湖北工業大學
HUBEI UNIVERSITY OF TECHNOLOGY

Course Title	Advanced Algorithm Design and Analysis
Course Code	CMPT 4441
Semester	Winter 2027, Dec. 21, 2026 - Jan. 15, 2027
Course Length	4 Weeks, 60 Contact Hours
Credits	4
Instructor	Wei Chao
Office	Science and Technology Building A403
Email	9030885373@qq.com
Prerequisite(s)	CMPT 2141 Data Structures and Algorithms MATH 2331 Discrete Mathematics A working knowledge of a compiled programming language, preferably C++ or Java, is expected.

Course Description:

This course presents an in-depth study of advanced techniques for the design, theoretical analysis, and implementation of algorithms. Building upon basic data structures and algorithmic foundations, this course covers concrete models of computation, amortized analysis, advanced tree and range query data structures, network flow network architectures, linear programming and duality, game theory, randomized algorithms, string hashing, online learning (multiplicative weights update), and streaming algorithms.

Through mathematical analysis, algorithm development, programming exercises, and problem-solving activities, students develop the ability to select appropriate computational approaches, establish their correctness, and evaluate their efficiency under different conditions.

Delivery Mode: In-Person, Face-to-Face Instruction

Course Goals:

Students who successfully complete this course will demonstrate competency in the following general education core goals:

- **Critical Thinking Skills** – Students will engage in analytical thinking, demonstrating the ability to critically evaluate, synthesize, and apply knowledge to complex problems, and construct well-reasoned solutions and arguments.
- **Independent Research and Inquiry** – Students will conduct independent research, utilizing academic resources to explore relevant topics, formulating research questions, analyzing data, and presenting findings in a coherent, scholarly manner.

- **Problem-Solving and Application** – Students will apply theoretical concepts and methodologies learned in the course to real-world problems, demonstrating the ability to develop practical solutions informed by academic inquiry.
- **Global and Cultural Awareness** – Students will gain awareness of the global and cultural contexts relevant to the course, appreciating diverse perspectives and considering the implications of their studies in a broader, international context.

Student Learning Outcomes:

Upon completion of this course, students will be able to:

- Construct adversary arguments and decision-tree lower bounds for comparison, exchange, and query models;
- Implement and derive complexity guarantees for segment trees with lazy propagation, universal hashing, and disjoint set structures using potential functions;
- Model complex discrete and continuous optimization problems using Network Flows and Linear Programming Duality;
- Design probabilistic algorithms and analyze online decision-making processes under uncertainty using multiplicative weight updates;
- Formulate formal proofs of correctness, asymptotic running time, and minimax optimality;
- Translate algorithmic designs into working programs and evaluate their performance through computational experiments.

Textbooks/Supplies/Materials/Equipment/ Technology or Technical Requirements:

Primary textbook:

Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein. *Introduction to Algorithms.* 4th ed. MIT Press, 2022. (Hereafter referred to as *CLRS*)

Supplementary textbook:

- Jon Kleinberg and Eva Tardos. *Algorithm Design.* 1st ed. Pearson, 2022. (Hereafter referred to as *K&T*)
- (Additional lecture notes, research readings, and programming exercises may be provided by the instructor to supplement the prescribed readings.)

Software & Technology Requirements:

- Programming environment: A C++ or Java development environment, such as GCC/Clang, Visual Studio Code, or IntelliJ IDEA.
- Computational tools: Basic command-line utilities and debugging tools for program development and performance testing.

Course Requirements:

1. Written Problem Sets (20%)

Five written assignments will assess students' ability to formulate algorithmic solutions and provide mathematical justifications. Problems will involve asymptotic analysis, recurrence relations, algorithm correctness, data structures, graph theory, optimization, and computational complexity. Students must explain their reasoning,

demonstrate the validity of their proposed methods, and establish relevant performance bounds. Each assignment will be submitted individually.

2. Knowledge Checks (15%)
Periodic assessments will evaluate students' understanding of lecture materials and their ability to apply algorithmic concepts to specific computational problems. Questions may include complexity calculations, algorithm tracing, identification of appropriate design strategies, and interpretation of mathematical properties. These assessments will be conducted throughout the semester to reinforce learning and monitor progress.

3. Algorithmic Case Study and Oral Presentation (20%)
Students will select or be assigned a computational problem and develop an algorithmic solution for analysis. They will prepare a short technical report describing the problem, proposed algorithm, correctness considerations, and computational complexity, followed by an individual oral presentation.

4. Exams (45%)
Midterm Examination (20%): Closed-book written exam covering Lectures 1-14. It will include analytical and problem-solving questions concerning asymptotic analysis, sorting, data structures, graph algorithms, and dynamic programming.
Final Examination (25%): Comprehensive closed-book written exam covering the entire semester. It will cover advanced algorithmic techniques, graph optimization, randomized computation, computational complexity, approximation methods, and related topics.

Assessments: Activity	Percent Contribution
Written Problem Sets (5)	20%
Knowledge Checks	15%
Algorithmic Case Study and Oral Presentation	20%
Midterm Examination	20%
Final Examination	25%

Grading:

Final grades will be based on the sum of all possible course points as noted above.

Grade	Percentage of available points
A	94-100
A-	90-93
B+	87-89
B	84-86
B-	80-83
C+	77-79
C	74-76
C-	70-73
D	64-69
D-	60-63
F	0-59

Course Schedule:
The schedule of activities is subject to change at the reasonable discretion of the instructor. Minor changes will be announced in class, major ones provided in writing.

CMPT 4441 Schedule		
Lecture	Topic	Readings
L1	RAM Model, Linear-Time Selection (QuickSelect & Median-of-Medians)	*CLRS,* Ch. 9
L2	Lower Bounds & Decision Trees for Comparison Sorting	*CLRS,* Ch. 8
L3	Non-Comparison Sorting: Counting Sort & Radix Sort	*CLRS,* Ch. 8
L4	Universal Hashing & Hash Tables	*CLRS,* Ch. 11
L5	String Matching & Karp-Rabin Fingerprinting *Written Problem Set 1 due*	*CLRS,* Ch. 32
L6	Advanced Dynamic Programming I: Knapsack & Subproblems	*CLRS,* Ch. 14
L7	Advanced Dynamic Programming II: RNA Secondary Structure & Sequence Alignment	*CLRS,* Ch. 14
L8	Dynamic Range Query Structures: Segment Trees & Interval Aggregations	*CLRS,* Ch. 17
L9	Advanced Tree Augmentation & Range Operations	*CLRS,* Ch. 12, 13, 17
L10	Amortized Analysis: Aggregate, Banker's, and Potential Methods	*CLRS,* Ch. 16
L11	Disjoint Set Structures (Union-Find) & Path Compression *Written Problem Set 2 due*	*CLRS,* Ch. 19
L12	Graph Algorithms & Network Flows I: Ford-Fulkerson & Residual Graphs	*CLRS,* Ch. 24
L13	Network Flows II: Max-Flow Min-Cut Theorem & Bipartite Matching	*CLRS,* Ch. 24, 25
L14	Polynomial Max-Flow & Augmenting Path Strategies	*CLRS,* Ch. 24
EXAM	Midterm Examination (Lectures 1-14)	
L15	Advanced Network Flow Applications (Image Segmentation & Project Selection)	*K&T*, Ch. 7.10-7.11
L16	Linear Programming I: Formulations and Geometric Intuition	*CLRS,* Ch. 29
L17	Linear Programming II: Duality Theory & Strong Duality *Written Problem Set 3 due*	*CLRS,* Ch. 29
L18	Game Theory & Best-Response Dynamics / Nash Equilibria *Algorithmic Case Study - Short Technical Report due*	*K&T*, Ch. 12.7
L19	Online Algorithms & Competitive Analysis Frameworks *Algorithmic Case Study - Presentation (Part.1)*	*CLRS,* Ch. 27
L20	NP-Completeness: Polynomial-Time Reductions & Core Definitions *Algorithmic Case Study - Presentation (Part.2)*	*CLRS,* Ch. 34
L21	PSPACE Complexity & Quantified Boolean Games *Written Problem Set 4 due*	*K&T*, Ch. 9.1-9.3
L22	Approximation Algorithms: Pricing Method & LP Rounding for Vertex Cover	*CLRS,* Ch. 35
L23	Extending Tractability: Fixed-Parameter Tractability & Tree Decompositions *Written Problem Set 5 due*	*K&T*, Ch. 10.1-10.2

L24	Randomized Algorithms: Global Min-Cut & Probabilistic Bounds	*CLRS,* Ch. 5
L25	Parallel Algorithms & Comprehensive Course Review	*CLRS,* Ch. 26
EXAM	Final Examination (Lectures 1-25)	

Accommodation Statement

Academic accommodations may be made for any student who notifies the instructor of the need for an accommodation. It is imperative that you take the initiative to bring such needs to the instructor's attention. Students who may require assistance in emergency evacuations should contact the instructor as to the most appropriate procedures to follow.

Academic Integrity Statement

Each student is expected to maintain the highest standards of honesty and integrity in academic and professional matters. The University reserves the right to take disciplinary action, up to and including dismissal, against any student who is found guilty of academic dishonesty or otherwise fails to meet the standards. Any student judged to have engaged in academic dishonesty in coursework may receive a reduced or failing grade for the work in question and/or for the course.

Academic dishonesty includes, but is not limited to, dishonesty in quizzes, tests, or assignments; claiming credit for work not done or done by others; hindering the academic work of other students; misrepresenting academic or professional qualifications within or without the University; and nondisclosure or misrepresentation in filling out applications or other University records.

Other Items:

Attendance and Expectations

All students are required to attend every class, except in cases of illness, serious family concerns, or other major problems. We expect that students will arrive on time, be prepared to listen and participate as appropriate, and stay for the duration of a meeting rather than drift in or out casually. In short, we anticipate that students will show professors and fellow students maximum consideration by minimizing the disturbances that cause interruptions in the learning process. This means that punctuality is a must, that cellular phones be turned off, and that courtesy is the guiding principle in all exchanges among students and faculty. You will be responsible for the materials and ideas presented in the lecture.

Assignment Due Dates

All written assignments must be turned in at the time specified. Late assignments will not be accepted unless prior information has been obtained from the instructor. If you believe you have extenuating circumstances, please contact the instructor as soon as possible.

Make-Up Work

The instructor will not provide students with class information or make-up assignments/quizzes/exams missed due to an unexcused absence. Absences will be excused and assignments/quizzes/exams may be made up only with written documentation of an authorized absence. Every effort should be made to avoid scheduling appointments during class. An excused student is responsible for requesting any missed information from the instructor and setting up any necessary appointments outside of class.

Access, Special Needs, and Disabilities
Please notify the instructor at the start of the semester if you have any documented disabilities, a medical issue, or any special circumstances that require attention, and the school will be happy to assist.