UNSW Course Outline

COMP9101 Algorithm Design and Analysis - 2024

Published on the 23 May 2024

General Course Information

Course Code : COMP9101
Year : 2024
Term : Term 2
Teaching Period : T2
Is a multi-term course? : No
Faculty : Faculty of Engineering
Academic Unit : School of Computer Science and Engineering
Delivery Mode : In Person
Delivery Format : Standard
Delivery Location : Kensington
Campus : Sydney
Study Level : Postgraduate
Units of Credit : 6

Useful Links

Handbook Class Timetable

Course Details & Outcomes

Course Description

How would you convince a colleague that your program is correct, and that theirs is flawed?

How do you estimate how long a program will run for, and design test cases to find bugs?

Can all problems be solved efficiently, or are some problems "too hard"?

In this course, you will learn the building blocks to develop problem-solving software in fields as diverse as finance, logistics, policy and entertainment. You will apply techniques including divide-and-conquer, greedy selection and dynamic programming in order to develop accurate and fast algorithms in contexts such as graphs and strings. You will also develop the ability to think critically and communicate about algorithms in terms of correctness and efficiency.

Join us to find out how you can become a better programmer without writing any code.

Course Aims

In this course, students will be introduced to a variety of algorithm design techniques (greedy, dynamic programming, divide and conquer, etc), and most importantly learn how to apply them in different settings.

This course aims to develop students' theoretical knowledge in order to design correct and efficient software, as well as problem solving, critical thinking and written communication skills.

By understanding algorithm design principles, analysing and evaluating algorithms, and applying these principles to solve unfamiliar problems, students will become more capable and responsible problem solvers.

Relationship to Other Courses

This course extends COMP9024 Data Structures and Algorithms and COMP9020 Foundations of Computer Science, and prepares students for further study in algorithms and theoretical computer science, including COMP4121 Advanced Algorithms, COMP4128 Programming Challenges, COMP4141 Theory of Computation and COMP6741 Algorithms for Intractable Problems. Proficiency in LaTeX is also developed in order to help Honours students write a thesis.

Students with an interest in computer science research should consider taking COMP9801 Extended Algorithm Design and Analysis, which covers more theoretical topics and in greater depth.

Course Learning Outcomes

Course Learning Outcomes
CLO1 : Explain how standard design techniques are used to develop algorithms
CLO2 : Solve problems by creatively applying algorithm design techniques
CLO3 : Communicate algorithmic ideas at different abstraction levels
CLO4 : Evaluate the efficiency of algorithms and justify their correctness
CLO5 : Apply the LaTeX typesetting system to produce high-quality technical documents

Course Learning Outcomes	Assessment Item
CLO1 : Explain how standard design techniques are used to develop algorithms	• Portfolio • Final Exam
CLO2 : Solve problems by creatively applying algorithm design techniques	• Portfolio • Final Exam
CLO3 : Communicate algorithmic ideas at different abstraction levels	• Portfolio • Final Exam
CLO4 : Evaluate the efficiency of algorithms and justify their correctness	• Portfolio • Final Exam
CLO5 : Apply the LaTeX typesetting system to produce high-quality technical documents	• Portfolio • Final Exam

Learning and Teaching Technologies

Moodle - Learning Management System | EdStem | Formatif | Echo 360 | Microsoft Teams

Learning and Teaching in this course

Lectures will be recorded on Echo360. Where capacity permits, students enrolled in the WEB stream are welcome to attend in person.

The tutorial will function as a recitation class, aimed to revise and reinforce lecture content, and exhibit how to solve standard problems. Students are of course welcome to ask questions, and may be asked to interact in pairs or small groups.

All lab classes will be held in person. Students are encouraged to attend their timetabled lab classes, and in particular must get checkpoint tasks marked off during class. Students may change their lab group in Formatif until week 4 if capacity permits.

You do not need to inform anyone if you will be missing a class, and you do not need to apply for

Special Consideration.

Consultation will be held from 3pm to 4pm every Monday and Thursday at K17 504.

Additional consultations will be scheduled for the exam period.

Help sessions will be held

- Monday 6pm to 8pm
- Thursday 2pm to 4pm
- Friday 11am to 1pm
- Friday 4pm to 6pm,

at K17 G05 (CSE Help).

Separate arrangements will be made for help sessions in weeks 6 and 11.

Assessments

Assessment Structure

Assessment Item	Weight	Relevant Dates
Portfolio Assessment Format: Individual	60%	Start Date: Week 0 Due Date: Week 11: 05 August - 11 August
Final Exam Assessment Format: Individual	40%	Start Date: TBA during exam period Due Date: TBA during exam period

Assessment Details

Portfolio

Assessment Overview

Portfolio consists of responses to a collection of formative tasks completed during the term.

Tasks will be made available upon the commencement of each module, using contract grading. Individual written or audio feedback will be provided promptly upon submission (in labs or online), after which the student can resubmit until the task is complete or the deadline expires.

Students are expected to complete all (or almost all) tasks up to the grade they have contracted for in order to earn that grade, with more open-ended and extension tasks to distinguish between the highest achievers. The final portfolio will be evaluated using delayed summative assessment,

with input from the student's tutor and a formula to quantify tasks completed.

Course Learning Outcomes

- CLO1 : Explain how standard design techniques are used to develop algorithms
- CLO2 : Solve problems by creatively applying algorithm design techniques
- CLO3 : Communicate algorithmic ideas at different abstraction levels
- CLO4 : Evaluate the efficiency of algorithms and justify their correctness
- CLO5 : Apply the LaTeX typesetting system to produce high-quality technical documents

Detailed Assessment Description

There are four types of tasks, with different submission requirements:

- Discussion tasks (D) require you to initiate a text conversation with their instructor in the sidebar.
- Moodle tasks (M) require you to complete a learning activity (usually a quiz) on Moodle.
- Regular tasks (R) require you to submit a PDF document. You are welcome to use the LaTeX template provided in the 'Task Resources' section, but you can instead use any other method, such as a word processor or clear handwriting.
- LaTeX tasks (L) require you to submit a PDF document *and* the LaTeX source code used to produce it.

Certain tasks are also designated as checkpoint tasks (C). To get a checkpoint task marked as complete, you must have either:

- discussed the task with an instructor during a lab, or
- completed another checkpoint task in the same module and grade level.

The listed due dates are the last opportunity to receive feedback on a task. You can submit after the due date, but you will not receive further rounds of feedback, so it is solely your responsibility to complete the task to the required standard. Short extensions (one week) can be requested within the Formatif system, so Special Consideration is only required for long periods of impact.

Assessment information

You can change your grade contract in Formatif at any time, for example if you want a greater challenge or if you are falling behind. You can also apply for an extension on any task with brief reasoning.

The first module (Foundations) will have an unusually large number of tasks, as this includes revision of prerequisite material.

You may make reference to published course material (e.g. lecture slides, tutorial solutions) without providing a formal citation. The same applies to material from prerequisite courses. You

may make reference to either of the recommended textbooks with a citation in any format. You may reproduce material from external sources in your own words, along with a citation in any format. You may discuss the assignment problems with other students, so long as you acknowledge them in a citation. All writing must be your own, with the exception of translation of words or short phrases; editing and translation using generative AI are forbidden. Please review the UNSW plagiarism policy.

Assignment submission Turnitin type

Not Applicable

Final Exam

Assessment Overview

The final examination (2 hours during UNSW exam period) tests critical thinking and general understanding of the course material, in addition to the application of algorithm design techniques to analyse algorithms and solve unseen problems.

Marking will be completed using a rubric.

Course Learning Outcomes

- CLO1 : Explain how standard design techniques are used to develop algorithms
- CLO2 : Solve problems by creatively applying algorithm design techniques
- CLO3 : Communicate algorithmic ideas at different abstraction levels
- CLO4 : Evaluate the efficiency of algorithms and justify their correctness
- CLO5 : Apply the LaTeX typesetting system to produce high-quality technical documents

Detailed Assessment Description

The final exam will be held on campus using INSPERA, under invigilation. Students will have to bring their own laptop with the Safe Exam Browser installed. Information and resources are available at https://www.student.unsw.edu.au/exams/inspera/on-campus, including the UNSW supported version of Safe Exam Browser.

The exam will include multiple choice, short answer and extended response questions.

Assignment submission Turnitin type

Not Applicable

Hurdle rules

Students must demonstrate individual attainment of the courrse learning outcomes by achieving an exam mark of at least 40%. Students who do not meet the hurdle requirement will receive UF

grade.

General Assessment Information

Up to 5 bonus marks will be awarded for contribution to other students' learning. This is awarded on the basis of constructive participation in lecture, tutorial and lab classes, as well as activity (including anonymous activity) on the Ed forum. We typically award at least one bonus mark to 3-5% of students, and about half as many students receive each additional mark.

Grading Basis

Standard

Requirements to pass course

To pass the course, students must achieve a total mark of at least 50 out of 100, and pass the hurdle requirement in the final exam.

Course Schedule

Teaching Week/Module	Activity Type	Content
Week 1 : 27 May - 2 June	Lecture	Monday: Welcome (vision, introduction to algorithm design and analysis, course overview, administration)
	Lecture	Thursday: Foundations (time complexity, revision of data structures and algorithms, revision of proof techniques)
	Tutorial	Foundations
Week 2 : 3 June - 9 June	Lecture	Monday: Divide and Conquer I (recursive problem solving, binary search, merge sort, quicksort, recursion on trees)
	Lecture	Thursday: Divide and Conquer II (analysing correctness and efficiency of divide and conquer algorithms, fast multiplication of integers)
	Tutorial	Foundations, Divide and Conquer
Week 3 : 10 June - 16 June	Lecture	Thursday: Divide and Conquer III (convolutions and the Fast Fourier Transform)
	Tutorial	Divide and Conquer
Week 4 : 17 June - 23 June	Lecture	Monday: Greedy Algorithms I (greedy paradigm, optimal selection)
	Lecture	Thursday: Greedy Algorithms II (optimal ordering, optimal merging)
	Tutorial	Greedy Algorithms
Week 5 : 24 June - 30 June	Lecture	Monday: Greedy Algorithms III (strongly connected components, directed acyclic graphs, single source shortest paths, minimum spanning tree)
	Lecture	Thursday: Flow Networks I (maximum flow, minimum cut)
	Tutorial	Greedy Algorithms, Flow Networks
Week 7 : 8 July - 14 July	Lecture	Monday: Flow Networks II (flow network constructions, bipartite matching)
	Lecture	Thursday: Dynamic Programming I (overlapping subproblems, one-dimensional state spaces)
	Tutorial	Dynamic Programming
Week 8 : 15 July - 21 July	Lecture	Monday: Dynamic Programming II (making change, knapsack, two-dimensional state spaces)
	Lecture	Thursday: Dynamic Programming III (two-dimensional state spaces with linear recurrences, string matching)
	Tutorial	Dynamic Programming
Week 9 : 22 July - 28 July	Lecture	Monday: Dynamic Programming IV (directed acyclic graphs, single source shortest paths, all pairs shortest paths)
	Lecture	Thursday: Intractable Problems I (computationally difficult problems, class P, linear programming, class NP)
	Tutorial	Dynamic Programming
Week 10 : 29 July - 4 August	Lecture	Monday: Intractable Problems II (polynomial-time reductions, NP-complete problems, practical algorithms for NP-complete problems)
	Lecture	Thursday: Intractable Problems III (NP-hard problems, approximation algorithms for NP-hard optimisation problems)
	Tutorial	Intractable Problems

Attendance Requirements

Students are strongly encouraged to attend all classes and review lecture recordings.

General Schedule Information

Formatif will open in 0-week to allow students to commence revision tasks.

No classes will be held in week 6 (flexibility week), or on Monday of week 3 (King's Birthday). Students enrolled in lab classes on King's Birthday are advised to attend other classes in that

week.

Course Resources

Recommended Resources

The course resources are intended to be self-contained, but students may consult either of the following textbooks for supplemental reading.

- Kleinberg and Tardos: *Algorithm Design*
- Cormen, Leiserson, Rivest and Stein: *Introduction to Algorithms*

Course Evaluation and Development

This course is evaluated each session using the myExperience system.
Last term, students reported that the labs became too busy at the end of term with students trying to get late worked marked, so we have implemented earlier deadlines for the first three modules. The number of tasks in modules 1 and 2 has also been reduced according to student feedback.

Staff Details

Position	Name	Email	Location	Phone	Availability	Equitable Learning Services Contact	Primary Contact
Convenor	Aleksandar Ignjatovic		K17 504			No	No
	COURSE EMAIL					Yes	Yes

Other Useful Information

Academic Information

I. Special consideration and supplementary assessment

If you have experienced an illness or misadventure beyond your control that will interfere with your assessment performance, you are eligible to apply for Special Consideration prior to, or within 3 working days of, submitting an assessment or sitting an exam.

Please note that UNSW has a Fit to Sit rule, which means that if you sit an exam, you are declaring yourself fit enough to do so and cannot later apply for Special Consideration.

For details of applying for Special Consideration and conditions for the award of supplementary assessment, please see the information on UNSW's Special Consideration page.

II. Administrative matters and links

All students are expected to read and be familiar with UNSW guidelines and polices. In particular, students should be familiar with the following:

- Attendance
- UNSW Email Address
- Special Consideration
- Exams
- Approved Calculators
- Academic Honesty and Plagiarism
- Equitable Learning Services

III. Equity and diversity

Those students who have a disability that requires some adjustment in their teaching or learning environment are encouraged to discuss their study needs with the course convener prior to, or at the commencement of, their course, or with the Equity Officer (Disability) in the Equitable Learning Services. Issues to be discussed may include access to materials, signers or note-takers, the provision of services and additional exam and assessment arrangements. Early notification is essential to enable any necessary adjustments to be made.

IV. Professional Outcomes and Program Design

Students are able to review the relevant professional outcomes and program designs for their streams by going to the following link: https://www.unsw.edu.au/engineering/student-life/student-resources/program-design.

Note: This course outline sets out the description of classes at the date the Course Outline is published. The nature of classes may change during the Term after the Course Outline is published. Moodle or your primary learning management system (LMS) should be consulted for the up-to-date class descriptions. If there is any inconsistency in the description of activities between the University timetable and the Course Outline/Moodle/LMS, the description in the Course Outline/Moodle/LMS applies.

Academic Honesty and Plagarism

UNSW has an ongoing commitment to fostering a culture of learning informed by academic integrity. All UNSW students have a responsibility to adhere to this principle of academic integrity. Plagiarism undermines academic integrity and is not tolerated at UNSW. *Plagiarism at*

UNSW is defined as using the words or ideas of others and passing them off as your own.

Plagiarism is a type of intellectual theft. It can take many forms, from deliberate cheating to accidentally copying from a source without acknowledgement. UNSW has produced a website with a wealth of resources to support students to understand and avoid plagiarism, visit: student.unsw.edu.au/plagiarism. The Learning Centre assists students with understanding academic integrity and how not to plagiarise. They also hold workshops and can help students one-on-one.

You are also reminded that careful time management is an important part of study and one of the identified causes of plagiarism is poor time management. Students should allow sufficient time for research, drafting and the proper referencing of sources in preparing all assessment tasks.

Repeated plagiarism (even in first year), plagiarism after first year, or serious instances, may also be investigated under the Student Misconduct Procedures. The penalties under the procedures can include a reduction in marks, failing a course or for the most serious matters (like plagiarism in an honours thesis or contract cheating) even suspension from the university. The Student Misconduct Procedures are available here:

www.gs.unsw.edu.au/policy/documents/studentmisconductprocedures.pdf

Submission of Assessment Tasks

Work submitted late without an approved extension by the course coordinator or delegated authority is subject to a late penalty of five percent (5%) of the maximum mark possible for that assessment item, per calendar day.

The late penalty is applied per calendar day (including weekends and public holidays) that the assessment is overdue. There is no pro-rata of the late penalty for submissions made part way through a day. This is for all assessments where a penalty applies.

Work submitted after five days (120 hours) will not be accepted and a mark of zero will be awarded for that assessment item.

For some assessment items, a late penalty may not be appropriate. These will be clearly indicated in the course outline, and such assessments will receive a mark of zero if not completed by the specified date. Examples include:

- Weekly online tests or laboratory work worth a small proportion of the subject mark;
- Exams, peer feedback and team evaluation surveys;
- Online quizzes where answers are released to students on completion;
- Professional assessment tasks, where the intention is to create an authentic assessment that has an absolute submission date; and,
- Pass/Fail assessment tasks.

Faculty-specific Information

Engineering Student Support Services – The Nucleus - enrolment, progression checks, clash requests, course issues or program-related queries

Engineering Industrial Training – Industrial training questions

UNSW Study Abroad – study abroad student enquiries (for inbound students)

UNSW Exchange – student exchange enquiries (for inbound students)

UNSW Future Students – potential student enquiries e.g. admissions, fees, programs, credit transfer

Phone

(+61 2) 9385 8500 – Nucleus Student Hub

(+61 2) 9385 7661 – Engineering Industrial Training

(+61 2) 9385 3179 – UNSW Study Abroad and UNSW Exchange (for inbound students)

School Contact Information

CSE Help! - on the Ground Floor of K17

- For assistance with coursework assessments.

The Nucleus Student Hub - https://nucleus.unsw.edu.au/en/contact-us

- Course enrolment queries.

Grievance Officer - grievance-officer@cse.unsw.edu.au

- If the course convenor gives an inadequate response to a query or when the courses convenor does not respond to a query about assessment.

Student Reps - stureps@cse.unsw.edu.au

- If some aspect of a course needs urgent improvement. (e.g. Nobody responding to forum queries, cannot understand the lecturer)

You should **never** contact any of the following people directly:

- Vice Chancellor

- Pro-vice Chancellor Education (PVCE)

- Head of School

- CSE administrative staff

- CSE teaching support staff

They will simply bounce the email to one of the above, thereby creating an unnecessary level of indirection and a delay in the response.