The Hong Kong University of Science and Technology (Guangzhou)

UG Course Syllabus Template

Course Title: Introduction to Advanced Algorithmic Techniques

Course Code: AIAA4711

No. of Credits: 3

Prerequisite: DSAA2043

Name: Ge BAI

Email: gebai@hkust-gz.edu.cn

Office Hours: Every Friday 15:00-17:00, at W4 520

Course Description

The course aims to introduce advanced methods for designing and analyzing algorithms for solving tough problems. The topics cover useful advanced data structures, graph algorithms, heuristic searching algorithms and optimization algorithms that promote the development of AI. Also, students will learn algorithm design and analysis skills for computationally intractable problems, such as NP-completeness, randomized algorithms, approximation algorithms and amortized analysis. Additional topics, such as string matching, geometric and number-theoretic algorithms, will also be introduced. The course presents the topics at an introductory level and aims at senior undergraduate students. Through the class, students will develop a deeper understanding of algorithm design and demonstrate a basic understanding of the principles of advanced algorithms.

Methods of instruction: lectures, in-class tests, in-class presentation and visualization of individual projects, both written and programming assignments.

Intended Learning Outcomes (ILOs)

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

1. Know the rationale and basic design techniques within each of the paradigms.

2. Design new algorithms for concrete problems within each of the considered algorithm design paradigms using the covered algorithm design techniques.

3. Analyze the performance of algorithms in terms of time and space complexity.

4. Prove NP-hard problems.

5. Develop fundamental understanding on randomized and approximation algorithms.

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:

[List specific assessed tasks, exams, quizzes, their weightage, and due dates; perhaps, add a summary table as below, to precede the details for each assessment.]

Assessment Task	Contribution to Overall Course grade (%)	Due date
Mid-term examination	30%	13/03/2026
In-class test	10%	The day of the lecture
Assignments	20%	Depending on each assignment
Final examination	40%	To be announced

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

In-class tests: In-class problem solving sessions, no marks deducted for wrong answers. Should be handed in to the instructor in-class.

Assignments: approximately 15% paper-based problems, 35% online-judge problems, 50% individual projects.

Mid-term and final examinations: paper-based, focuses on conceptual understanding, algorithm analysis, and algorithm design (writing pseudocode).

Mapping of Course ILOs to Assessment Tasks

Assessed Task	Mapped ILOs	Explanation
In-class tests	ILO1, ILO3, ILO4, ILO5	In-class tests encourage continuous engagement, provide in-time feedback, and help students solidify core concepts (ILO1, ILO5) and practice basic analysis (ILO3) and proof techniques (ILO4).
Assignments	ILO1, ILO2, ILO3, ILO4, ILO5	Assignments contain both written analysis and programming. The written assignments check the understanding of concepts (ILO1, ILO5) and the ability to formalize analysis (ILO3) and proofs (ILO4). The programming assignments require students to have comprehensive understanding of concepts (ILO1), formulation of problems (ILO2), and analysis and optimization techniques (ILO3).
Mid-term examination	ILO1, ILO2, ILO3	The mid-term exam is a comprehensive assessment of the core concepts (ILO1, ILO5), algorithm design (ILO2) and analysis techniques (ILO3).

Final examination	ILO1, ILO2, ILO3, ILO4, ILO5	The final exam is a comprehensive assessment of the core concepts (ILO1, ILO5), algorithm design (ILO2) and analysis techniques (ILO3). It also includes mathematical proofs about algorithm analysis and problem hardness (ILO4).

Grading Rubrics

In-class tests rubrics

Criteria	**Excellent**	**Good**	**Satisfactory**	**Marginal**	**Fail**
Understanding of core concepts	The student demonstrates excellent and in-depth understanding of the concepts and the principles of the data structures and algorithms.	The student demonstrates good understanding of the concepts and the principles of the data structures and algorithms, without misunderstandings.	The student demonstrates satisfactory understanding of the concepts and the principles of the data structures and algorithms, with minor inaccuracies.	The student demonstrates limited understanding of the concepts and the principles of the data structures and algorithms, with errors and gaps in knowledge.	The student demonstrates little understanding of the concepts and the principles of the data structures and algorithms, with fundamental incorrectness or flaws.
Analysis and proof techniques	The student expertly applies the analysis and proof techniques to analyze problems and algorithms.	The student correctly applies the analysis and proof techniques to analyze problems and algorithms.	The student almost correctly applies the analysis and proof techniques to analyze problems and algorithms, with minor flaws.	The student applies incompletely the analysis and proof techniques to analyze problems and algorithms, with major flaws.	The student is unable to apply the techniques taught or apply them with fundamental errors.

Assignments rubrics

Criteria	**Excellent**	**Good**	**Satisfactory**	**Marginal**	**Fail**
Understanding of core concepts	The student demonstrates excellent and in-depth understanding of the concepts and the principles of the data	The student demonstrates good understanding of the concepts and the principles of the data structures and algorithms, without misunderstandings.	The student demonstrates satisfactory understanding of the concepts and the principles of the data structures and	The student demonstrates limited understanding of the concepts and the principles of the data structures and	The student demonstrates little understanding of the concepts and the principles of the data structures and

	structures and algorithms.		algorithms, with minor inaccuracies.	algorithms, with errors and gaps in knowledge.	algorithms, with fundamental incorrectness or flaws.
Analysis and proof techniques	The student expertly applies the analysis and proof techniques to analyze problems and algorithms.	The student correctly applies the analysis and proof techniques to analyze problems and algorithms.	The student almost correctly applies the analysis and proof techniques to analyze problems and algorithms, with minor flaws.	The student applies incompletely the analysis and proof techniques to analyze problems and algorithms, with major flaws.	The student is unable to apply the techniques taught or apply them with fundamental errors.
Problem-solving techniques	The student formalizes all the problems elegantly and precisely and provides solutions efficiently.	The student formalizes most of the problems precisely and provides solutions correctly.	The student formalizes simpler ones of the problems with moderate precision and provides solutions with minor errors.	The student is able to formalize the simplest problems with moderate precision and provides solutions with significant errors.	The student is unable to formalize problems or provide solutions.
Programming and optimization techniques	The student writes accurate and readable codes and well-optimized programs.	The student writes accurate codes with some readability and manages to optimize the program to some extent.	The student writes codes with minor flaws and little readability, and is unable to optimize the program.	The student writes codes with major flaws, and is unable to optimize the program.	The student is unable to write codes or writes codes with fundamental incorrectness.

Mid-term and final examinations rubrics

Criteria	**Excellent**	**Good**	**Satisfactory**	**Marginal**	**Fail**
Understanding of core concepts	The student demonstrates excellent and in-depth understanding of the concepts and the principles of the data structures and algorithms.	The student demonstrates good understanding of the concepts and the principles of the data structures and algorithms, without misunderstandings.	The student demonstrates satisfactory understanding of the concepts and the principles of the data structures and algorithms,	The student demonstrates limited understanding of the concepts and the principles of the data structures and algorithms, with errors	The student demonstrates little understanding of the concepts and the principles of the data structures and algorithms, with

			with minor inaccuracies.	and gaps in knowledge.	fundamental incorrectness or flaws.
Analysis and proof techniques	The student expertly applies the analysis and proof techniques to analyze problems and algorithms.	The student correctly applies the analysis and proof techniques to analyze problems and algorithms.	The student almost correctly applies the analysis and proof techniques to analyze problems and algorithms, with minor flaws.	The student applies incompletely the analysis and proof techniques to analyze problems and algorithms, with major flaws.	The student is unable to apply the techniques taught or applies them with fundamental errors.
Problem-solving techniques	The student formalizes all the problems elegantly and precisely and provides solutions efficiently.	The student formalizes most of the problems precisely and provides solutions correctly.	The student formalizes simpler ones of the problems with moderate precision and provides solutions with minor errors.	The student is able to formalize the simplest problems with moderate precision and provides solutions with significant errors.	The student is unable to formalize problems or provide solutions.

Final Grade Descriptors:

Grades	Short Description	Elaboration on subject grading description
A	Excellent Performance	Demonstrates a comprehensive understanding of advanced algorithm design and analysis techniques, and an expertise in formulating problems elegantly, writing efficient programs, and formalizing rigorous proofs.
B	Good Performance	Demonstrates a good understanding of advanced algorithm design and analysis techniques, competence in problem solving, and the ability to write efficient programs, analyze algorithms and formalize rigorous proofs.
C	Satisfactory Performance	Demonstrates an adequate understanding of the core algorithm design and analysis techniques, some capacity in problem solving, ability in programming with less efficiency or minor errors, some capacity in algorithm analysis and mathematical proofs.
D	Marginal Pass	Demonstrates a threshold understanding of the core concepts and techniques, ability in programming with less efficiency or minor errors, and difficulties in algorithm analysis and mathematical proofs.
F	Fail	Demonstrates insufficient understanding of the core concepts. Implementation of data structures and algorithms is inaccurate or fundamentally inefficient. Shows limited ability to solve problems, write programs, algorithm analysis and mathematical proofs.

Course AI Policy

The use of generative AI is partly permitted in completing programming assignments and individual projects of this course. It is allowed to use generative AI only for non-algorithmic parts of programming, such as graphical interface and use of libraries, and it is not allowed to finish any essential algorithm design directly using generative AI.

The direct use of generative AI is NOT allowed for paper-based assignments, in-class tests, mid-term examination or final examination.

This policy is because generative AI may produce answers without the user demonstrating their own understanding in algorithm design and analysis. However, it is partly permitted to complete tedious and distracting coding work that is non-essential for this course.

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

Each assignment can only be assessed once after the submission deadline. For students who seek clarification regarding the scores, they should communicate with the instructor within five working days after the feedback is received.

The decision to grant a reassessment or resubmission opportunity rests solely with the instructor. This discretion allows the instructor to consider the individual circumstances surrounding a student's request, and the policies of the university.

Required Texts and Materials

Textbooks:

Introduction to Algorithms. Cormen, Leiserson, Rivest, and Stein (For most course content)

Algorithm Design. Kleinberg and Tardos (For the following topics: union-find data structure, combinatorial search, graph algorithms)

Other references:

The Algorithm Design Manual. Steven Skiena

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.

Additional Resources

An online introduction to many advanced algorithms: https://cp-algorithms.com/

Course Schedule:

Week	Date	Topic	Content	Course ILO
1	Jan 23	Introduction	Course overview Lower bound of comparison sorting Radix sort	ILO1, ILO3
2	Jan 30	Trees	Recap: binary search tree, heaps, sets and dictionaries Interval tree k-d tree and Barnes-Hut simulation	ILO1, ILO2
3	Feb 6	Amortized analysis	Recap: time complexity Accounting method and potential method Dynamic arrays Union-find sets	ILO1, ILO3
4	Feb 27	Search and optimization algorithms	Recap: breadth-first and depth-first search Recap: dynamic programming Combinatorial search Heuristic search, A* algorithm Game theoretic search Pruning	ILO1, ILO2, ILO3
5	Mar 6	Graph Algorithms I & II	Recap: shortest path Heuristic shortest path Network flow and min-cut Matching	ILO1, ILO2, ILO3
6	Mar 13	Review and Mid-term test		ILO1, ILO2, ILO3
7	Mar 20	Geometric Algorithms	Matrices Convex hull Closest pair of points Collision tests	ILO1, ILO2, ILO3
8	Mar 27	Polynomials and the FFT	Fast Fourier transform Image and audio processing	ILO1, ILO2, ILO3
9	Apr 3	Number Theoretic Algorithms	Recap: Euclid algorithm Extended Euclid algorithm Prime sieve, prime test	ILO1, ILO3
10	Apr 10	String Matching	String matching algorithm Deterministic finite automata Regular expressions	ILO1, ILO2, ILO3
11	Apr 17	NP-Completeness	Computational models SAT and 3-SAT problem Polynomial reductions Set cover, knapsack problem, independent sets	ILO1, ILO3, ILO4
12	Apr 24	Probabilistic and approximation algorithms I	Probabilistic analysis Monte Carlo method	ILO1, ILO2, ILO3, ILO5

13	May 8	Probabilistic and approximation algorithms II	Approximate optimization problems Integer programming and linear programming	
15-16	TBA	Final exam		ILO1, ILO2, ILO3, ILO4, ILO5