



COMP795

Research Topic in Computing 5

S1 Day 2014

Computing

Contents

<u>General Information</u>	2
<u>Learning Outcomes</u>	2
<u>Assessment Tasks</u>	3
<u>Delivery and Resources</u>	5
<u>Unit Schedule</u>	6
<u>Policies and Procedures</u>	6
<u>Assessment Standards</u>	8

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

Unit Convenor

Luke Mathieson

luke.mathieson@mq.edu.au

Contact via luke.mathieson@mq.edu.au

Credit points

4

Prerequisites

Admission to MRes

Corequisites

Co-badged status

Unit description

This unit comprises study of an advanced topic in the areas of computing and information systems. The area studied each year is tailored to the current student cohort. Emphasis is on both the understanding of analytical/theoretical concepts and problem-solving applications.

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:

Explain key ideas in the field of algorithmics and the workings of key algorithms, and compare and evaluate algorithmic solutions for computational problems.

Understand and formally analyse the computational complexity of algorithms.

Implement key algorithms.

Develop algorithmic solutions for computational problems by constructing new algorithms and combining existing algorithms.

Investigate topics in advanced algorithms and synthesise the output for presentation in oral and written form.

Assessment Tasks

Name	Weighting	Due
<u>Presentation</u>	15%	Week 13
<u>Weekly Homework</u>	30%	Week 2 to Week 13
<u>Programming Assignment</u>	25%	Week 12
<u>Knowledge Checklist</u>	30%	Week 2 to Week 13

Presentation

Due: **Week 13**

Weighting: **15%**

An oral presentation supported by appropriate presentation materials. The presentation will be 20 minutes long followed by 5 minutes for questions. The presentations will be made in class in week 13.

On successful completion you will be able to:

- Explain key ideas in the field of algorithmics and the workings of key algorithms, and compare and evaluate algorithmic solutions for computational problems.
- Understand and formally analyse the computational complexity of algorithms.
- Investigate topics in advanced algorithms and synthesise the output for presentation in oral and written form.

Weekly Homework

Due: **Week 2 to Week 13**

Weighting: **30%**

Each week students will be asked to complete some short exercises related to the weekly material.

On successful completion you will be able to:

- Explain key ideas in the field of algorithmics and the workings of key algorithms, and compare and evaluate algorithmic solutions for computational problems.
- Understand and formally analyse the computational complexity of algorithms.
- Implement key algorithms.
- Develop algorithmic solutions for computational problems by constructing new algorithms and combining existing algorithms.

- Investigate topics in advanced algorithms and synthesise the output for presentation in oral and written form.

Programming Assignment

Due: **Week 12**

Weighting: **25%**

Students will be required to develop and program algorithmic solutions to an individually chosen computational problem. The program will be accompanied by a brief report analysing the complexity of the chosen solution and discussing the reasons for the choice of algorithm(s). The implementation language may be chosen by the student subject to approval of the lecturer.

On successful completion you will be able to:

- Explain key ideas in the field of algorithmics and the workings of key algorithms, and compare and evaluate algorithmic solutions for computational problems.
- Understand and formally analyse the computational complexity of algorithms.
- Implement key algorithms.
- Develop algorithmic solutions for computational problems by constructing new algorithms and combining existing algorithms.
- Investigate topics in advanced algorithms and synthesise the output for presentation in oral and written form.

Knowledge Checklist

Due: **Week 2 to Week 13**

Weighting: **30%**

The students will be furnished with a checklist of questions ranging from basic knowledge questions to more complex analytical and knowledge-synthesis questions covering material from across the course. The students may during the lecturer's office hours attempt to answer the questions orally, with the following restrictions:

1. No question may be attempted more than once per week.
2. The lecturer may, at his discretion, limit the number of question attempted in a given sitting (to ensure fair access).

The questions will be weighted commensurate to their difficulty, and partial credit may be given for answers that demonstrate partial achievement.

On successful completion you will be able to:

- Explain key ideas in the field of algorithmics and the workings of key algorithms, and compare and evaluate algorithmic solutions for computational problems.
- Understand and formally analyse the computational complexity of algorithms.

- Develop algorithmic solutions for computational problems by constructing new algorithms and combining existing algorithms.
- Investigate topics in advanced algorithms and synthesise the output for presentation in oral and written form.

Delivery and Resources

Classes

Each week has three hours of face-to-face class. These classes will be a mixture lecture material, discussion and tutorial material.

Recommended Reading and References

There is no set text for the course, but the following *far from exhaustive* list of texts may be useful for reference, study and further reading:

- J. P. Ignizio and T. M. Cavalier, "Linear Programming", Prentice Hall, 1994.
- T. C. Hu, "Integer Programming and Network Flows", Addison Wesley, 1969.
- F. Rossi, P. van Beek and T. Walsh (eds.), "Handbook of Constraint Programming", Elsevier, 2006.
- R. K. Ahuja, T. L. Magnanti and J. B. Orlin, "Network Flows: Theory, Algorithms, and Applications", Prentice Hall, 1993.
- R. G. Downey and M. R. Fellows, "Parameterized Complexity", Springer, 1999.
- R. G. Downey and M. R. Fellows, "Fundamentals of Parameterized Complexity", Springer, 2013.
- J. Flum and M. Grohe, "Parameterized Complexity Theory", Springer, 2006.
- R. Niedermeier, "An Invitation to Parameterized Algorithms", Springer, 2006.
- V. V. Vazirani, "Approximation Algorithms", Springer, 2001.
- G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, M. Protasi, "Complexity and Approximation: Combinatorial optimization problems and their approximability properties", Springer, 2009.

Unit Webpage, Materials and Technologies Used

The unit webpage "can be found" at <http://comp.mq.edu.au/units/comp795>.

The materials for the unit including notes, discussion fora, electronic submission links etc. will be through the iLearn system.

The programming assignment can be done in any programming language subject to prior approval of the course convener. The default expectation for programming language is Java, and as such does not require approval to use. Similarly, additional libraries may be used subject to prior approval of the course convener. In any case where libraries or code other than that of the student's is used, proper attribution must be given.

Unit Schedule

Algorithms are the essence of computer science. In this unit we build on the undergraduate understanding of algorithms and look at interesting and useful algorithms, both fundamental and cutting edge.

The particular material covered will depend on the cohort but may include topics such as approximation algorithms, exponential-time exact and parameterized algorithms, linear and constraint programming and fundamental graph algorithms such as max-flow algorithms, matching algorithms and so on.

Throughout the unit there is an emphasis on developing not just a knowledge of algorithms, but of general algorithmic techniques alongside a deeper understanding of why algorithms work at a formal level. This helps facilitate the student's ability to construct a 'toolkit' of algorithmic techniques that can be adapted and employed to solve computational problems in their ongoing studies and work.

Part of this understanding will be the ongoing development of an intuition and formal approaches for comparing and evaluating algorithms. As such the unit will also employ appropriate tools from complexity theory to analyse the performance of the algorithms studied.

Students entering this unit should have programming experience commensurate with the completion of an undergraduate computing degree, a working knowledge of basic and intermediate algorithms and experience with discrete mathematics. Further algorithmic and mathematical study will be of benefit, but is not necessary. Directly relevant undergraduate units offered at Macquarie include COMP225 - Algorithms and Data Structures, DMTH237 - Discrete Mathematics II and their prerequisites. Completion of COMP333 - Algorithm Theory and Design would be helpful.

The following is the initial schedule for topics in the unit, this may be subject to change depending on the cohort and the material.

Week	Topic
1	Introduction and unit outline.
1	Review of background materials.
2-4	Linear Programming, the simplex method, interior point methods, integer programming, constraint programming.
5-6	Network flows, matchings, connectivity and cuts.
7-9	Exact exponential time algorithms and parameterized algorithms.
10-12	Approximation algorithms.
13	Presentations and review.

Policies and Procedures

Macquarie University policies and procedures are accessible from [Policy Central](#). Students should be aware of the following policies in particular with regard to

Learning and Teaching:

Academic Honesty Policy http://mq.edu.au/policy/docs/academic_honesty/policy.html

Assessment Policy <http://mq.edu.au/policy/docs/assessment/policy.html>

Grading Policy <http://mq.edu.au/policy/docs/grading/policy.html>

Grade Appeal Policy <http://mq.edu.au/policy/docs/gradeappeal/policy.html>

Grievance Management Policy http://mq.edu.au/policy/docs/grievance_management/policy.html

Disruption to Studies Policy http://www.mq.edu.au/policy/docs/disruption_studies/policy.html *The Disruption to Studies Policy is effective from March 3 2014 and replaces the Special Consideration Policy.*

In addition, a number of other policies can be found in the [Learning and Teaching Category](#) of Policy 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/support/student_conduct/

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://informatics.mq.edu.au/hel>

p/.

When using the University's IT, you must adhere to the [Acceptable Use Policy](#). The policy applies to all who connect to the MQ network including students.

Assessment Standards

COMP795 will be assessed and graded according to the University assessment and grading policies.

Submission Deadlines

Assessment deadlines are strict, unless an application for special consideration is received (preferably in advance) accompanied by appropriate documentary evidence. Late submissions will be penalised at the rate of 20% of the full marks for the assessment per day or part thereof late.

Standards

The following general standards of achievement will be used to assess each of the assessment tasks with respect to the letter grades.

Pass: Has a basic understanding of the algorithms and concepts as discussed in class. Can describe and reproduce definitions and fundamental algorithms. Can perform a basic research investigation in the area and present the results of that research in rudimentary written and oral forms.

Credit: As for Pass plus: Is able to apply the algorithmic techniques we have discussed to derive solutions to computational problems. Can develop, generalise and apply the concepts discussed in class to address basic theoretical and practical questions, and can effectively communicate these insights. Shows more than basic insights into the results of a research investigation and is able to communicate those insights.

Distinction/High Distinction: As for Credit plus: Is able to generalise and synthesise knowledge to address more complex topics beyond the material discussed in class. Can critically evaluate the limits of the techniques and algorithms discussed.

Assessment Process

These assessment standards will be used to give a numeric mark out of 100 to each assessment submission during marking. The mark will correspond to a letter grade for that task according to the University guidelines. The final raw 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.

We will look at your overall performance on all assessments when determining your final grade. A total raw mark of at least 50% and a mark of at least 50% on each of the assessment categories will be sufficient to pass the unit. Students who do not meet this cut-off will be examined on a case-by-case basis.

On occasion your raw mark for the unit may not be the same as the Standardised Numeric Grade (SNG) which you receive as the result. Under the Senate guidelines, raw marks may be

scaled to ensure that there is a degree of comparability across the university, so that units with the same past performances of their students should achieve similar results.