The Hong Kong University of Science and Technology (Guangzhou)

UG Course Syllabus

[Course Title] Advanced Algorithms

[Course Code] DSAA3041

[No. of Credits] 3

[Any pre-/co-requisites] DSAA2043

Name: Shangqi Lu

Email: shangqilu@hkust-gz.edu.cn

Office Hours: Monday 10-11 am, E3-3F-308

Lecture: WeFr 1:30 PM - 2:50 PM, Rm 202, W4

Tutorial: Fr 3:00 PM - 3:50 PM Rm 227, E1

Course Description

This course introduces advanced algorithmic techniques, including amortized analysis, randomized algorithms, and approximation algorithms. Students will learn about advanced data structures and their applications, as well as advanced solutions for optimization problems like linear programming and network flow. The course emphasizes the design and analysis of these techniques while also covering problem hardness and tractability, providing a solid foundation in advanced algorithms.

Intended Learning Outcomes (ILOs)

By the end of this course, students should be able to:

1. Formulate complex computational problems and consider possible solutions.

2. Apply advanced algorithmic techniques to design efficient algorithms for computational problems.

3. Evaluate the computational complexity of a problem or algorithm.

4. Explain the functioning of complex algorithms to peers and experts.

Weekly Schedule and Weekly ILOs

Week	Topics	Weekly ILOs
1	Introduction and Complexity Analysis Review	ILO 1,3,4
2	Trees	ILO 1,2,3,4
3	Amortized Analysis	ILO 1,2,3,4
4	More on Hashing	ILO 1,2,3,4

5	Heuristic Search Algorithms	ILO 1,2,3,4
6	Network Flow	ILO 1,2,3,4
7	Bipartite Graph Matching	ILO 1,2,3,4
8	Mid-term Exam	ILO 1,2,3,4
9	Geometric Algorithms	ILO 1,2,3,4
10	Linear Programming	ILO 1,2,3,4
11	NP Hard and Proving Hardness	ILO 1,3,4
12	Approximation Algorithms I	ILO 1,2,3,4
13	Approximation Algorithms II	ILO 1,2,3,4

Assessment and Grading

This course will be assessed using criterion-referencing and grades will not be assigned using a curve. Detailed rubrics for each assignment are provided below, outlining the criteria used for evaluation.

Assessments:

Assessment Task	Contribution to Overall Course grade (%)	Due date
Mid-Term	30%	01/04/2026 (in class)*
Written Assignments	30%	TBD*
Final Examination	40%	TBD*

* Assessment marks for individual assessed tasks will be released within two weeks of the due date.

Mapping of Course ILOs to Assessment Tasks

Assessed Task	Mapped ILOs	Explanation
Mid-Term Examination	ILO1, ILO2, ILO3, ILO4	The mid-term examination is a closed-book, hand-written assessment focusing on students' conceptual understanding of advanced algorithmic techniques and their ability to analyze algorithmic complexity (ILOs 1–3). Problems require students to formulate algorithmic solutions, justify correctness, and explain key ideas clearly and rigorously (ILO 4), emphasizing theoretical reasoning rather than implementation details.
Written assignments	ILO1, ILO2, ILO3, ILO4	Written assignments consist of a small number of carefully designed tasks, including both hand-written theoretical problem sets and limited programming assignments. Hand-written assignments emphasize problem formulation, algorithm design, proofs of correctness, and complexity analysis (ILOs 1–3). Programming components, when included, are used to reinforce

		algorithmic ideas rather than software engineering skills, and require students to explain design choices and analytical results (ILO 4).
Final Examination	ILO1, ILO2, ILO3, ILO4	The final examination is a comprehensive, hand-written assessment covering the full scope of the course, including amortized analysis, randomized algorithms, network flow, linear programming, NP-hardness, and approximation algorithms. Students are assessed on their ability to synthesize advanced techniques, rigorously analyze computational complexity, and clearly communicate algorithmic reasoning and insights (ILOs 1–4).

Grading Rubrics

[Detailed rubrics for each assignment will be provided. These rubrics clearly outline the criteria used for evaluation. Students can refer to these rubrics to understand how their work will be assessed.]

Written Assignments Grading Rubrics

Criteria	**Mapped ILOs**	**Expectation**
Problem Formulation and Algorithm Design	ILO 1, ILO 2, ILO 3	Demonstrate a clear and accurate formulation of computational problems. Design appropriate algorithms using advanced techniques (e.g., amortized analysis, randomization, reductions, linear programming, approximation). Solutions should reflect sound algorithmic intuition and correct use of theoretical tools.
Correctness and Complexity Analysis	ILO 2, ILO 3	Provide rigorous arguments for algorithm correctness and derive tight time and/or space complexity bounds. Analysis should be logically structured, mathematically sound, and clearly justified rather than heuristic or informal.
Clarity of Explanation and Reasoning	ILO 4	Clearly explain algorithmic ideas, design decisions, and analytical steps using precise language and well-organized reasoning. Written solutions should be readable, logically coherent, and suitable for peer or expert review.

Mid-term Examination Grading Rubrics

Criteria	**Mapped ILOs**	**Expectation**
Algorithmic Understanding and Application	ILO 1, ILO 2, ILO 3	Demonstrate a strong understanding of advanced algorithmic techniques covered in the course. Correctly apply these techniques to solve unseen problems, selecting appropriate approaches and justifying their use.
Correctness Reasoning and	ILO 2, ILO 3	Provide clear arguments for correctness and derive

Complexity Analysis		appropriate complexity bounds. Partial credit is awarded for correct reasoning even if final results are incomplete, while unsupported claims receive limited credit.
Explanation and Communication of Ideas	ILO4	Clearly present solutions, intermediate reasoning steps, and conclusions. Answers should be logically organized and concise, making algorithmic insights easy to follow under exam conditions.

Final Examination Grading Rubrics

Criteria	**Mapped ILOs**	**Expectation**
Algorithmic Understanding and Application	ILO 1, ILO 2, ILO 3	Demonstrate the ability to integrate multiple advanced techniques (e.g., reductions, LP formulations, approximation strategies, hardness arguments) to solve complex problems. Solutions should reflect a global understanding of the course rather than isolated topics.
Correctness Reasoning and Complexity Analysis	ILO 2, ILO 3	Provide correct, well-structured proofs or analytical arguments, including correctness, approximation guarantees (when applicable), and complexity bounds. Reasoning should be precise and logically complete.
Explanation and Communication of Ideas	ILO4	Communicate algorithmic reasoning clearly and effectively. Solutions should be well organized, with appropriate use of notation, diagrams (if helpful), and explanatory text.

Final Grade Descriptors:

This course adopts an absolute grading system, with final grades assigned according to the following score ranges:

A: [100,85]; B: (85,70]; C: (70,55]; D: (55,40]; F: (40,0].

As this course is being offered for the first time**,** the instructor reserves the right to make reasonable adjustments to the final grade boundaries**,** if necessary, based on overall student performance and with clear academic justification.

Grades	**Short Description**	**Elaboration on subject grading description**
A	Excellent Performance	The student demonstrates a deep and mature understanding of advanced algorithmic concepts and techniques. They consistently design correct and efficient algorithms, provide rigorous proofs or correctness arguments, and derive tight complexity or approximation bounds. Solutions are well-structured, logically sound, and clearly explained. The student is able to synthesize multiple techniques and articulate algorithmic

		ideas at a level suitable for peer or expert discussion.
B	Good Performance	The student shows a strong understanding of most advanced algorithmic topics covered in the course. They are able to design correct algorithms and provide mostly accurate complexity analyses, with only minor gaps in rigor or completeness. Explanations are generally clear, though some arguments may lack depth or polish. Overall performance reflects solid mastery with limited weaknesses.
C	Satisfactory Performance	The student demonstrates a basic understanding of core algorithmic techniques and can solve standard problems using appropriate methods. Correctness arguments and complexity analyses are present but may be incomplete, informal, or contain minor errors. Solutions are generally understandable but may lack clarity, rigor, or efficiency in algorithm design.
D	Marginal Pass	The student exhibits a limited and fragmented understanding of advanced algorithms. Solutions often rely on incomplete ideas, incorrect reasoning, or inappropriate techniques. Correctness arguments and complexity analyses are weak or largely missing. The student shows minimal ability to explain algorithmic ideas clearly, indicating insufficient preparedness in the subject.
F	Fail	The student demonstrates a fundamental lack of understanding of advanced algorithmic concepts. Solutions are largely incorrect or missing, with little evidence of valid algorithm design, correctness reasoning, or complexity analysis. Explanations, if provided, are unclear or incorrect, indicating a failure to meet the minimum learning outcomes of the course.

Course AI Policy

Generative AI tools may be used to support learning and self-study (e.g., reviewing concepts or exploring examples). However, the use of Generative AI in all assessed components of this course is prohibited, as early reliance on AI-generated content may hinder the development of a solid foundation in algorithmic reasoning and problem solving.

A. During examinations, the use of any other tools, including but not limited to Generative AI tools, internet resources, mathematical software, or electronic reference materials, is strictly prohibited.

B. For written assignments, the use of external tools (including Generative AI) is not explicitly prohibited. However, students are strongly encouraged to solve all problems independently. Working through the assignments on your own is essential for developing a deep understanding of the course material and for effective preparation for the mid-term and final examinations.

Communication and Feedback

Assessment marks for individual assessed tasks will be communicated via Canvas within two weeks of submission. Feedback on assignments will include breakdown marks and areas for improvement. Students who have further questions about the feedback including marks should consult the instructor within five working days after the feedback is received.

Resubmission Policy

If a student has an unforeseen, uncontrollable, and unavoidable reason, resubmitting work or reassessment opportunity will be considered. In such a case, the student is expected to contact the instructor immediately, but no later than within five working days, for further details.

Required Texts and Materials

The lecture notes for this course are self-contained and serve as the primary learning material.

For students interested in additional exercises, examples, or alternative perspectives on selected topics, the following textbooks are recommended:

- Introduction to Algorithms, by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein.
- Algorithm Design, by Jon Kleinberg and Éva Tardos.

All recommended materials are supplementary, and all examinable content will be covered in the lecture notes.

Academic Integrity

Students are expected to adhere to the university's academic integrity policy. Students are expected to uphold HKUST(GZ)'s Academic Honor Code and to maintain the highest standards of academic integrity. The University has zero tolerance of academic misconduct. Please refer to Regulations for Academic Integrity and Student Conduct for the University's definition of plagiarism and ways to avoid cheating and plagiarism.