COMP225

Algorithms and Data Structures

S1 Day 2018

Dept of Computing

Contents

General Information	2
Learning Outcomes	3
General Assessment Information	3
Assessment Tasks	4
Delivery and Resources	7
Unit Schedule	8
Policies and Procedures	9
Graduate Capabilities	10
Changes from Previous Offering	14
Changes since First Published	14

Disclaimer

Macquarie University has taken all reasonable measures to ensure the information in this publication is accurate and up-to-date. However, the information may change or become out-dated as a result of change in University policies, procedures or rules. The University reserves the right to make changes to any information in this publication without notice. Users of this publication are advised to check the website version of this publication [or the relevant faculty or department] before acting on any information in this publication.

General Information

Unit convenor and teaching staff

Convenor
Annabelle McIver
annabelle.mciver@mq.edu.au
Contact via Email

Lecturer
Mark Dras
mark.dras@mq.edu.au
Contact via Email

Tutor
Daniel Sutantyo
daniel.sutantyo@mq.edu.au
Contact via Email

Tutor
Kym Haines
kym.haines@mq.edu.au

Tutor
Sonit Singh
sonit.singh@mq.edu.au

Credit points
3

Prerequisites
COMP125 and 3cp from (MATH132-MATH136 or DMTH137 or STAT150 or STAT170 or STAT171 or STAT175)

Corequisites

Co-badged status

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:

Be familiar with a variety of algorithm design techniques and understand how they can improve either efficiency or clarity.

Be aware of the importance of correctness for algorithms, and able to apply strategies for achieving correctness.

Be able to apply commonly used data structures (including trees, graphs, lists and their variations).

Carrying out advanced and broadly based problem solving - most particularly when designing and writing programs in Java to meet a given specification.

Be able to describe 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.

Late penalty: Late submission of assignments will normally accrue a late penalty of up to 20% reduction for each day late.

Extension requests: 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 (http://www.students.mq.edu.au/student_admin/manage_your_study_program/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

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 and will not be eligible for a second special consideration approval based on pre-existing commitments. 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 one week 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.

Assessment Tasks

Name	Weighting	Hurdle	Due
Assignment 1	20%	No	Week 7, after the break

Name	Weighting	Hurdle	Due
Programming test	5%	No	Week 3
In class test	10%	No	Week 9
Assignment 2	20%	No	Week 12
Final exam	45%	No	Examination period

Assignment 1

Due: **Week 7, after the break**
Weighting: **20%**

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 use recursion on list or tree data structures.

Submission method: via iLearn.

On successful completion you will be able to:

- Be familiar with a variety of algorithm design techniques and understand how they can improve either efficiency or clarity.
- Be aware of the importance of correctness for algorithms, and able to apply strategies for achieving correctness.
- Be able to apply commonly used data structures (including trees, graphs, lists and their variations).
- Carrying out advanced and broadly based problem solving - most particularly when designing and writing programs in Java to meet a given specification.

Programming test

Due: **Week 3**
Weighting: **5%**

Java programming assessment to take place during the scheduled weekly class. This will be a diagnostic test which will review general programming skills. Your code will be assessed via automated tests that will be provided.

On successful completion you will be able to:

- Be familiar with a variety of algorithm design techniques and understand how they can improve either efficiency or clarity.

- Carrying out advanced and broadly based problem solving - most particularly when designing and writing programs in Java to meet a given specification.
- Be able to describe results of analysing algorithms.

In class test

Due: **Week 9**
Weighting: **10%**

In class test based on tutorial questions in weeks 1--9.

Submission method: on paper.

On successful completion you will be able to:

- Be familiar with a variety of algorithm design techniques and understand how they can improve either efficiency or clarity.
- Carrying out advanced and broadly based problem solving - most particularly when designing and writing programs in Java to meet a given specification.
- Be able to describe results of analysing algorithms.

Assignment 2

Due: **Week 12**
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.

Submission method: via iLearn.

On successful completion you will be able to:

- Be familiar with a variety of algorithm design techniques and understand how they can improve either efficiency or clarity.
- Be aware of the importance of correctness for algorithms, and able to apply strategies for achieving correctness.
- Be able to apply commonly used data structures (including trees, graphs, lists and their variations).
- Carrying out advanced and broadly based problem solving - most particularly when designing and writing programs in Java to meet a given specification.

Final exam

Due: **Examination period**
Weighting: **45%**

A formal written examination based on lectures, class work (weeks 1--12) and activities and assignments.

Submission method: Formal examination organised by the University.

On successful completion you will be able to:

- Be familiar with a variety of algorithm design techniques and understand how they can improve either efficiency or clarity.
- Be aware of the importance of correctness for algorithms, and able to apply strategies for achieving correctness.
- Be able to apply commonly used data structures (including trees, graphs, lists and their variations).
- Carrying out advanced and broadly based problem solving - most particularly when designing and writing programs in Java to meet a given specification.
- Be able to describe results of analysing algorithms.

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 **3 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 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 you will not be required to submit work every week, however 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:

Robert Sedgewick and Kevin Wayne. Algorithms (4th edition) - available online at https://algs4.cs.princeton.edu/home/

Clifford Shaffer. Data Structures and Algorithm Analysis - available online at https://people.cs.vt.edu/shaffer/Book/JAVA3e20130328.pdf

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

COMP225 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.

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 textbook.

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
16-27 April	Mid semester break
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

Policies and Procedures

Macquarie University policies and procedures are accessible from Policy Central (https://staff.mq.edu.au/work/strategy-planning-and-governance/university-policies-and-procedures/policy-central). 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
- Grade Appeal Policy
- Complaint Management Procedure for Students and Members of the Public
- Special Consideration Policy *(**Note:** The Special Consideration Policy is effective from 4 December 2017 and replaces the Disruption to Studies Policy.)*

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

If you would like to see all the policies relevant to Learning and Teaching visit Policy Central (https://staff.mq.edu.au/work/strategy-planning-and-governance/university-policies-and-procedures/p

olicy-central).

Student Code of Conduct

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

Results

Results shown in *iLearn*, 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.

Student Support

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

Learning Skills

Learning Skills (mq.edu.au/learningskills) provides academic writing resources and study strategies to improve your marks and take control of your study.

- Workshops
- StudyWise
- Academic Integrity Module for Students
- Ask a Learning Adviser

Student Services and Support

Students with a disability are encouraged to contact the Disability Service who can provide appropriate help with any issues that arise during their studies.

Student Enquiries

For all student enquiries, visit Student Connect at ask.mq.edu.au

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.

Graduate Capabilities

Creative and Innovative

Our graduates will also be capable of creative thinking and of creating knowledge. They will be imaginative and open to experience and capable of innovation at work and in the community. We

want them to be engaged in applying their critical, creative thinking.

This graduate capability is supported by:

Learning outcomes

- Be able to apply commonly used data structures (including trees, graphs, lists and their variations).
- Carrying out advanced and broadly based problem solving - most particularly when designing and writing programs in Java to meet a given specification.

Assessment tasks

- Assignment 1
- Assignment 2
- Final exam

Capable of Professional and Personal Judgement and Initiative

We want our graduates to have emotional intelligence and sound interpersonal skills and to demonstrate discernment and common sense in their professional and personal judgement. They will exercise initiative as needed. They will be capable of risk assessment, and be able to handle ambiguity and complexity, enabling them to be adaptable in diverse and changing environments.

This graduate capability is supported by:

Assessment tasks

- Programming test
- In class test

Commitment to Continuous Learning

Our graduates will have enquiring minds and a literate curiosity which will lead them to pursue knowledge for its own sake. They will continue to pursue learning in their careers and as they participate in the world. They will be capable of reflecting on their experiences and relationships with others and the environment, learning from them, and growing - personally, professionally and socially.

This graduate capability is supported by:

Learning outcomes

- Be familiar with a variety of algorithm design techniques and understand how they can improve either efficiency or clarity.
- Be aware of the importance of correctness for algorithms, and able to apply strategies for achieving correctness.

Assessment tasks

- Programming test
- In class test
- Final exam

Discipline Specific Knowledge and Skills

Our graduates will take with them the intellectual development, depth and breadth of knowledge, scholarly understanding, and specific subject content in their chosen fields to make them competent and confident in their subject or profession. They will be able to demonstrate, where relevant, professional technical competence and meet professional standards. They will be able to articulate the structure of knowledge of their discipline, be able to adapt discipline-specific knowledge to novel situations, and be able to contribute from their discipline to inter-disciplinary solutions to problems.

This graduate capability is supported by:

Learning outcomes

- Be familiar with a variety of algorithm design techniques and understand how they can improve either efficiency or clarity.
- Be aware of the importance of correctness for algorithms, and able to apply strategies for achieving correctness.
- Be able to apply commonly used data structures (including trees, graphs, lists and their variations).

Assessment tasks

- Assignment 1
- Assignment 2
- Final exam

Critical, Analytical and Integrative Thinking

We want our graduates to be capable of reasoning, questioning and analysing, and to integrate and synthesise learning and knowledge from a range of sources and environments; to be able to critique constraints, assumptions and limitations; to be able to think independently and systemically in relation to scholarly activity, in the workplace, and in the world. We want them to have a level of scientific and information technology literacy.

This graduate capability is supported by:

Learning outcomes

- Be familiar with a variety of algorithm design techniques and understand how they can improve either efficiency or clarity.

- Be aware of the importance of correctness for algorithms, and able to apply strategies for achieving correctness.
- Be able to apply commonly used data structures (including trees, graphs, lists and their variations).
- Carrying out advanced and broadly based problem solving - most particularly when designing and writing programs in Java to meet a given specification.
- Be able to describe results of analysing algorithms.

Assessment tasks

- Assignment 1
- Programming test
- In class test
- Assignment 2
- Final exam

Problem Solving and Research Capability

Our graduates should be capable of researching; of analysing, and interpreting and assessing data and information in various forms; of drawing connections across fields of knowledge; and they should be able to relate their knowledge to complex situations at work or in the world, in order to diagnose and solve problems. We want them to have the confidence to take the initiative in doing so, within an awareness of their own limitations.

This graduate capability is supported by:

Learning outcomes

- Be familiar with a variety of algorithm design techniques and understand how they can improve either efficiency or clarity.
- Be able to apply commonly used data structures (including trees, graphs, lists and their variations).
- Carrying out advanced and broadly based problem solving - most particularly when designing and writing programs in Java to meet a given specification.

Assessment tasks

- Assignment 1
- Programming test
- In class test
- Assignment 2
- Final exam

Effective Communication

We want to develop in our students the ability to communicate and convey their views in forms effective with different audiences. We want our graduates to take with them the capability to read, listen, question, gather and evaluate information resources in a variety of formats, assess, write clearly, speak effectively, and to use visual communication and communication technologies as appropriate.

This graduate capability is supported by:

Learning outcomes

- Be able to apply commonly used data structures (including trees, graphs, lists and their variations).
- Carrying out advanced and broadly based problem solving - most particularly when designing and writing programs in Java to meet a given specification.
- Be able to describe results of analysing algorithms.

Assessment task

- Final exam

Changes from Previous Offering

There is a programming test in week 3, based on revision of COMP125 material.

There is a single in class test in week 10.

The weightings for each component are slightly different than last year.

Changes since First Published

Date	Description
27/02/2018	David Lewis dropped out as tutor a week ago. We have replaced him with Sonit Singh, therefore the only change is the list of tutors.