DTS203TC
Design and Analysis of Algorithms

Dr. Qi Chen
School of AI and Advanced Computing

Module Information

- Title: Design and Analysis of Algorithms
- Code: DTS203TC
- Length: 6 weeks
- Mode of Delivery: Lectures + Labs + Seminar
- Credit value: 5.0
- Private study (reflection and consideration of lecture material and background reading)

Teaching Activities

- Time and Venue:
  - Lectures
    - D1/1 Monday 16:00 – 18:00 TC-AB-2003
    - D2/1 Tuesday 13:00 – 15:00 TC-AB-2003
    - D3/1 Wednesday 10:00 – 12:00 TC-AB-2003
    - D4/1 Thursday 16:00 – 18:00 TC-E-2032
  - Labs
    - D1/1 Thursday 18:00 – 19:00 TC-D-2001
    - D1/2 Thursday 18:00 – 19:00 TC-D-3001
  - Seminar
    - D1/1 Thursday 16:00 – 18:00 TC-E-2032
  - Changes (if necessary) will be informed in advance

Assessment

- Coursework 40%
- EXAM 60%
- Resit: EXAM 100%

Teaching Team

Dr. Qi Chen
Office Room: D5007
Email: qi.chen02@xjtlu.edu.cn
Office hours: Monday 14:00 – 16:00 and Tuesday 15:00 – 17:00

Dr. Pascal Lefevre
Office Room: D5012
Email: Pascal.Lefevre@xjtlu.edu.cn
Office hours: Wednesday 14:00 – 16:00 and Thursday 14:00 – 16:00

Teaching Assistant

- Tong Chen
- tong.chen19@student.xjtlu.edu.cn

- Jingxuan Liu
- jingxuan.liu19@student.xjtlu.edu.cn

Aims of the module

- Demonstrate how the study of algorithmics has been applied in a number of different domains
- Introduce formal concepts of measures of complexity and algorithms analysis
- Introduce fundamental methods in data structures and algorithms design
- Introduce computationally hard problems and possible ways of coping with them

Learning Outcomes

- Describe the different classes of algorithms and design principles associated with them; Illustrate these classes by examples from classical algorithmic areas, current research and applications.
- Identify the design principles used in a given algorithm, and apply design principles to produce efficient algorithmic solutions to a given problem
- Apply using basic data structures in conjunction with classical algorithmic problems
- Show familiarity with formal theories providing evidence that many important computational problems are inherently intractable, e.g., NP-completeness.

Course Contents

- Efficiency of Algorithms and Complexity Measures
  - Examples of algorithmic problems and introduction of complexity in terms of various resources
  - Asymptotic complexity and notation in conjunction with a discussion on the worst-case versus the average-case complexity
- Algorithms and Data Structures
  - Core algorithmic primitives
  - Introduction and analysis of basic data structures with their efficient implementation: stack, queue, and priority queue
  - Rooted trees – efficient data structures with implementation: tree traversal, binary search trees, balanced trees – AVL and 2-3 trees, Graph and their implementations

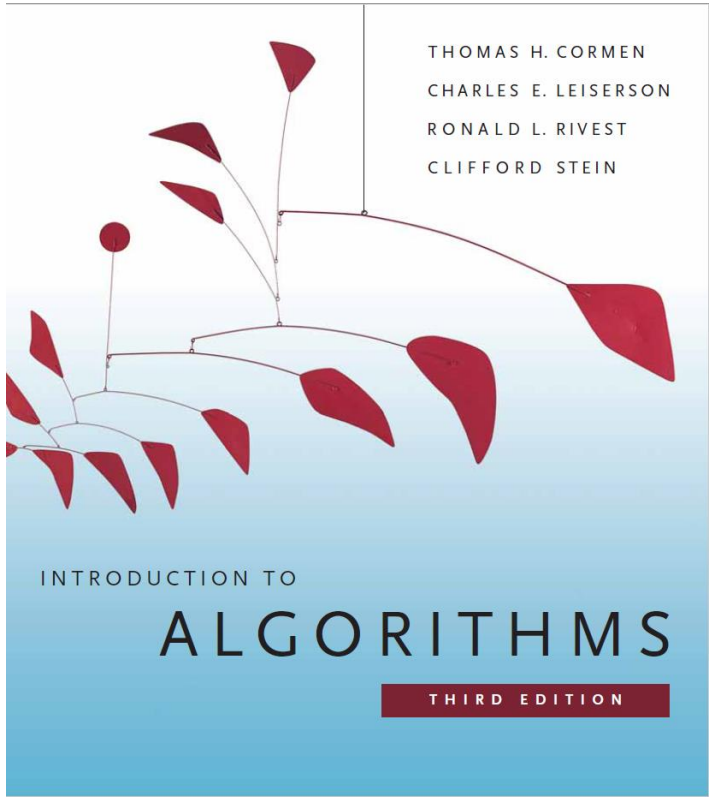
Course Contents – cont'd

- Algorithms and Data Structures – cont'd
  - Advanced graph algorithms, including: network flow algorithms and bipartite matchings.
  - Elementary number theory, Euclid's GCD algorithm, cryptography (from: symmetric encryption, public-key cryptosystem, RSA).
  - Greedy algorithms and divide-and-conquer algorithms, dynamic programming
  - Text processing, including pattern matching (from: Knuth-Morris-Pratt, Boyer-Moore, Rabin-Karp), longest common subsequence (dynamic programming).

Course Contents – cont'd

- Computational Intractability and NP-Completeness
  - Introduction: Comparison of two 'similar' problems (Euler and Hamiltonian cycle); other example problems: 3-Colouring, Satisfiability, k-Clique, and so on; Common features of the problems.
  - The Complexity Class NP: formulation of computational problems in terms of questions about witnesses to solutions; completeness; background to Cook's Theorem and its significance, intuitions behind Cook's Theorem

Recommended Textbook



Introduction to Algorithms (3rd edition)

Thomas H. Cormen,

Charles E. Leiserson,

Ronald L. Rivest

Clifford Stein

The MIT Press