The Hong Kong University of Science and Technology (Guangzhou)

UG Course Syllabus

Design and Analysis of Algorithms

DSAA2043

3 Credits

Prerequisites: UFUG 2602

Name: Yanlin Zhang

Email: yanlinzhang@hkust-gz.edu.cn

Office Hours: Tuesday 10:00-11:00pm, E1-5F-511

Course Description

Design and Analysis of Algorithms is an important course that bridges students to a number of advanced courses in data science and analytics. This course introduces core data structures and algorithms. It covers advanced asymptotic complexity analysis, introduces common algorithmic paradigms (e.g., divide-and-conquer, greedy, and dynamic programming), a collection of classic algorithms (e.g., graph algorithms) and introduces the computational complexity theory. The course employs a range of assessment methods, including individual projects, coding exercises and closed-book exams, to foster both theoretical and practical foundation.

Intended Learning Outcomes (ILOs)

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

1. Demonstrate a good understanding of a wide variety of data structures and algorithms concepts and techniques.

2. Demonstrate the comprehension of analytical methods for modelling the complexity of data structure and algorithms.

3. Apply appropriate data structure and algorithms to solve computational problems effectively.

4. Apply algorithmic design techniques to solve computational problems effectively.

5. Implement and evaluate the performance of different data structures and 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:

Assessment Task	**Contribution to Overall Course grade (%)**	**Due date**
Class Participation	5%	In class
Written assignment	15%	several due dates*
Midterm test	30%	Week 8*
Final Exam	40%	TBD*
Project Report	10%	Week 9* & Week 13*

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**
Class Participation	ILO1, ILO2, ILO3	Class participation encourages continuous engagement, fosters critical thinking through active discussion, and helps students solidify core concepts. By attending classes, asking and answering questions, discussing algorithmic strategies, and reflecting on peers' insights, students reinforce foundational knowledge (e.g., ILO-1, ILO-2), and begin to develop problem-solving techniques (e.g., ILO-3).
Lab exercises	ILO1, ILO2, ILO3, ILO4, ILO5	Lab exercises, encompassing both written analysis and programming, effectively map to all ILOs: they require students to demonstrate a comprehensive understanding of data structures and algorithms (ILO-1), apply analytical methods to model complexity (ILO-2), and apply and design algorithms to solve problems (ILO-3, ILO-4). Additionally, by implementing and evaluating the performance of these structures and algorithms (ILO-5), students showcase their ability to translate theory into practice and analyze efficiency, thus addressing the full spectrum of intended learning outcomes through hands-on application and critical analysis.
Individual Project	ILO1, ILO2, ILO3, ILO4, ILO5	The individual project is a capstone assessment that aligns with all the intended learning outcomes (ILOs). It requires students to independently demonstrate a deep understanding of various data structures and algorithms (ILO-1), apply analytical methods to model algorithmic complexity (ILO-2), and effectively apply and design algorithms to solve computational problems (ILO-3, ILO-4). Additionally, the project involves implementing these solutions and critically evaluating their performance, thereby addressing ILO-5 and ensuring a comprehensive assessment of the students' knowledge, skills, and ability to apply theoretical concepts in practical scenarios.
Mid-term Examination	ILO-1, ILO-2, ILO-3, ILO-4, ILO-5	The mid-term exam is a comprehensive assessment that evaluates students' understanding of data structures and algorithms concepts (ILO-1), analytical methods for modeling complexity (ILO-2), and their ability to apply and design algorithms to solve computational problems (ILO-3 and ILO-4). Additionally, the exam may include questions on implementing and evaluating the performance of different data structures and algorithms (ILO-5), ensuring a well-rounded assessment of all intended learning outcomes.
Final examination	ILO1, ILO2, ILO3, ILO4, ILO5	The final exam is a comprehensive assessment that evaluates students' understanding of data structures and algorithms concepts (ILO-1), analytical methods for modeling complexity (ILO-2), and their ability to apply and design algorithms to solve computational problems (ILO-3 and ILO-4). Additionally, the exam may include questions on implementing and evaluating the performance of different data structures and algorithms (ILO-5), ensuring a well-rounded assessment of all intended learning outcomes.

Grading Rubrics

Lab Exercises Student Rubric

Criteria	**Excellent**	**Good**	**Satisfactory**	**Marginal**	**Fail**	**Mapping to ILOs**
Understanding of Concepts	Demonstrates a comprehensive understanding of data structures, providing clear and accurate written analysis	Shows a strong understanding with minor gaps, and written analysis is mostly clear and accurate.	Demonstrates a basic understanding, with some errors in written analysis.	Shows a limited understanding, with significant errors in analysis.	Lacks basic understanding and provides little to no analysis.	ILO-1, ILO-2
Application of Algorithms in Programming	Effectively applies algorithms in programming tasks, demonstrating high-level problem-solving skills.	Applies algorithms competently, with good problem-solving skills and minor optimizations.	Uses algorithms to complete tasks, but with some inefficiencies or less optimal solutions.	Struggles to apply algorithms in programming, often resulting in incorrect or inefficient code.	Unable to apply algorithms to complete programming tasks	ILO-3, ILO-4, ILO-5

Individual Project Student Rubric

Criteria	**Excellent**	**Good**	**Satisfactory**	**Marginal**	**Fail**	**Mapping to ILOs**
Understanding of data structures and algorithms	The student demonstrates exceptional knowledge and understanding of a wide range of data structures and algorithms. They can explain subtle differences and advantages in different contexts with evidence of deep insight.	The student shows a strong grasp of various data structures and algorithms, with the ability to discuss their uses and limitations correctly.	The student displays an adequate understanding of the fundamental data structures and algorithms, with some minor inaccuracies.	The student has a superficial understanding of data structures and algorithms, with evident gaps in knowledge.	The student demonstrates a lack of understanding of basic data structures and algorithms, with fundamental misconceptions.	ILO-1
Analytical methods to model algorithmic complexity	The student expertly applies advanced analytical methods to model algorithmic complexity, providing precise calculations and thorough explanations.	The student correctly uses analytical methods to model algorithmic complexity with minor errors or omissions in calculations.	The student demonstrates a basic ability to apply analytical methods to model algorithmic complexity, with some errors.	The student struggles with analytical methods and often incorrectly models algorithmic complexity.	The student is unable to apply analytical methods to model algorithmic complexity accurately.	ILO-2
Application and design of algorithms to solve computational problems	The student designs and applies algorithms with exceptional proficiency, solving complex problems efficiently and effectively.	The student applies and designs functional algorithms that solve problems correctly, though not always optimally.	The student solves problems using standard algorithms, with limited efficiency and little innovation.	The student has difficulty applying or designing algorithms to solve problems effectively.	The student fails to apply or design functional algorithms, resulting in unsolved or incorrectly solved problems.	ILO-3, ILO-4
Implementation and critical evaluation of solutions	The student implements solutions with outstanding skill and provides an in-depth, critical evaluation of their performance, considering various scenarios and edge cases.	The student implements solutions with competence and provides a solid critical evaluation of their performance, identifying strengths and weaknesses.	The student's implementation of solutions is adequate, with a basic evaluation of their performance that may overlook some issues.	The student's implementation has significant flaws, and their evaluation of performance lacks depth and critical analysis.	The student fails to implement solutions correctly and does not provide a meaningful evaluation of their performance.	ILO-5

Final Examination Student Rubric

Criteria	**Excellent**	**Good**	**Satisfactory**	**Marginal**	**Fail**	**Mapping to ILOs**
Understanding of Algorithm Concepts	Demonstrates a comprehensive and nuanced understanding of all algorithm concepts covered in the course.	Demonstrates a strong understanding of most algorithm concepts with a high level of detail.	Demonstrates a clear understanding of the basic algorithm concepts.	Shows a limited understanding of algorithm concepts with gaps in knowledge.	Demonstrates no understanding or a fundamental misunderstanding of algorithm concepts.	ILO-1, ILO-2
Problem-Solving Skills	Solves complex algorithm problems with ease, applying concepts creatively and efficiently.	Solves most algorithm problems effectively, applying concepts correctly.	Solves simpler algorithm problems with some guidance, applying concepts adequately.	Struggles with problem-solving, making significant errors in approach or execution.	Unable to solve algorithm problems or applies concepts incorrectly.	ILO-1, ILO-2, ILO-3, ILO-4
Correctness and Accuracy	All algorithm solutions are correct, efficient, and demonstrate a deep understanding of the problem.	Solutions are mostly correct with minor inefficiencies or errors.	Solutions are generally correct but may lack efficiency or contain minor errors.	Solutions contain several significant errors and are inefficient.	Solutions are predominantly incorrect or do not address the problem.	ILO-3, ILO-4, ILO-5
Algorithm Analysis	Provides thorough and accurate analysis of algorithm complexity (time and space).	Provides mostly accurate analysis of algorithm complexity with some minor omissions.	Provides basic analysis of algorithm complexity with some inaccuracies.	Provides incomplete or inaccurate analysis of algorithm complexity.	Does not attempt or provides a completely incorrect analysis of algorithm complexity.	ILO-3, ILO-4, ILO-5

Final Grade Descriptors:

Grades	Short Description	Elaboration on subject grading description
A	Excellent Performance	The student exhibits a mastery of data structures and algorithms, applying them expertly in lab exercises with insightful analysis. Their understanding of algorithmic complexity is profound, and their programming solutions are both efficient and innovative. Code implementation is top-notch, with a focus on best practices and performance optimization.
B	Good Performance	The student demonstrates a solid understanding of data structures and algorithms, applying them effectively in the lab with mostly accurate analysis. Their comprehension of algorithmic complexity is clear, and programming solutions are generally efficient. The implementation of data structures and algorithms is proficient, with minor issues in documentation or optimization.
C	Satisfactory Performance	Shows a basic understanding of data structures and algorithms, capable of completing lab tasks with some errors in analysis. Their grasp of algorithmic complexity is adequate, and programming solutions are functional but may lack efficiency. Implementation is competent, though with notable inefficiencies or lack of documentation.
D	Marginal Pass	Exhibits a limited understanding of data structures and algorithms, with significant errors in lab analysis and implementation. Their approach to algorithmic complexity is shaky, and programming solutions are inconsistent, often inefficient. Code may be incomplete or contain substantial errors, indicating a need for greater proficiency.
F	Fail	Demonstrates a lack of understanding of data structures and algorithms, with frequent and fundamental errors in lab exercises and written analysis. Their grasp of algorithmic complexity is inadequate, and programming solutions are either incorrect or non-existent. Implementation of data structures and algorithms is ineffective, showing a clear deficiency in skills.

Course AI Policy

The use of Generative AI is permitted in this course, students should obey policy in each task whenever a specific generative AI policy is applied.

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

N/A

Required Texts and Materials

Textbook:

Introduction to Algorithms. Cormen, Leiserson, Rivest, and Stein

Reference books:

Algorithm Design. Kleinberg and Tardos

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.

Course Outline

Week	Topic	Content (if any)
1	Introduction	Review of basic data structure and asymptotic complexity
2	Preliminaries	Loop invariant in algorithms (Insertion sort, bubble sort, KMP algorithm, etc.)
3	Sorting algorithms	Divide and conquer algorithms (Merge sort, quick sort)
4	Dynamic Programming I	Knapsack Problem, etc.
5	Dynamic Programming II	Longest Common Subsequence, backtracking, etc.
6	Greedy algorithms	Scheduling, MST
7	Advanced algorithms I	Algorithms on Strings
8	Advanced algorithms I and Midterm Exam	
9	Advanced algorithms II	Algorithms on Trees
10	Graph algorithms I	Graph representation, Graph traversal, Topological sorting, etc.
11	Graph algorithms II	Shortest path algorithms, etc.
12	Graph algorithms III	Network Flow, etc.
13	Complexity Theory	P/NP, NP-complete, etc.
-	Final exam	