The Hong Kong University of Science and Technology

UG Course Syllabus (Spring 2025-26)

Course Title: Honors Design and Analysis of Algorithms

Course Code: COMP 3711H

No. of Credits: 4-credit

Any pre-/co-requisites: (Grade B+ or above in COMP 2011 / COMP 2012 / COMP 2012H) AND (Grade A- or above in COMP 2711 / COMP 2711H / MATH 2343) Exclusion: COMP3711

Name: Sunil ARYA

Email: arya@cse.ust.hk

Course Description

Techniques for designing algorithms, proving their correctness, and analyzing their running times. Topics covered include: sorting, selection, heaps, balanced search trees, divide-and-conquer, greedy algorithms, dynamic programming, and graph algorithms. The class will also provide an introduction to advanced techniques such as amortized analysis and the design of randomized and approximation algorithms, as well as providing exposure to more advanced algorithmic solutions to optimization problems, e.g. linear programming and network flow.

Intended Learning Outcomes (ILOs)

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

1. Describe fundamental concepts and techniques for determining the asymptotic behavior of real-valued functions defined in natural numbers.
2. Explain recurrence equations and solve common recurrences using a variety of techniques.
3. Analyze an algorithm described in plain language or some form of pseudocode in terms of its time (or space) efficiency as a function of the size of a problem instance.
4. Explain how various data structures, including trees, heaps and disjoint set structures, influence the time efficiency of algorithms.
5. Apply general algorithmic design and analysis techniques to solving problems, including greedy, divide-and-conquer and dynamic programming.
6. Identify randomization in algorithms and analyze basic randomized algorithms such as randomized quicksort and selection.

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	35%	TBA
Assignment 1	5%	4/03/2026
Assignment 2	5%	18/03/2026
Assignment 3	5%	22/04/2026
Assignment 4	5%	6/05/2026
Final examination	45%	TBA

* 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
Assignments, midterm, final examination	ILO1, ILO2, ILO3. ILO4, ILO5, ILO6	Each assigned task will assess students' ability to analyze problems, describe the algorithms and data structures for solving the problems, and analyze the performance guarantees of the proposed solutions. So different aspects of ILO1 to ILO5 are involved. Certain ILOs such as ILO4, ILO5 and ILO6 concern with specific types of problems, and they will be assessed in assignments, midterms, and/or final examinations after the corresponding topics have been covered in class.

Grading Rubrics

Course Learning Outcome	Exemplary	Competent	Needs Work	Unsatisfactory
1. Describe fundamental concepts and techniques for determining the asymptotic behavior of real-valued functions defined in natural numbers.	Demonstrates thorough understanding of the definitions and usage of asymptotic analysis of real-valued functions. Is able to use them to describe the running time of very complicated algorithms.	Demonstrates sufficient understanding of the definitions and usage of asymptotic analysis of real-valued functions. Is able to use them to describe the running time of basic algorithms.	Demonstrates some preliminary understanding of the definitions and usage of asymptotic analysis of real-valued functions.	Demonstrates deficient understanding of the definitions and/or usage of asymptotic analysis of real-valued functions.
2. Explain recurrence equations and solve common recurrences using a variety of techniques.	Demonstrates thorough understanding of recurrence relations and how to solve recurrences that are major modifications of the common varieties taught in class.	Demonstrates sufficient understanding of recurrence relations and how to solve recurrences that are slight modifications of the common varieties taught in class.	Demonstrates basic understanding of what a recurrence relation models and how to solve simple recurrence relations.	Demonstrates deficient knowledge of how to formulate and solve even basic recurrence equations.
3. Analyze an algorithm described in plain language or some form of pseudocode in terms of its time (or space) efficiency as a function of the size	Demonstrates ability to fully analyze the efficiency of complicated algorithms described in plain-language/pseudocode.	Demonstrates sufficient ability to fully analyze the efficiency of simple algorithms described in plain-language/pseudocode and at least partially analyze more complicated algorithms.	Demonstrates preliminary understanding of how to analyze efficiency of an algorithm described in plain-language/pseudocode.	Does not understand how to translate a plain-language/pseudocode description of an algorithm into a format amenable to analysis

of a problem instance				
4. Explain how various data structures, including trees, heaps and disjoint set structures, influence the time efficiency of algorithms.	Demonstrates strong understanding of the influence of the usage of data structures on the time efficiency of algorithms. Is usually able to use this knowledge to formulate accurate equations describing this time efficiency	Demonstrates sufficient understanding of the influence of the usage of data structures on the time efficiency of algorithms. Is often able to use this knowledge to formulate accurate equations describing this time efficiency.	Demonstrates preliminary understanding of the direct influence of the usage of various data structures on the time efficiency of algorithms.	Has very limited understanding of the influence of the usage of data structures on the time efficiency of algorithms.
5. Apply general algorithmic design and analysis techniques to solving problems, including greedy, divide-and-conquer and dynamic programming.	Is usually able to apply general algorithmic design and analysis techniques to deriving the most efficient algorithms to solve problems	Is usually able to apply general algorithmic design and analysis techniques to deriving efficient algorithms to solve problems	Is sometimes unable to apply general algorithmic design and analysis techniques to deriving efficient algorithms to solve simple problems.	Is usually unable to apply general algorithmic design and analysis techniques to deriving efficient algorithms to solve problems.
6. Identify randomization in algorithms and analyze basic randomized algorithms such as randomized quicksort and selection.	Is able to identify randomization in algorithms and to analyze more complicated randomized algorithms	Is able to identify randomization in algorithms and to usually analyze basic randomized algorithms	Is able to identify randomization in algorithms and to sometimes analyze basic randomized algorithms.	Is often unable to identify randomization in algorithms and to analyze basic randomized algorithms

Final Grade Descriptors:

Grades	Short Description	Elaboration on subject grading description
A	Excellent Performance	Demonstrate a comprehensive understanding of algorithm design techniques. Demonstrate a high-level of expertise in problem-solving and performance analysis. Exhibits a high capacity for scholarship, going beyond core requirements to achieve learning goals.
B	Good Performance	Show good understanding of algorithm design techniques. Demonstrate a good mastery of skills in problem-solving and performance analysis.
C	Satisfactory Performance	Possess adequate knowledge of algorithm design techniques. Demonstrate a satisfactory level of skills in problem-solving and performance analysis.
D	Marginal Pass	Demonstrate a basic level of understanding of algorithm design techniques. Demonstrate a basic level of skills in problem-solving and performance analysis.
F	Fail	Do not possess a basic level of understanding of algorithm design techniques. Do not demonstrate a basic level of skills in problem-solving and performance analysis.

Course AI Policy

The use of Generative AI is permitted with proper acknowledgment. However, the student needs to write up his or her own solution to avoid plagiarism issues.

Communication and Feedback

Assessment marks for individual assessed tasks will be communicated via Canvas within two weeks of submission. Students who have further questions about the feedback including marks should consult the teaching assistant or instructor.

Resubmission Policy

Late submission of assignments will not be accepted unless a prior approval has been given by the instructor. No make-ups will be given for the midterm or final exam unless prior approval is granted by the instructor, or you are in an unfavorable medical condition with a physician's documentation for the day of the midterm or final exam.

Academic Integrity

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

Required Texts and Materials

[List required textbooks, readings, and any other materials]

Textbooks

- Algorithms by Dasgupta, Papadimitriou, and Vazirani, McGraw-Hill.
- Algorithm Design by Kleinberg and Tardos, Addison-Wesley.

Additional Resources

N/A