

RSA加密法於1977年由Rivest、Shamir和Adleman提出,因為極大數的質因數分解困難度,此方法成為世界上應用最廣泛的加密法。目前被破解的RSA密鑰最長紀錄是768個位元,因此一般認為2048位元的密鑰非常安全可靠。

然而德國密碼學家Claus Peter Schnorr在自己新論文摘要中的最後一句宣稱:本文"摧毀"了世界各大機構都在用的RSA加密法。此文一放上網就引起轟動。

如果這篇論文出自無名小卒,大家只會當成笑話。但是這篇作者Claus Peter Schnorr是知名密碼學家,他提出的Schnorr簽章在加密貨幣如比特幣中被廣泛應用,他還是RSA數學卓越獎和萊布尼茲獎得主。

一般認為,我們要等到使用秀爾演算法的量子電腦普及後,RSA加密法才會被破解。然而本文宣稱透過晶格密碼學中的SVP法(尋找最接近向量),即使使用傳統電腦,我們也有機會比二次篩選法和普通數域篩選法(已知最快的傳統因數分解演算法)更快完成分解。

這篇論文目前還未通過同行評審。GitHub上已經有人實作文中的算法,但是沒人成功。也有人指出論文中的可能漏洞:作者宣稱如果使用新算法,"將整數的指數大小加倍"只會讓操作數增加一個數量級。這表示:過去被認為屬於NP問題的操作,被本文證明屬於P,這樣豈不就證明P=NP了?(然而質因數分解從沒被證明是NP完全問題)

Ref: https://www.ptt.cc/bbs/Gossiping/M.1618235350.A.604.html

Take Home Message

- 電腦科學中如何區分難易度？
- P 問題是？
- NP 是？
- NP-complete (NPC) 是？
- NP-hard 是？
- NP-complete (NPC) 跟 NP-hard 差在哪？
- 多項式時間代表？