湖北工業大學
HUBEI UNIVERSITY OF TECHNOLOGY

Course Title	Algorithmic Paradigms and Complexity Analysis
Course Code	CMPT 4521
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 (or equivalent); MATH 2331 Discrete Mathematics (or equivalent); MATH 2401 Linear Algebra and Matrix Theory (or equivalent); Proficiency in a compiled programming language (C++ or Java) and formal proof techniques is required.

Course Description:

This upper-division computer science course provides a rigorous theoretical and analytical foundation in the design, formal proof of correctness, asymptotic complexity analysis, and efficient implementation of advanced algorithms. Building directly upon discrete mathematics, linear algebra, and foundational data structures, the course examines systematic algorithmic paradigms including dynamic programming over trees, intervals, and directed acyclic graphs; amortized complexity analysis via aggregate, accounting, and potential functions; randomized hashing and polynomial fingerprinting; maximum network flows and minimum-cost circulations; linear programming duality; and online decision-making via multiplicative weights.

In addition to exact combinatorial optimization, students investigate fundamental computational limits and approximation guarantees for intractable problems. Core topics include information-theoretic lower bounds in decision-tree, adversary, and query models; polynomial-time reductions and NP-completeness; deterministic and randomized LP rounding for approximation algorithms; Fast Fourier Transform polynomial convolution; and sublinear-space data stream sketches. Integrated problem-solving workshops complement lectures with structured proof construction, edge-case verification, oral proof defenses, and compiled C++/Java benchmarking.

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 formal proofs of correctness and derive tight recurrence and asymptotic running-time bounds for divide-and-conquer, greedy exchange, and multi-dimensional dynamic programming algorithms;
- Prove amortized complexity bounds using aggregate, accounting, and potential-function arguments across advanced data structures (lazy segment trees, self-adjusting trees, and disjoint-set forests);
- Formulate and solve maximum-flow, minimum-cut, bipartite matching, minimum-cost circulation, and linear programming duality models, and establish minimax optimality in two-player zero-sum games;
- Prove information-theoretic lower bounds via decision trees, adversary arguments, and query complexity models, and establish NP-completeness through polynomial-time reductions;
- Design and analyze randomized universal hashing, online competitive algorithms, multiplicative weights update frameworks, LP-rounding approximation algorithms, FFT convolutions, and streaming sketches;
- Implement complex algorithmic solutions in a compiled programming language (C++ or Java) and empirically validate scaling behavior against theoretical worst-case and expected bounds.

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

Primary Textbook:

Jon Kleinberg and Eva Tardos. *Algorithm Design.* 1st ed. Pearson, 2005.

Secondary / Reference Reading:

Jeff Erickson. *Algorithms.* 1st ed. Open-Access Edition, 2019.

Supplementary Lecture Notes: Instructor-provided mathematical monographs covering amortized potential functions, linear programming duality, online competitive analysis, multiplicative weights updates, and streaming sketches.

Software & Technical Requirements: C++ (GCC/Clang supporting C++17 or newer) or Java development environment for algorithmic implementation benchmarks; LaTeX or Markdown editor for typesetting formal mathematical proofs.

Course Requirements:

Written Proof & Problem Sets (20%)
Students complete six individual theoretical problem sets (4 multi-part proof problems per assignment). Each solution is graded rigorously for formal proof completeness, boundary/edge-case handling, correctness invariants, and tight asymptotic space/time analysis.

Problem-Solving Workshops & Oral Proof Defenses (15%)
Eight structured Problem-Solving Workshops are integrated evenly across the semester (Lectures 3, 6, 9, 12, 15, 18, 21, and 24) to practice proof construction, reduction design, and lower-bound arguments. Additionally, students complete three 15-minute small-group Oral Proof Defenses (Lectures 7, 15, and 22) where they present and defend formal proofs of selected assignment problems before the instructor.

Midterm Examination (20%)
A closed-book written examination administered after Lecture 12 covering Lectures 1-12. The exam requires formal proofs and complexity derivations across decision-tree lower bounds, universal hashing, amortized potential functions, dynamic programming, greedy exchange arguments, and max-flow min-cut duality.

Algorithmic Implementation Labs (15%)
Students complete three compiled-language (C++ or Java) programming and empirical benchmarking labs, implementing advanced algorithms (e.g., lazy segment trees, Dinic's blocking flow/bipartite matching, and Fast Fourier Transform convolution) and submitting scaling analyses on large benchmark instances.

Final Examination (30%)
A comprehensive closed-book written examination emphasizing network flow extensions, linear programming strong duality, minimax zero-sum games, online competitive analysis, multiplicative weights updates, NP-completeness reductions, LP-rounding approximations, FFT, and streaming algorithms (Lectures 13-25), while integrating proof methodologies from the entire course.

Assessments: Activity	**Percent Contribution**
Written Proof & Problem Sets (6 sets)	20%
Problem-Solving Workshops (8) & Oral Proof Defenses (3)	15%
Algorithmic Implementation Labs (3 C++/Java benchmarks)	15%
Midterm Examination (closed-book)	20%
Final Examination (cumulative closed-book)	30%

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 4521 Schedule

Lecture	Topic	Textbook & Deliverables
L1	Algorithm Analysis Foundations, Recurrences, and Linear-Time Selection Topics: Asymptotic complexity, recurrence solving, randomized QuickSelect, and deterministic median-of-medians selection.	*K&T* 2.1-2.4, 5.1-5.2; *Erickson* Ch. 1
L2	Concrete Models of Computation and Information-Theoretic Lower Bounds Topics: Comparison sorting decision trees, adversary arguments, and query complexity lower bounds.	*K&T* 13.5; Lecture Notes (Decision-Tree & Adversary Lower Bounds)
L3	Randomized Hashing, Universal Hash Families, and String Fingerprinting Topics: Universal hash functions, expected collision bounds, and Karp-Rabin polynomial rolling hashes. Workshop 1: Lower-Bound Proofs & Probabilistic Collision Analysis	*K&T* 13.3, 13.6, 13.12; Lecture Notes (Rolling Fingerprinting)
L4	Range-Query Data Structures and Segment Trees with Lazy Propagation Topics: Augmented binary trees, static/dynamic interval aggregation, and lazy range updates.	*K&T* 2.5; Lecture Notes (Segment Trees & Range Queries) Problem Set 1 due
L5	Amortized Analysis I: Aggregate, Accounting, and Potential Methods Topics: Potential-function proofs, multipop stacks, binary counters, and dynamic table resizing.	Lecture Notes (Amortized Complexity & Potential Functions)
L6	Amortized Analysis II: Disjoint-Set Forests and Self-Adjusting Structures Topics: Union-by-rank, path compression, inverse Ackermann bounds, and self-adjusting search trees. Workshop 2: Designing Potential Functions & Lazy Tree Invariants	*K&T* 4.6; Lecture Notes (Union-Find & Splay Analysis)
L7	Dynamic Programming I: Weighted Intervals, Knapsack, and DAG Shortest Paths	*K&T* 3.6, 6.1-6.4, 6.8; *Erickson* Ch. 4

	Topics: Memoization vs. iterative DP, subset sums, 0-1 knapsack, and shortest paths in DAGs. Oral Proof Defense 1	
L8	Dynamic Programming II: Interval, Sequence, and Tree Dynamic Programming Topics: RNA secondary structure (interval DP), sequence alignment, and independent sets on trees.	*K&T* 6.5-6.7, 10.2; *Erickson* Ch. 4 Problem Set 2 due
L9	Greedy Algorithms, Exchange Arguments, and Optimal Caching Topics: Interval scheduling, minimizing lateness, optimal offline caching, and Huffman prefix codes. Workshop 3: DP Subproblem Formulation & Exchange Proofs	*K&T* 4.1-4.3, 4.8; *Erickson* Ch. 5 Implementation Lab 1 due
L10	Network Flows I: Residual Networks, Augmenting Paths, and Max-Flow Min-Cut Topics: Ford-Fulkerson algorithm, max-flow min-cut theorem, and capacity-scaling augmenting paths.	*K&T* 7.1-7.3; *Erickson* Ch. 11
L11	Network Flows II: Preflow-Push, Bipartite Matching, and Disjoint Paths Topics: Preflow-push max-flow, bipartite matching reductions, and edge/vertex-disjoint paths.	*K&T* 7.4-7.6; *Erickson* Ch. 12
L12	Midterm Review & Workshop 4: Integrated Proof & Reduction Strategies Topics: Synthesis of lower bounds, amortized structures, DP, greedy exchange, and network flows.	Review of *K&T* Ch. 2-7, 13; *Erickson* Ch. 1, 4-5, 11-12 Problem Set 3 due
EXAM	**Midterm Examination**	Covers L1-12 (closed-book)
L13	Network Flows III: Circulations with Demands, Project Selection, and Min-Cost Matchings Topics: Demand circulations, image segmentation/project selection cuts, and minimum-cost matchings.	*K&T* 7.7-7.8, 7.10-7.13; *Erickson* Ch. 12
L14	Linear Programming I: Formulations, Polytope Geometry, and Simplex Topics: Linear optimization modeling, feasible polytopes, basic feasible solutions, and simplex pivoting.	Lecture Notes (Linear Programming & Simplex)
L15	Linear Programming II: Weak/Strong Duality and Complementary Slackness Topics: Dual LP construction, strong duality theorem, and flow/cut duality via linear programming. Workshop 5: Network Flow Modeling & Dual LP Construction / Oral Proof Defense 2	Lecture Notes (LP Duality & Complementary Slackness)
L16	Zero-Sum Matrix Games, Nash Equilibria, and Minimax Optimality Topics: Two-player zero-sum games, mixed strategies, Von Neumann's minimax theorem, and best-response dynamics.	*K&T* 12.7; Lecture Notes (Minimax & Zero-Sum Games) Problem Set 4 due
L17	Online Algorithms, Randomized Caching, and Competitive Analysis Topics: Online vs. offline adversaries, ski-rental, deterministic/randomized paging, and competitive ratios.	*K&T* 13.8; Lecture Notes (Competitive Analysis)
L18	Multiplicative Weights Update (MWU) Method and Learning with Experts	Lecture Notes (Multiplicative

	Topics: Prediction with expert advice, regret bounds, and approximate LP solving via MWU. Workshop 6: Competitive Ratio & Regret Bound Proofs	Weights Framework) Implementation Lab 2 due
L19	Computational Intractability I: Polynomial-Time Reductions and NP-Completeness Topics: Complexity classes P, NP, co-NP, SAT gadget reductions, and canonical NP-complete problems.	*K&T* 8.1-8.5, 8.9; *Erickson* Ch. 13
L20	Computational Intractability II: Graph Coloring, Numerical NP-Completeness, and PSPACE Topics: Graph coloring, subset/numerical reductions, and quantified Boolean games in PSPACE.	*K&T* 8.6-8.8, 9.1-9.3; *Erickson* Ch. 13 Problem Set 5 due
L21	Approximation Algorithms I: Greedy Bounds, Pricing Method, and Knapsack FPTAS Topics: Load balancing, center selection, greedy set cover, primal-dual pricing, and knapsack FPTAS. Workshop 7: Polynomial Reductions & Combinatorial Approximation Ratios	*K&T* 11.1-11.5, 11.8
L22	Approximation Algorithms II: LP Relaxation, Rounding, and Randomized MAX-3-SAT Topics: Deterministic and randomized LP rounding, integrality gaps, and expectation-based approximations. Oral Proof Defense 3	*K&T* 11.6-11.7, 13.3-13.4
L23	Algebraic Algorithms: Polynomials and the Fast Fourier Transform (FFT) Topics: Coefficient vs. point-value representations, complex roots of unity, and O(n log n) convolution.	*K&T* 5.5-5.6 Implementation Lab 3 due
L24	Randomized Min-Cut, Chernoff Bounds, and Sublinear-Space Streaming Topics: Karger's global min-cut, Chernoff tail bounds, and one-pass data stream sketching. Workshop 8: LP Rounding, Probabilistic Bounds & Streaming Analysis	*K&T* 13.2, 13.9-13.10; Lecture Notes (Streaming Sketches) Problem Set 6 due
L25	Unified Algorithmic Paradigms and Final Review Topics: Cross-cutting review of optimization duality, randomness, reductions, and approximation guarantees.	Course Wrap-Up
EXAM	**Final Examination**	Cumulative: Covers L1-25 (closed-book)

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.