計算方法設計與分析
Design and Analysis of Algorithms

Chin Lung Lu (盧錦隆)

Department of Computer Science

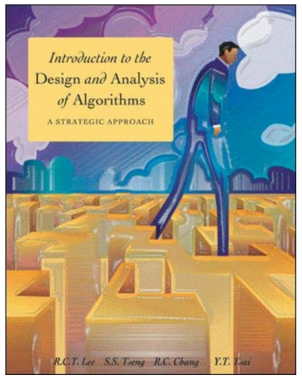
National Tsing Hua University

課程說明 (Course description)

- 讓同學了解何謂演算法，包括知道演算法有哪些設計策略、如何去分析一個演算法的好壞，來進一步加強撰寫程式的能力，以期未來可以根據不同的情況，運用不同的策略，來設計出自己的有效率演算法。

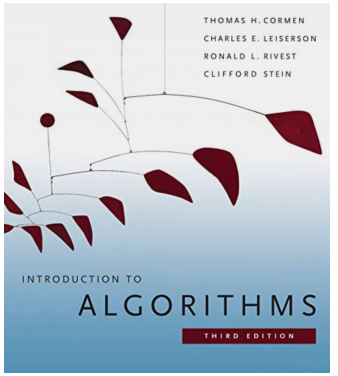
教科書 (Text book)

- Lee, R. C. T. (李家同教授), Chang, R. C., Tseng, S. S. and Tsai, Y. T., "Introduction to the Design and Analysis of Algorithms, a Strategic Approach", McGraw Hill, 2005.



參考書籍 (Reference book)

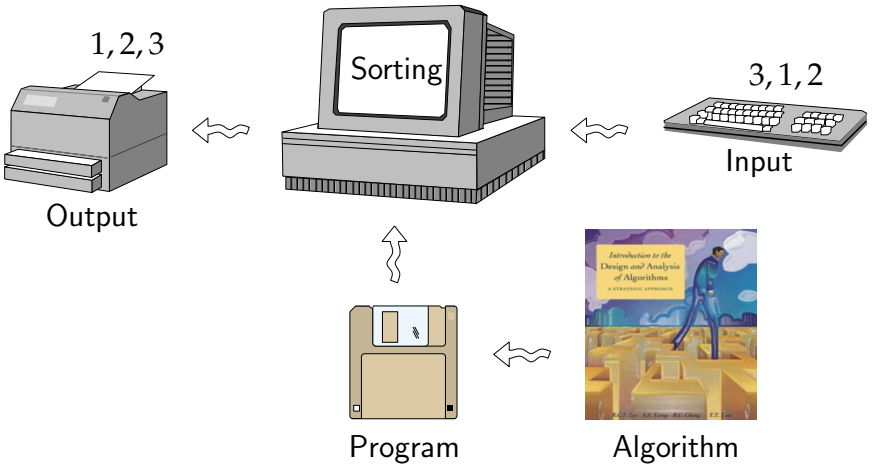
- Cormen, T. H., Leiserson, C. E., Rivest, R. L. and Stein, C., "Introduction to Algorithms" (3rd edition), MIT Press, 2009.



What is algorithm?

- Simply speaking, an algorithm is a computational method that can be used by computers to solve a problem.
- More importantly, we can implement a program based on this algorithm such that the program can automatically solve the problem.

What is algorithm? (cont'd)



What is algorithm? (cont'd)

Algorithms are recipes of computational problems

Why should we study algorithms?

- It is commonly believed that in order to obtain high speed computation, it suffices to have a very high speed computer.
- However, this is not entirely true, because a good algorithm implemented on a slow computer may perform much better than a bad algorithm implemented on a fast computer.

Why should we study algorithms? (cont'd)

Sorting problem

Sorting problem:

- Input: A sequence of data elements.
  Example: 11, 7, 14, 1, 5, 9, 10.
- Output: A sorted sequence of data elements.
  Example: 1, 5, 7, 9, 10, 11, 14.

Consider two sorting algorithms below:

1. Insertion sort algorithm
2. Quick sort algorithm

Why should we study algorithms? (cont'd)

Comparison of insertion and quick sorts

- In fact, it can be proved theoretically that quick sort is much better than insertion sort.
- The question is how good it is.
- To compare quick sort and insertion sort, the authors (李家同教授等人) of the textbook implemented quick sort on an Intel 486 (slow personal computer) and insertion sort on an IBM SP2 (fast super computer).

Why should we study algorithms? (cont'd)

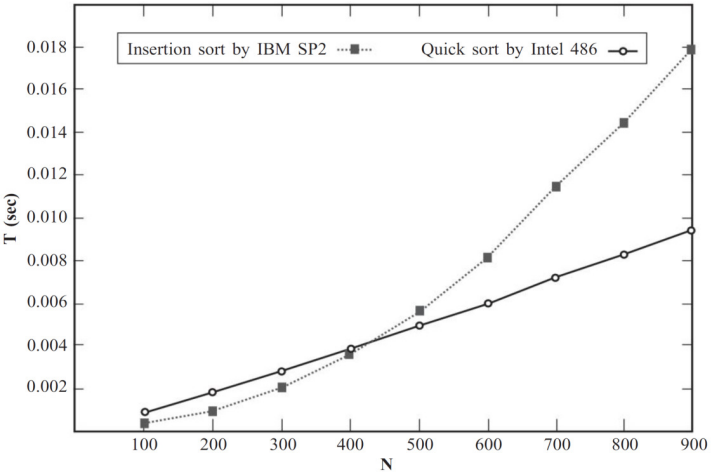
Comparison of insertion and quick sorts

PC (Intel 486)
Quick sort

Super computer (IBM SP2)
Insertion sort

Why should we study algorithms? (cont'd)

Comparison of insertion and quick sorts



A fast computer with an inferior algorithm may perform worse than a slow computer with a superior algorithm.

Algorithm strategies

- The study of algorithm design is almost the study of strategies.
- Many researchers have discovered excellent strategies that can be used to design efficient algorithms.
  - Greedy strategy (Chapter 3)
  - Divide and conquer strategy (Chapter 4)
  - Branch and bound strategy (Chapter 5)
  - Prune and search strategy (Chapter 6)
  - Dynamic programming strategy (Chapter 7)
- Moreover, we must be able to analyze algorithms to determine their performance.
- Some basic concepts related to the analysis of algorithms will be introduced in Chapter 2.

Complexity of problems

- Given a problem, if we already know that there is an algorithm that solves the problem in polynomial time, then it is an easy problem; otherwise, it is a difficult problem.
- However, if we have not found any polynomial-time algorithm to solve the problem, then we can hardly conclude that we can never find any polynomial-time algorithm to solve this problem in the future.

Complexity of problems (cont'd)

- Luckily, there exists a theory of NP-completeness which can be used to measure the complexity of a problem.
- If a problem is proved to be NP-complete, this problem will be viewed as a difficult problem.
- The probability that a polynomial-time algorithm can be found to solve an NP-complete problem is very small.
- The theory of NP-completeness will be introduced in Chapter 8.

成績考核 (Grade evaluation)

- 作業 (Homework): 30%
- 第一次期中考 (First midterm exam): 20%
- 第二次期中考 (Second midterm exam): 20%
- 期末考 (Final exam): 30%

數位學習平台

https://eeclass.nthu.edu.tw/

約法三章

- 作業不能抄襲、遲交。
- 考試不能做弊、補考。
- 成績不能要求加分。