COMP90038 Algorithms and Complexity

Credit Points:	12.5
Level:	9 (Graduate/Postgraduate)
Dates & Locations:	2016, Parkville This subject commences in the following study period/s: Semester 1, Parkville - Taught on campus. Semester 2, Parkville - Taught on campus.
Time Commitment:	Contact Hours: 36 hours, comprising of one 2-hour lecture and one 1-hour tutorial. Total Time Commitment: 200 hours
Prerequisites:	An undergraduate degree in a cognate discipline.
Corequisites:	None
Recommended Background Knowledge:	Basic discrete mathematics (sets and relations); elementary data structures (arrays, records, and linked lists).
Non Allowed Subjects:	

Subject	Study Period Commencement:	Credit Points:
COMP20003 Algorithms and Data Structures	Semester 2	12.50
COMP20007 Design of Algorithms	Semester 1	12.50

Core Participation Requirements:	<p>For the purposes of considering request for Reasonable Adjustments under the Disability Standards for Education (Cwth 2005), and Student Support and Engagement Policy, academic requirements for this subject are articulated in the Subject Overview, Learning Outcomes, Assessment and Generic Skills sections of this entry.</p> <p>It is University policy to take all reasonable steps to minimise the impact of disability upon academic study, and reasonable adjustments will be made to enhance a student's participation in the University's programs. Students who feel their disability may impact on meeting the requirements of this subject are encouraged to discuss this matter with a Faculty Student Adviser and Student Equity and Disability Support: <a href="http://services.unimelb.edu.au/disability">http://services.unimelb.edu.au/disability</a></p>
Coordinator:	Assoc Prof Harald Sondergaard, Dr Antonette Mendoza, Prof Lars Kulik
Contact:	Semester 1 Prof Lars Kulik **lkulik@unimelb.edu.au (mailto:lkulik@unimelb.edu.au)** Semester 2 A/Prof Harald Sondergaard **harald@unimelb.edu.au (mailto:harald@unimelb.edu.au)**
Subject Overview:	**AIMS** The aim of this subject is for students to develop familiarity and competence in assessing and designing computer programs for computational efficiency. Although computers manipulate data very quickly, to solve large-scale problems, we must design strategies so that the calculations combine effectively. Over the latter half of the 20th century, an elegant theory of computational efficiency developed. This subject introduces students to the fundamentals of this theory and to many of the classical algorithms and data structures that solve key computational questions. These questions include distance computations in networks, searching items in large collections, and sorting them in order. **INDICATIVE CONTENT**

	Topics covered include 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.
Learning Outcomes:	**INTENDED LEARNING OUTCOMES (ILO)** On completion of this subject the student should be able to: 1 Design, manipulate and reason about a variety of techniques for solving sorting, searching and graph problems 2 Write efficient algorithms and data structures for a variety of fundamental problems 3 Conduct formal reasoning about problem complexity and algorithmic efficiency 4 Recognize the design techniques of standard algorithms, and apply these techniques to develop new computational solutions to problems
Assessment:	Project work during semester due around weeks 6 and 11, expected to take approximately 25 - 30 hours of work (20%). Addresses all Intended Learning Ootcomes, (ILOs) 1 - 4. A written 50-minute test (ILOs 1, 3, and 4), around week 7 (10%) A written 3-hour closed book examination (70%) Hurdle requirements: The examination is a hurdle and must be passed to pass the subject The student must also successfully complete at least eight of the weekly online quizzes.
Prescribed Texts:	A. Levitin, Introduction to the Design and Analysis of Algorithms, Pearson, 3rd edition, 2012
Breadth Options:	This subject is not available as a breadth subject.
Fees Information:	Subject EFTSL, Level, Discipline & Census Date, http://enrolment.unimelb.edu.au/fees
Generic Skills:	On completion of this subject students should have the following skills: # Application of knowledge of basic science and engineering fundamentals # Effective communication about computational efficiency # Capacity to reason and solve problems # Ability to undertake problem identification, formulation and solution # Capacity for creativity and innovation # Profound respect for truth and intellectual integrity, and for the ethics of scholarship.
Notes:	**LEARNING AND TEACHING METHODS** The subject involves two weekly one -hour lectures and one tutorial class. The lectures are a mix of direct delivery and interactive student problem solving. Although written assignments are submitted by students individually, in-plenum discussion of the problems is encouraged. **INDICATIVE KEY LEARNING RESOURCES** Students are provided with lecture slides, and links on the LMS to the in-house animated software *Algorithms in Action*. The slides are integrated with a well-established textbook. **CAREERS / INDUSTRY LINKS** With Big Data at the forefront of modern computing solutions, industry is ever-more focused on efficient computational analysis methods. Software engineers, developers and data analysts will find not only the analysis techniques, but also the fundamental algorithmic design concepts, highly applicable to the handling of significant datasets. Building on an initial connection in a similar undergraduate offering, there is scope for industry liaison with this subject.
Related Course(s):	Doctor of Philosophy - Engineering Master of Philosophy - Engineering Master of Science (Bioinformatics)
Related Majors/Minors/ Specialisations:	Approved Masters level subjects from other departments Computer Science Computer Science MIT Computing Specialisation

	MIT Distributed Computing Specialisation MIT Health Specialisation MIT Spatial Specialisation Master of Engineering (Software with Business) Master of Engineering (Software)