East China Normal University

COMC 21 - Algorithms and Analysis

Instructor: TBD

Email: TBD

Home University: TBD

Semester: June27 to July15, 2022

Course Hour: Monday through Friday, 160 mins per teaching day;

Total Contact Hours: 64 contact hours

Credits: 4

Designated Textbooks with ISBN:

【1】Wu Yonghui, Wang Jiande. Algorithm Design Practice: for Collegiate Programming Contest and Education. CRC Press. 2018. ISBN 9781498776639

【2】Wu Yonghui, Wang Jiande. Data Structure Practice: for Collegiate Programming Contests and Education. CRC Press. 2016. ISBN 9781482215397.

Course Prerequisite:

Programming Language, Data Structure, Discrete Mathematics, Linear Algebra

**Notes: The course might be moved to online delivery due to COVID-19 pandemic. Students will be notified once such decision is made.*

Course Overview

Contents for the course includes:

1) Fundamental Programming Skills: Simple Computing & Recursion;
2) Algorithms for Data Structure;
3) Ad Hoc: Solving Problems by Mechanism Analysis, Solving Problems by Statistical Analysis;
4) Simulation: Simulation of Direct Statement, Simulation by Sieve Method, Simulation by Construction;
5) Greedy Algorithms: Greedy Algorithms in Data Structure, Practice for Greedy Algorithms, Greedy-Choices Based on Sorted Data;
6) Dynamic Programming: Linear Dynamic Programming, Tree-Like Dynamic Programming, Dynamic Programming with State Compression;
7) Algorithms for Number Theory and Combinatorics: Prime Numbers, Greatest Common Divisors and Indeterminate Equations, Generating Permutations, Calculating Numbers of Combinations;
8) Graph Algorithms: Applications of Graph Traversal, State Space Search.

Teaching methods for the course are lectures and experiments.

Learning Outcomes

Upon completion of this course “Algorithm and Analysis”, students should be able to master algorithms systematically, and can master programming skills solving problems using algorithms and data structure.

The course can also be as the fundamental training for programming contests for students.

Grading Scale and Notes

The following definitions will be used as a guide for the assignment of grades:

Number Grade	Letter Grade	Definitions
94-100	A	Extraordinary distinction, indicating a full mastery of course content and excellent work.
90-93	A-	
87-89	B+	Strong performance demonstrating a high level of attainment, indicating a good comprehension of the course material and the student's full engagement with the course requirements and activities.
84-86	B	
80-83	B-	
77-79	C+	Acceptable performance, demonstrating an adequate and satisfactory comprehension of the course material and the student has met the basic requirements for completing assignments and participating in class activities.
70-76	C	
60-69	D	A marginal performance in the required exercises demonstrating a minimal passing level of attainment.
0-59	F	An unacceptable performance. The F grade indicates that the student's performance has revealed almost no understanding of the course content.

Assessment Policy

Assessment	Final Grade
Experiments	30%
Mid-Term Examination	30%
Final Examination	30%
Attendance	10%

Course Schedule

Date	Lecture	Reading/Assignments/ Examination
Day 1	Fundamental Programming Skills(I): Simple Computing	Chapter 1, Data Structure Practice / Experiment
Day 2	Fundamental Programming Skills(II): Recursion	Chapter 3, Data Structure Practice / Experiment
Day 3	Algorithm for Array and String	Chapter 4, Data Structure Practice / Experiment
Day 4	Sorting algorithm (including sorting by STL)	Chapter 7, Data Structure Practice / Experiment
Day 5	Algorithms for Binary Tree	Chapter 9, 10, Data Structure Practice / Experiment
Day 6	Algorithm for Graphs (I): Graph Traversal	Chapter 11, Data Structure Practice / Experiment
Day 7	Algorithm for Graphs (II): Best Path	Chapter 13, Data Structure Practice / Experiment
Day 8	**Mid-Term Examination**	Experiment for solving problems by programming
Day 9	Ad Hoc (Solving Problems by Mechanism Analysis, Solving Problems by Statistical Analysis)	Chapter 1, Algorithm Design Practice / Experiment
Day 10	Simulation (Direct Statement, Simulation by Sieve Method, Simulation by Construction)	Chapter 2, Algorithm Design Practice / Experiment
Day 11	Greedy Algorithms(II): Practice for Greedy Algorithms, Greedy-Choices Based on Sorted Data	Chapter 5, Algorithm Design Practice / Experiment
Day 12	Dynamic Programming: Linear Dynamic Programming	Chapter 6, Algorithm Design Practice / Experiment
Day 13	Algorithms for Number Theory and Combinatorics	Chapter 3 & 4, Algorithm Design Practice / Experiment
Day 14	Graph Algorithms: Applications of Graph Traversal, State Space Search	Chapter 9, Algorithm Design Practice / Experiment
Day 15	**Final Examination**	Experiment for solving problems by programming

Reading List:

【1】Data Structure Practice: for Collegiate Programming Contests and Education

Chapter 1: Practice for Simple Computing (1.1: Improving programming Style, 1.2: Multiple Test Cases, 1.3: Precision of Real Number, 1.4: Improving Time Complexity by Dichotomy)

Chapter 3: Simple Recursion (3.1: Calculation of Recursive Functions, 3.2: Solving Problems by Recursive Algorithms, 3.3: Solving Recursive datum)

Chapter 4: Experiments for Linear Lists Using Direct Access (4.1: Application of Arrays(1): Calculation of Dates, 4.2: Application of Arrays(2): Calculation of High Precision Numbers, 4.3: Application of Arrays(3): Representation and Computation of Polynomials, 4.4: Application of Arrays(4): Calculation of Numerical Matrices, 4.5: Character Strings(1): Storage Structure of Character Strings, 4.6: Character Strings(2): Pattern Matching of Character Strings)

Chapter 7: Sort of Linear Lists (7.1: Using Sort Function in STL, 7.2: Using Sort Algorithms)

Chapter 9: Application of Binary Trees (9.3: Traversal of Binary Trees)

Chapter 10: Application of Classical Trees (10.1: Binary Search Trees, 10.2: Binary Heap, 10.3: Huffman Trees)

Chapter 11: Application of Graph Traversal (11.1: BFS Algorithm, 11.2: DFS Algorithm, 11.3: Topological Sort)

Chapter 12: Algorithms of Minimum Spanning Trees (12.1: Kruskal Algorithm, 12.2: Prim Algorithm)

Chapter 13: Algorithms of Shortest Path (13.1: Warshall Algorithm and floyed-warshall Algorithm, 13.2: Dijkstra Algorithm, 13.3: Bellman-Ford Algorithm)

【2】Algorithm Design Practice : for Collegiate Programming Contest and Education

Chapter 1: Practice for Ad Hoc Problems (1.1: Mechanism Analysis Method, 1.2: Statistical Analysis Method)

Chapter 2: Practice for Simulation Problems (2.1: Straightforward Simulation, 2.2: Sieve Method, 2.3: Construction Method)

Chapter 3: Practice for Number Theory (3.1: Practice for Prime Numbers, 3.2: Practice for Indeterminate Equations and Congruence)

Chapter 4: Practice for Combinatorics (4.1: Generating Permutations, 4.2: Enumeration of Permutations and Combinations)

Chapter 5: Practices for Greedy Algorithms (5.1: Practices for Greedy Algorithms, 5.2: Greedy-Choices based on Sorted Data)

Chapter 6: Dynamic Programming (6.1: Linear Dynamic Programming)

Chapter 9: Practices for State Space Search (9.1: Constructing a State Space Tree, 9.2: Optimizing State Space Search)