COMP2010

Algorithms and Data Structures

Session 1, In person-scheduled-weekday, North Ryde 2024

School of Computing

Contents

General Information

Unit convenor and teaching staff
Convener, Lecturer
Bernard Mans
bernard.mans@mq.edu.au

Lecturer
Mark Dras
mark.dras@mq.edu.au

Credit points
10

Prerequisites
(COMP1010 or COMP125) and 10cp from (MATH132-MATH136 or DMTH137 or MATH1007-MATH1025 or (STAT150 or STAT1250) or (STAT170 or STAT1170) or (STAT171 or STAT1371) or (STAT175 or STAT1175))

Corequisites

Co-badged status
COMP6011

Unit description
This unit provides a study of algorithms, data structures and programming techniques. The topics covered include: trees; graphs and heaps; advanced sorting techniques; elements of storage management; and complexity. The presentation emphasises the role of data abstraction and correctness proofs.

Important Academic Dates

Information about important academic dates including deadlines for withdrawing from units are available at https://www.mq.edu.au/study/calendar-of-dates

Learning Outcomes

On successful completion of this unit, you will be able to:

ULO1: Demonstrate an understanding of a variety of algorithm design techniques and how they can improve either efficiency or clarity.

ULO2: Apply strategies for achieving correctness in a range of algorithms.

ULO3: Apply commonly used data structures, including trees, graphs, lists and their

variations.

ULO4: Carry-out advanced and broadly based problem solving, particularly when designing and writing programs to meet a given specification.

ULO5: Describe the results of analysing algorithms.

General Assessment Information

Standards and Grading

The final mark for the unit will be calculated by combining the marks for all assessment tasks according to the percentage weightings shown in the assessment summary.

Requirements to Pass this Unit

To pass this unit you must achieve a total mark equal to or greater than 50%

Weekly Exercises (Programming Task)

Full marks for these components (10%) can be achieved by completing the tasks for 8 weeks.

Late Assessment Submission Penalty

Unless a Special Consideration request has been submitted and approved, a 5% penalty (of the total possible mark of the task) will be applied for each day a written report or presentation assessment is not submitted, up until the 7thday (including weekends). After the 7th day, a grade of '0' will be awarded even if the assessment is submitted. The submission time for all uploaded assessments is **11:55 pm**. A 1-hour grace period will be provided to students who experience a technical concern.

For any late submission of time-sensitive tasks, such as scheduled tests/exams, performance assessments/presentations, and/or scheduled practical assessments/labs, please apply for Special Consideration.

Assessments where Late Submissions will be accepted

Assignments One and Two: YES, Standard Late Penalty applies

All others: NO, unless Special Consideration is granted

Extension Requests and Special Consideration

The Special Consideration Policy aims to support students who have been impacted by short-term circumstances or events that are serious, unavoidable and significantly disruptive, and which may affect their performance in assessment. If you experience circumstances or events that affect your ability to complete the assessments in this unit on time, please inform the convenor and submit a Special Consideration request through ask.mq.edu.au. (You would **not normally submit Special Consideration requests for Weekly Exercises**, as the marks can be gained in other ways: you only need 8 weeks to get full marks, for example.)

Please note if you cannot submit on time because of illness or other circumstances, please contact the lecturer **before** the due date. If you experience a disruption to studies, you should notify the university. Please note that this is a centralised process, and resolution can take some

time. This may mean, for example, that you are notified that your disruption request has been approved only after any reasonable length extension for an assignment could be granted: for instance, the assignment might have already been handed back. **With respect to assignments, you should therefore also notify the lecturer responsible for the assignment, and submit a solution to the assignment via iLearn, at the same time as you lodge your official disruption notification**. Failure to do so means that an extension may not be possible, leaving only some other remedy listed under the disruption to study outcomes schedule (e.g. partake in assessment task next available session).

Special Consideration for Exams

If you receive special consideration for the final exam, a supplementary exam will be scheduled in the interval between the regular exam period and the start of the next session. By making a special consideration application for the final exam you are declaring yourself available for a resit during the supplementary examination period. Please ensure you are familiar with the policy prior to submitting an application. You can check the supplementary exam information page on FSE101 in iLearn (bit.ly/FSESupp) for dates, and approved applicants will receive an individual notification prior to the exam with the exact date and time of their supplementary examination.

Summary of achievement required corresponding to each final grade

- **HD and D** Overall the quality of the work demonstrates a mature and considered appreciation of the programming and algorithmic concepts, and an excellent technical mastery of Java programming (sufficient to complete the advanced programming tasks). A systematic demonstration of the ability to problem solve independently and a thorough knowledge of how to critique the proposed solution, in terms of performance, correctness and other technical issues.
- **Cr** Overall the quality of the work demonstrates a reasonable appreciation of the programming and algorithmic concepts, and a good technical mastery of Java programming (sufficient to complete the required programming tasks). A systematic demonstration of the ability to solve basic problems and to present the solutions clearly with an attempt to give reasons why they meet their stated objectives. Some knowledge of how to critique the proposed solution, in terms of performance, correctness and other technical issues is demonstrated, but the answers given might not cover all cases.
- **P** The quality of work demonstrates a basic technical mastery of the Java language, a basic understanding of how to program using the studied algorithms and a knowledge of how to implement and use the basic algorithmic data structures and programming techniques introduced in the course. The assessment work demonstrates a basic understanding of performance and correctness issues relative to all of the algorithms and data structures studied in the unit, and the appropriateness of a particular algorithm relative to a given data structure.

Weekly Exercises

Assessment Type [1]: Programming Task Indicative Time on Task [2]: 12 hours Due: **weekly** Weighting: 10**%**

Each week you will be asked to submit the solutions to problems based on lecture material.

Please note that **mixed classes** will start in **week 2**.

On successful completion you will be able to:

- Demonstrate an understanding of a variety of algorithm design techniques and how they can improve either efficiency or clarity.
- Apply strategies for achieving correctness in a range of algorithms.
- Apply commonly used data structures, including trees, graphs, lists and their variations.
- Carry-out advanced and broadly based problem solving, particularly when designing and writing programs to meet a given specification.

Assignment One

Assessment Type [1]: Programming Task Indicative Time on Task [2]: 10 hours Due: **week 6** Weighting: **15%**

In this assignment you will be asked to design and analyse an algorithm based on material studied in weeks 1--5. Your algorithm will be implemented in the Java programming language using some of the design techniques taught in lectures and the weekly exercises. The focus is on correctness and the ability to write programs on list or tree data structures.

On successful completion you will be able to:

- Demonstrate an understanding of a variety of algorithm design techniques and how they can improve either efficiency or clarity.
- Apply strategies for achieving correctness in a range of algorithms.
- Apply commonly used data structures, including trees, graphs, lists and their variations.

Mid semester test

Assessment Type [1]: Quiz/Test Indicative Time on Task [2]: 10 hours Due: **week 10 lecture** Weighting: **10%**

Mid semester test based on tutorial questions in weeks 1--9. This will be delivered as an iLearn Quiz.

On successful completion you will be able to:

- Demonstrate an understanding of a variety of algorithm design techniques and how they can improve either efficiency or clarity.
- Apply strategies for achieving correctness in a range of algorithms.
- Apply commonly used data structures, including trees, graphs, lists and their variations.

Assignment Two

Assessment Type [1]: Programming Task Indicative Time on Task [2]: 20 hours Due: **week 11** Weighting: **20%**

You will be asked to design and implement an algorithm in Java based on graph data structures using some of the more advanced techniques discussed in lectures.

On successful completion you will be able to:

- Demonstrate an understanding of a variety of algorithm design techniques and how they can improve either efficiency or clarity.
- Apply strategies for achieving correctness in a range of algorithms.
- Apply commonly used data structures, including trees, graphs, lists and their variations.

Final Exam

Assessment Type [1]: Examination Indicative Time on Task [2]: 13 hours Due: **exam period** Weighting: **45%**

A formal written examination based on lectures, class work, activities, and assignments.

On successful completion you will be able to:

- Demonstrate an understanding of a variety of algorithm design techniques and how they can improve either efficiency or clarity.
- Apply strategies for achieving correctness in a range of algorithms.
- Apply commonly used data structures, including trees, graphs, lists and their variations.
- Carry-out advanced and broadly based problem solving, particularly when designing and writing programs to meet a given specification.

- Describe the results of analysing algorithms.

[1] If you need help with your assignment, please contact:

- the academic teaching staff in your unit for guidance in understanding or completing this type of assessment
- the Writing Centre for academic skills support.

[2] Indicative time-on-task is an estimate of the time required for completion of the assessment task and is subject to individual variation

Assessment Tasks

Name	Weighting	Hurdle	Due
Mid semester test	10%	No	Week 10 Lecture
Final Exam	45%	No	Weeks 14-16
Assignment Two	20%	No	End of Week 11 (19 May)
Assignment One	15%	No	End of Week 6 (31 March)
Weekly Exercises	5%	No	weekly on Sundays
Contribution to Learning	5%	No	weekly

Mid semester test

Assessment Type [1]: Quiz/Test
Indicative Time on Task [2]: 10 hours
Due: **Week 10 Lecture**
Weighting: **10%**

Mid semester test based on tutorial questions in weeks 1--9. This will be delivered as an iLearn Quiz.

On successful completion you will be able to:

- Demonstrate an understanding of a variety of algorithm design techniques and how they can improve either efficiency or clarity.
- Apply strategies for achieving correctness in a range of algorithms.
- Apply commonly used data structures, including trees, graphs, lists and their variations.

Final Exam

Assessment Type [1]: Examination
Indicative Time on Task [2]: 13 hours
Due: **Weeks 14-16**
Weighting: **45%**

A formal written examination based on lectures, class work, activities, and assignments.

On successful completion you will be able to:

- Demonstrate an understanding of a variety of algorithm design techniques and how they can improve either efficiency or clarity.
- Apply strategies for achieving correctness in a range of algorithms.
- Apply commonly used data structures, including trees, graphs, lists and their variations.
- Carry-out advanced and broadly based problem solving, particularly when designing and writing programs to meet a given specification.
- Describe the results of analysing algorithms.

Assignment Two

Assessment Type [1]: Programming Task
Indicative Time on Task [2]: 20 hours
Due: **End of Week 11 (19 May)**
Weighting: **20%**

You will be asked to design and implement an algorithm in Java based on graph data structures using some of the more advanced techniques discussed in lectures.

On successful completion you will be able to:

- Demonstrate an understanding of a variety of algorithm design techniques and how they can improve either efficiency or clarity.
- Apply strategies for achieving correctness in a range of algorithms.
- Apply commonly used data structures, including trees, graphs, lists and their variations.

Assignment One

Assessment Type [1]: Programming Task

Indicative Time on Task [2]: 10 hours
Due: **End of Week 6 (31 March)**
Weighting: **15%**

In this assignment you will be asked to design and analyse an algorithm based on material studied in weeks 1--5. Your algorithm will be implemented in the Java programming language using some of the design techniques taught in lectures and the weekly exercises. The focus is on correctness and the ability to write programs on list or tree data structures.

On successful completion you will be able to:

- Demonstrate an understanding of a variety of algorithm design techniques and how they can improve either efficiency or clarity.
- Apply strategies for achieving correctness in a range of algorithms.
- Apply commonly used data structures, including trees, graphs, lists and their variations.

Weekly Exercises

Assessment Type [1]: Programming Task
Indicative Time on Task [2]: 12 hours
Due: **weekly on Sundays**
Weighting: **5%**

Each week you will be asked to submit the solutions to problems based on lecture material.

On successful completion you will be able to:

- Demonstrate an understanding of a variety of algorithm design techniques and how they can improve either efficiency or clarity.
- Apply strategies for achieving correctness in a range of algorithms.
- Apply commonly used data structures, including trees, graphs, lists and their variations.
- Carry-out advanced and broadly based problem solving, particularly when designing and writing programs to meet a given specification.

Contribution to Learning

Assessment Type [1]: Participatory task
Indicative Time on Task [2]: 0 hours
Due: **weekly**
Weighting: **5%**

The participation assessment encourages active and consistent engagement in COMP2010 content. There are two ways to obtain marks. (a) Attend a weekly workshop and complete additional participation exercises (0.5 mark from the tutor at the workshop). (b) Good citizenship eg consistent posting useful comments and contributions related to the material on the forum. Only tutors may nominate students for good citizenship participation (b), and the lecturers will be happy to consider such nominations.

On successful completion you will be able to:

- Demonstrate an understanding of a variety of algorithm design techniques and how they can improve either efficiency or clarity.
- Apply strategies for achieving correctness in a range of algorithms.
- Apply commonly used data structures, including trees, graphs, lists and their variations.
- Carry-out advanced and broadly based problem solving, particularly when designing and writing programs to meet a given specification.
- Describe the results of analysing algorithms.

---

[1] If you need help with your assignment, please contact:

- the academic teaching staff in your unit for guidance in understanding or completing this type of assessment
- the Writing Centre for academic skills support.

[2] Indicative time-on-task is an estimate of the time required for completion of the assessment task and is subject to individual variation

Delivery and Resources

Technology required

- Eclipse - download Eclipse IDE for Java Developers: The practical work in this unit involves programming in Java (www.java.com) using the Eclipse Integrated Development Environment (www.eclipse.org)
- Java SE JDK - download Java SE 8 to be compatible with the labs: Note that you need the Java JDK which includes the compiler tools, rather than the Java Runtime Environment (JRE) which you might already have installed on your computer to allow you to run Java applications.
- Any additional Java libraries will be made available for download.

- Learning Management System iLearn : This will be used primarily to enable email broadcasts and give access to Assessment marks.
- The lecture audio will be recorded, and will be available via iLearn.

Classes

Each week you should attend **2 hours of lectures** and **a two-hour mixed classes**. For details of days, times and rooms consult the timetables webpage.

You should have selected one two-hour mixed classes session at enrolment. **You must attend the session you are enrolled in**.

Please note that **mixed classes** will start in **week 2**.

Please note that you are **expected** to attend most of the mixed classes because that is your opportunity to seek clarification of any parts of the course and exercises you do not understand. Note that the in-class quiz will be strongly based on the weekly exercises. You are therefore **strongly advised** to complete the set class exercises, and to seek clarification when you are unable to complete a question.

Recommended Texts

The following texts can be used to supplement the material covered in lectures:

- Clifford Shaffer. Data Structures and Algorithm Analysis - available online at https://people.cs.vt.edu/shaffer/Book/JAVA3e20130328.pdf
- Robert Lafore. Data Structures and Algorithms in Java, 2nd edition - available online at https://learning.oreilly.com/home/ [You can use your Macquarie email to get a freely accessible account at O'Reilly for all of their books.]
- Robert Sedgewick and Kevin Wayne. Algorithms (4th edition) - available online at https://algs4.cs.princeton.edu/home/ [This is a good text if you want to go deeper, and is very well-known in the field.]
- Adam Drozdek [2005]. *Data Structures and Algorithms in Java* (2nd ed. or 3rd edition). Boston: Thomson Course Technology.
    - There is also a companion website by the publisher, containing data files for exercises. In addition, Drozdek has Java code from the book available on his webpage. (Note that these are written for Java 1.4.)

Unit Pages

The unit will make use of discussions hosted within iLearn. Please post questions there, they will be monitored by the staff on the unit.

Teaching and Learning Strategy

COMP2010 is taught via lectures and mixed classes in the laboratory. Lectures are used to introduce new theoretic material, give examples of the use these techniques and put them in a wider context. Mixed classes give you the opportunity to interact with your peers. You will be given problems to solve each week prior to each session; preparing solutions is important because it will allow you to discuss the problems effectively with your tutor thereby making the most of this activity. The aim of the mixed classes is to help you to develop problem-solving skills and teamwork, and you will be expected to work on problems in class. Mixed classes give you an opportunity to practice your programming skills, and to implement many of the ideas discussed in lectures. Each week you will be given a number of problems to work on; it is important that you keep up with these problems as doing so will help you understand the material in the unit and prepare you for the work in assignments and quizzes. Some of the questions are designated **priority** and they will be the ones that will be discussed in detail and on which the quizzes may be based. Additional questions are provided for extension and general practice.

There will be an opportunity to explore more deeply aspects of the course material which has not been covered in lectures or classes. These will sometimes be student-led, and in various forms including Q&A with the lecturer or short videos. Topics will for example include questions not covered in workshops, or hints and tips for assignments. More information for the timing of these sessions will be available on iLearn.

Lecture notes will be made available each week but these notes are intended as an outline of the lecture only and are not a substitute for your own notes or the more extensive discussions in the various suggested supplementary textbooks.

Unit Schedule

Week 1	Review of algorithms and related concepts
Week 2	Algorithm Correctness and Efficiency
Week 3	Algorithm Design Strategies
Week 4	Sorting
Week 5	Binary Trees
Week 6	Binary Trees (cont.)
Week 7	Priority Queues, Heaps and Heapsort
Week 8	Programming with Maps and Hashtables
Week 9	Graph Algorithms
Week 10	Graph Algorithms (cont.)

Week 11	Advanced Trees
Week 12	An Introduction to Computability
Week 13	Revision

Please note that **mixed classes** will start in **week 2**.

Policies and Procedures

Macquarie University policies and procedures are accessible from Policy Central (https://policies.mq.edu.au). Students should be aware of the following policies in particular with regard to Learning and Teaching:

- Academic Appeals Policy
- Academic Integrity Policy
- Academic Progression Policy
- Assessment Policy
- Fitness to Practice Procedure
- Assessment Procedure
- Complaints Resolution Procedure for Students and Members of the Public
- Special Consideration Policy

Students seeking more policy resources can visit Student Policies (https://students.mq.edu.au/support/study/policies). It is your one-stop-shop for the key policies you need to know about throughout your undergraduate student journey.

To find other policies relating to Teaching and Learning, visit Policy Central (https://policies.mq.edu.au) and use the search tool.

Student Code of Conduct

Macquarie University students have a responsibility to be familiar with the Student Code of Conduct: https://students.mq.edu.au/admin/other-resources/student-conduct

Results

Results published on platform other than eStudent, (eg. iLearn, Coursera etc.) or released directly by your Unit Convenor, are not confirmed as they are subject to final approval by the University. Once approved, final results will be sent to your student email address and will be made available in eStudent. For more information visit ask.mq.edu.au or if you are a Global MBA student contact globalmba.support@mq.edu.au

Academic Integrity

At Macquarie, we believe academic integrity – honesty, respect, trust, responsibility, fairness and courage – is at the core of learning, teaching and research. We recognise that meeting the expectations required to complete your assessments can be challenging. So, we offer you a

range of resources and services to help you reach your potential, including free online writing and maths support, academic skills development and wellbeing consultations.

Student Support

Macquarie University provides a range of support services for students. For details, visit http://students.mq.edu.au/support/

The Writing Centre

The Writing Centre provides resources to develop your English language proficiency, academic writing, and communication skills.

- Workshops
- Chat with a WriteWISE peer writing leader
- Access StudyWISE
- Upload an assignment to Studiosity
- Complete the Academic Integrity Module

The Library provides online and face to face support to help you find and use relevant information resources.

- Subject and Research Guides
- Ask a Librarian

Student Services and Support

Macquarie University offers a range of Student Support Services including:

- IT Support
- Accessibility and disability support with study
- Mental health support
- Safety support to respond to bullying, harassment, sexual harassment and sexual assault
- Social support including information about finances, tenancy and legal issues
- Student Advocacy provides independent advice on MQ policies, procedures, and processes

Student Enquiries

Got a question? Ask us via AskMQ, or contact Service Connect.

IT Help

For help with University computer systems and technology, visit http://www.mq.edu.au/about_us/offices_and_units/information_technology/help/.

When using the University's IT, you must adhere to the Acceptable Use of IT Resources Policy.

The policy applies to all who connect to the MQ network including students.

Changes from Previous Offering

Weekly exercises are now worth 10% (instead of 5% previously).

Unit information based on version 2024.01R of the Handbook