



計算機演算法

Computer Algorithm

資訊工程學系 丁德榮



Course Information

- Email: deron@cc.ncue.edu.tw or deron@ms45.hinet.net
- 研究室：工學院大樓 237室
- 電話：04-7232105-8445
- Course Web Site:
 - 課程錄影雲端學院 <http://dlearn.ncue.edu.tw>
- 助教：
 - 分機 8410
 - 實驗室：資工系1樓109室



Office Hours

資訊工程學系 專任教師 丁德榮

	星期一	星期二	星期三	星期四	星期五
第1節 08:10~09:00					
第2節 09:05~09:55				54016 資工三 計算機演算法 34304	
第3節 10:15~11:05			Office Hour 237研究室 分機:8445	54016 資工三 計算機演算法 34304	
第4節 11:10~12:00			Office Hour 237研究室 分機:8445	54016 資工三 計算機演算法 34304	
第14節 12:05~12:55					
第5節 13:10~14:00					
第6節 14:05~14:55		54019 資工物聯 碩一 人工智慧 33401教室54020 資工三 人工智慧 33401教室	研究生開會	Office Hour 237研究室 分機:8445	
第7節 15:15~16:05		54019 資工物聯 碩一 人工智慧 33401教室54020 資工三 人工智慧 33401教室	研究生開會	Office Hour 237研究室 分機:8445	
第8節 16:10~17:00		54019 資工物聯 碩一 人工智慧 33401教室54020 資工三 人工智慧 33401教室			
第9節 17:10~18:00					



學習目標

- 演算法的定義。
- 演算法的應用。
- 如何學習與為何學習演算法。
- 演算法問題介紹。



什麼是 Algorithm ?

- A number of rules, which are to be followed in a prescribed order, for solving a specific type of problems.
- Computer Algorithm 的要求：
 - Finiteness (有限個steps)
 - Definiteness (每一個step做的事確定)
 - Effectiveness (不只確定還要足夠的快能做完)
 - Input/Output (O.S.永不terminate只能稱是 computational procedure)



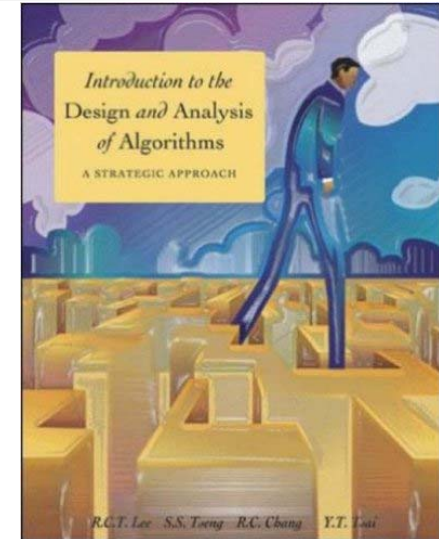
Algorithm is everywhere !

- Application of Algorithm:
 - Operating Systems
 - System Programming
 - Numerical Applications
 - Non-numerical Applications
- :
- 任何一領域都要用 Algorithm 去解決該領域所發生的問題。
- Algorithm Implement 的方式：
 - Software
 - Hardware
 - Firmware

Textbook and References

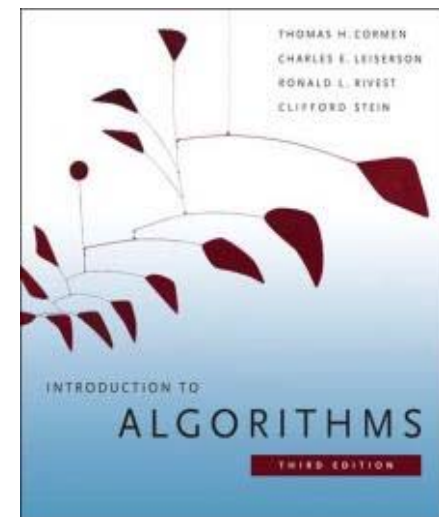
- 教科書：

- Introduction to the Design and Analysis of Algorithms- A Strategic Approach,
 - 李家同 R.C.T. Lee, 張瑞川 R.C. Chang, 曾憲雄 S.S. Tseng, 蔡英德 Y.T. Tsai
 - McGraw-Hill, 2008,
 - 旗標 02-23963257



- 參考書：

- Introduction to Algorithms by T. H. Cormen, C. E. Leiserson and R. L. Rivest, 2009 (3rd) 開發圖書 02-23629900, MIT
- Computer Algorithms C++ by E. Horowitz, S. Sahni and S. Rajasekaran, 2008 (2nd) 東華(新月)圖書 04-23265507





評估方式

- 期中考：35% (closed book)。
- 期末考：35% (closed book)。
- 程式設計：30% 5題，抄襲者0分。



Course Schedule (Part 1)

教學內容與進度

週次	上課日期	教學單元與進度	學生應預習之章節	作業評量與檢討
1	09/15	導論Chapter 1		
2	09/22	Chapter 2 Lower Bounds (Complexity)		
3	09/29	Chapter 2 Lower Bounds (Complexity)		
4	10/06	Chapter 2 Lower Bounds (Complexity)		
5	10/13	Chapter 3 Greedy Method		
6	10/20	Chapter 3 Greedy Method		
7	10/27	Chapter 4 Divide- & -Conquer Strategy		
8	11/03	Chapter 4 Divide- & -Conquer Strategy		
9	11/10	期中考	期中考 (Mid-term)	期中考 (Mid-term)
10				

Course Schedule (Part 2)

序	11/10	期中考	期中考 (Mid-term)	期中考 (Mid-term)
10	11/17	Chapter 4 Divide- & -Conquer Strategy		
11	11/24	Chapter 5 Searching Method		
12	12/01	Chapter 5 Searching Method	學校運動會	學校運動會
13	12/08	Chapter 6 Pruning and Search		
14	12/15	Chapter 6 Pruning and Search		
15	12/22	Chapter 7 Dynamic Programming Strategy		
16	12/29	Chapter 7 Dynamic Programming Strategy		
17	01/05	Chapter 8 NP-complete		
18	01/12	期末考		期末考 (Final exam)



Background for learning

- Background for learning

1. 一個清楚的頭腦
2. Data Structures
3. Discrete Mathematics
4. 用功
5. Coding 能力訓練

- Requirements for researching :

1. 勤讀papers
 2. 一個叛逆的個性
 3. 藝術家似的創造慾望
- ★均可培養



為什麼要學Algorithm？

- 如果不學，你可能用一個很花錢（Time, Space）的Algorithm去解決問題
- Algorithm 的學習是Life-time Job:
 - 永遠知道解某一問題最佳的Algorithm是什麼
 - 知道問題的難度、解法
 - 對於自己的領域隨時要查paper，update最佳的algorithms或至少知道可以問誰。



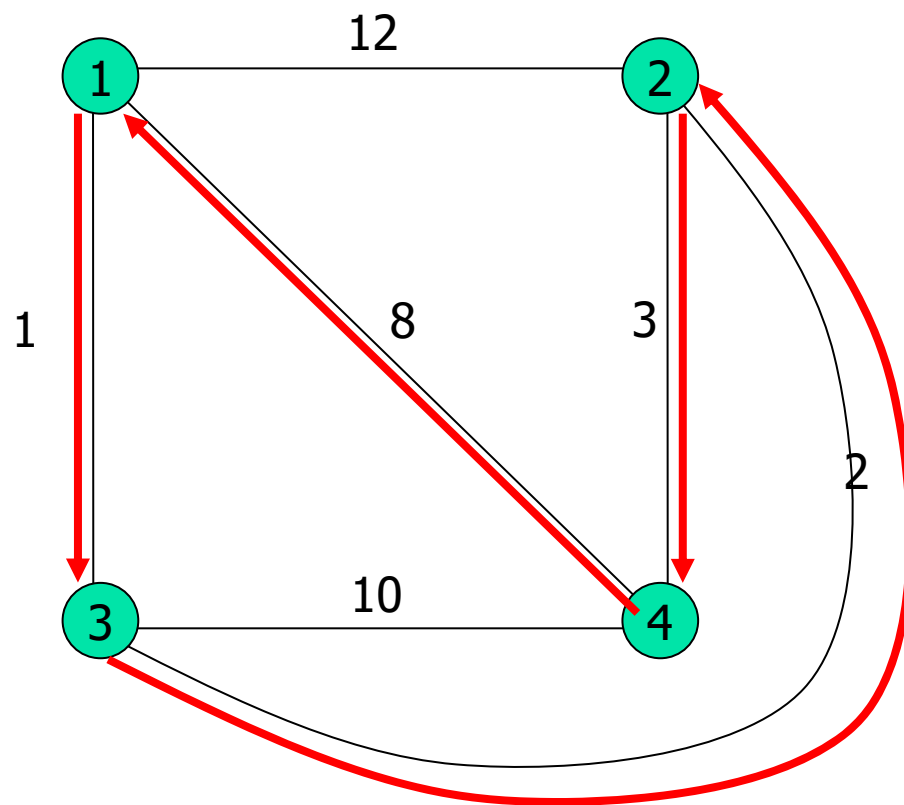
為什麼要學Algorithm ?

- 如果不學，你可能去對 **NP-Complete** 的問題去找 efficient 的解
- Life-time Job:
 - 去知道那些問題是NP-Complete 問題
 - Real Application
 - Average Performance 好的
 - 找Approximating的解
- Examples:
 - TSP : **NP-complete problem**
 - $n = 20$ 執行時間 771世紀
 - $O(n^3 \log n)$ in average (Branch & Bound algorithm)
 - Planar Graph Coloring **P** problem
 - (Maximum # = 4 已被證明)

旅行推銷員問題

Traveling Salesman Problem (TSP)

尋找一個從(1)出發，回到(1)的最短走法



TSP Example



24,978 Cities in Sweden
Solved in 2004



15,112 Cities in Germany
Solved in 2001

TSP是一個公認的難題 NP-Complete

- NP-Complete 之意義：

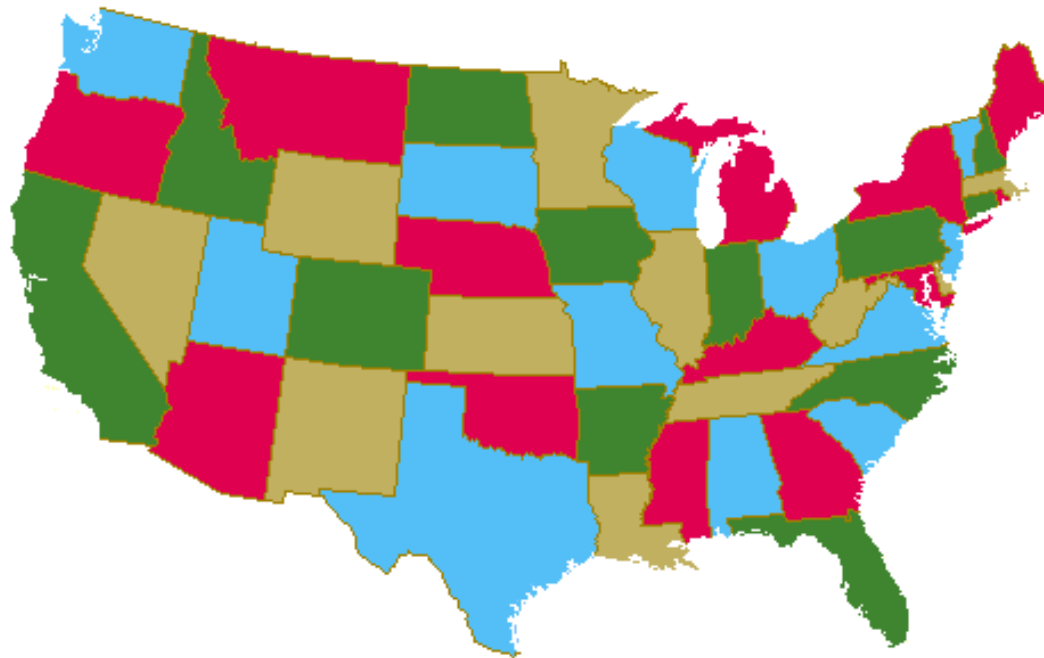
- 我們現在無法對所有輸入找到一個有效率的解法
- 當輸入個數夠多時，需要的解答時間以指數次方成長，至少 $O(2^n)$ 。
- 目前只有exponential algorithm，還沒有人找到polynomial algorithm (你也不妨放棄！)

- 避免浪費時間尋求更佳의解法

	10	30	50
N	0.00001 s	0.00003 s	0.00005 s
N^2	0.0001 s	0.0009 s	0.0025 s
2^n	0.001 s	17.9 min	35.7 year

平面圖著色 Planar Graph Coloring

- 你可知道一幅地圖 (平面圖) 最少可以用多少種顏色來製作呢？
- 限制條件(constraint):
 - 在相鄰國家的顏色必不相同。
 - 於海洋、冰山及文字等的顏色是不計在內的。

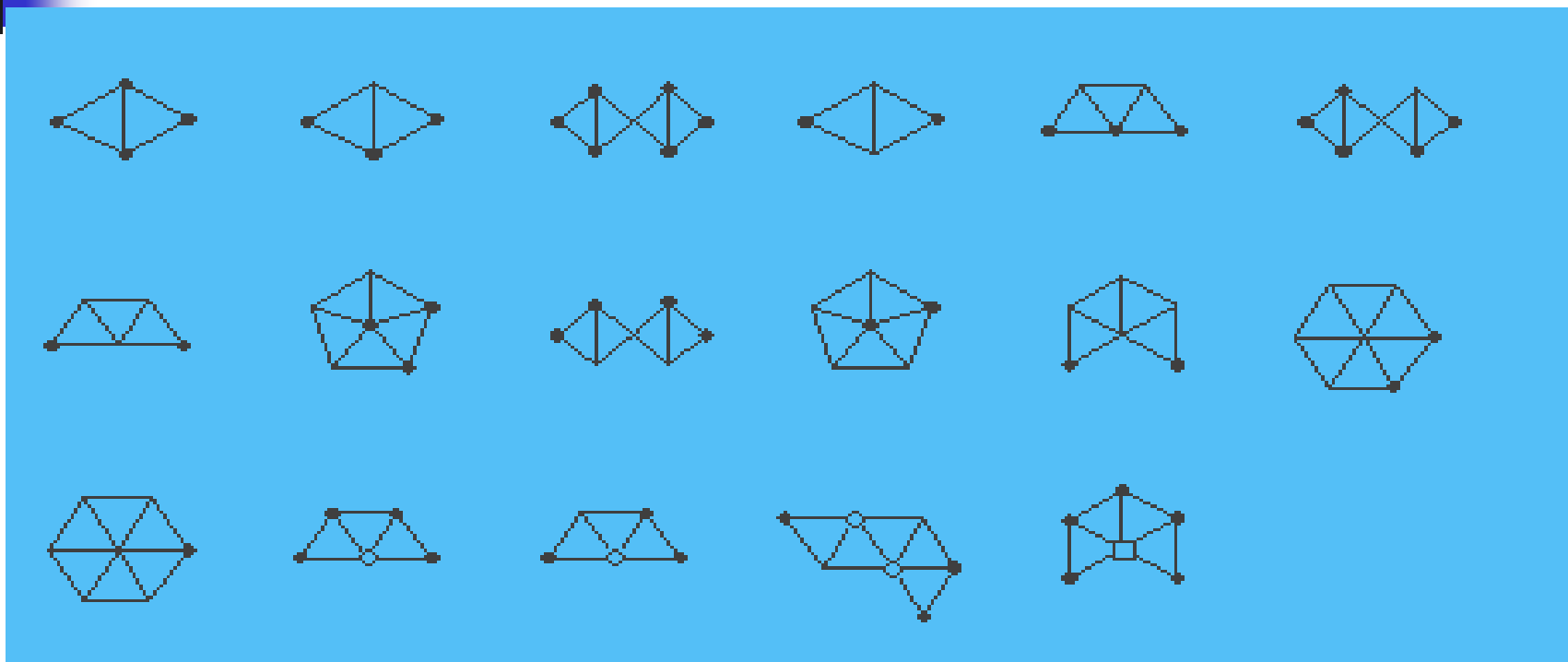




四色定理 證明

- 1970年，數學家才證明出所有少於三十九個區域的地圖，『**四色猜想**』是對的。
- 但是，如果有一千個區域，要等到哪一年才能證明出來呢？於是，有人從不同的方向著手，並成功地將無限多的地圖簡化成**1482種基本圖**。
- 問題是：每種基本圖的顏色組合，就幾乎已經等於無限多了，想要以人工來驗證這一千多種基本圖，根本是不可能的！
- 還好，電腦的出現，解決了這個難題！在**1975年**，數學家利用三台當時最先進的大型電腦，總共花了一千兩百小時的計算，分析驗證了**1482種基本圖**之後，終於證明成功，而使『四色猜想』正式成為『四色定理』！

基本圖



- N. Robertson, D. P. Sanders, P. D. Seymour and R. Thomas, [A new proof of the four colour theorem](#), [Electron. Res. Announc. Amer. Math. Soc. 2 \(1996\), 17-25 \(electronic\)](#).



Course Outline

1. Introduction
2. Lower Bounds (Complexity)
3. Greedy Method
4. Divide-&-Conquer Strategy
5. Tree Searching Strategy (Branch-&-Bound Strategy)
6. Prune-&-Search Strategy
7. Dynamic Programming Strategy
8. NP-Completeness
9. Approximate Algorithms
10. Amortized Analysis
11. Randomized Algorithms
12. On-line Algorithms



Some Books

- **[Cormen, Leiserson and Rivest 2009]** Cormen, T. H., Leiserson, C. E. and Rivest, R. L. : *Introduction to Algorithms*, McGraw-Hill, New York. (2009 3rd)
- **[Garey and Johnson 1979]** Garey, M. R. and Johnson, D. S. (1979): *Computers and Intractability: A Guide to the Theory of NP Completeness*, W. H. Freeman and Co., San Francisco, California.
- **[Horowitz, Sahni and Rajasekaran 2008]** Horowitz, E., Sahni, S. and Rajasekaran, S. : *Computer Algorithms*, W. H. Freeman, New York.
- **[Knuth 1968]** Knuth, D. E. (1968): *The Art of Programming, Vol. 1: Fundamental Algorithms*, Addison-Wesley, Reading, Mass.
- **[Knuth 1973]** Knuth, D. E. (1973): *The Art of Computer Programming, Vol. 3: Sorting and Searching*, Addison-Wesley, Reading, Mass.