Department of Computing and Information Systems
University of Melbourne

COMP90038: Algorithms and Complexity

Subject Overview

The material covered in Algorithms and Complexity is an essential part of any computing education. Data structures and their algorithms are central to practically every branch of computer science and of direct and practical relevance to any practitioner. The choice of inefficient algorithms and data structures can make an application useless, and, conversely, a careful choice of data structures and algorithms can result in a dramatic improvement in program performance.

In this subject we will look at many examples of algorithms and we will study techniques for analysing their performance. The two most prominent problems that we will consider are sorting and searching, but we will also look at graph algorithms (and, if time permits, the string matching problem).

Algorithms and programming languages are not entirely orthogonal. Sometimes your choice of programming language will affect your choice of algorithm. We shall largely be language-agnostic in this subject: Algorithms will be described in *pseudo-code* an idealised Java-style notation.

On completion of this subject students are expected to:

- design, manipulate and reason about a variety of techniques for solving sorting, searching and graph problems;
- write efficient algorithms and data structures for a variety of fundamental problems;
- conduct formal reasoning about problem complexity and algorithmic efficiency;
- recognize the design techniques of standard algorithms, and apply these techniques to develop new computational solutions to problems.

Teaching Staff

Staff member	Role	Email
A/Prof Michael Kirley	Lecturer	`mkirley@unimelb.edu.au`
Prof Lars Kulik	Lecturer	`lkulik@unimelb.edu.au`
Toby Davies	Head Tutor	`todavies@student.unimelb.edu.au`
Hengfeng Li	Tutor	`henli@student.unimelb.edu.au`
Yuan Sun	Tutor	`yuans2@student.unimelb.edu.au`
Dr Les Kitchen	Tutor	`ljk@unimelb.edu.au`

Contact hours

Lectures: 2 × 1 hour per week (Monday 11:05-11:55 and 12:05-12:55)

Workshop: 1 hour tutorial class (see the University timtetable)

Communication

The major source of out-of-class communication will be via the LMS announcements and discussion board. Frequently asked questions and significant points will be followed up in lecturers/tutorials and/or via the announcements page. If you need to contact the teaching staff directly via email, please include the subject code `COMP90038` in the message subject.

Textbook / Resources

We will use Anany Levitin, *Introduction to the Design and Analysis of Algorithms*, Pearson, 3rd ed., 2012. The book can be bought from the Book Co-op. Seven copies are available for overnight or 7-day loan in the Eastern Resource Centre. The ERC also has about a dozen copies of earlier editions, which are fairly close to the third edition.

There are many excellent books on algorithms, including some that you may find near Levitin's book in the library. I just mention here David Harel's Algorithmics: The Spirit of Computing (Addison-Wesley, third edition 2004) because it is a book written for a wider audience, it goes well beyond algorithms, and it manages to be engaging and clear without sacrificing rigour.

Syllabus / Weekly Schedule

The 'Weekly Schedule' link on the LMS provides a break down of the syllabus and access to the lecture notes and tutorial questions. A high-level overview of topics to be covered is listed below:

- complexity classes and asymptotic notation;
- empirical analysis of algorithms;
- abstract data types including queues, trees, priority queues and graphs;
- algorithmic techniques including brute force, divide-and-conquer, dynamic programming and greedy approaches;
- space and time trade-offs;
- and the theoretical limits of algorithm power

Recommended Background Knowledge: You are expected to come to the subject with some mathematical maturity, including basic discrete mathematics (sets and relations); and an elementary understanding of data structures (arrays, records, and linked lists).

Assessment

Exam 70% (3 hour written exam, held at the end of the semester).

Mid-semester test 10% (held during a lecture in Week 6/7 (date to be confirmed asap).

Projects 20% (2 projects with multiple tasks; due approximately Week 5/6 and Week 10/11).

Hurdle requirements: To pass the subject, you must obtain at least 50% overall and at least 30/70 in the written exam. You must also successfully complete at least eight of the weekly online quizzes.