MONASH University
Information Technology

FIT1029
Algorithmic problem solving

Unit Guide

Semester 2, 2011

Last updated: 22 Aug 2011

Table of Contents

FIT1029 Algorithmic problem solving - Semester 2, 2011

Algorithms are recipes for solving a problem. They are fundamental to computer science and software engineering. Algorithms are the formal foundation of computer programming but also exist independently of computers as systematic problem-solving procedures. This unit introduces algorithmics, the study of algorithms. It is not about programming and coding but rather about understanding and analysing algorithms and about algorithmic problem-solving, i.e. the design of systematic problem-solving procedures. The unit is very hands-on and students will develop algorithms to solve a wide variety of different problems, working individually as well as together in groups and as a class.

The unit will not require any knowledge of a programming language. The initial instruction will be performed independently of any programming language and only use simple pseudo-code that will be developed from scratch in the unit. Various means of visualising algorithm execution (manipulating sets of tangible physical object, using turtle graphics, using algorithm visualisations) will be employed to enable the students to trace the execution of algorithms and to complement their formal understanding with an intuitive understanding. Later stages of the unit will make use of the coding knowledge developed in FIT1002 to demonstrate how pseudo-code algorithms can be mapped to concrete programs.

Topics include: What is a computational problem and what is an algorithm; Basic control structures; basic data structures; Modular Algorithm Structure; Recursion; Problem-solving strategies for algorithm development; Arguing correctness of an algorithm; Arguing termination of an algorithm; Understanding the efficiency of an algorithm; and Limitations of algorithms.

Mode of Delivery

Clayton (Day)

Contact Hours

2 hrs lectures/wk, 2 hrs tutorials/wk

Workload

For on campus students, workload commitments per week are:

- two-hour lecture and
- two-hour tutorial (requiring advance preparation)
- a minimum of 2-3 hours of personal study per one hour of contact time in order to satisfy the reading and assignment expectations.
- You will need to allocate up to 2 hours per week in some weeks, for use of a computer, including time for newsgroups/discussion groups.

Unit Relationships

Prerequisites

Only for students in the Bachelor of Computer Science and Bachelor of Software Engineering, associated Double Degrees and major/minor sequences. Exceptions can be approved by the unit leader after assessment of mathematical background knowledge.

Chief Examiner

Dr David Albrecht

Campus Lecturer

Clayton

Peter Tischer

Contact hours: TBA

Academic Overview

Learning Objectives

At the completion of this unit students will have -
A knowledge and understanding of:

- the difference between algorithms and processes;
- basic ways to structure algorithms: basic data structures (simple variables, collections structure, specifically vectors, lists, sets, and tables); basic control structures (sequence, choice, iteration);
- recursion;
- modular algorithm structures;
- the equivalence of recursion and iteration;
- problem solving strategies suitable for algorithm development including top-down design and bottom-up design;
- simple standard patterns for algorithms (eg traversal, search);
- what makes a good algorithm
- limitations of algorithms (high level).

Developed the skills to:

- develop simple iterative and recursive algorithms
- argue the correctness of simple algorithms
- judge the efficiency of simple algorithms, and

Developed attitudes that enable them to:

- value clear specification of problems;
- understand the relation between algorithms and programs;
- appreciate the value of designing abstract algorithms before starting to code a program;
- have confidence that they can develop algorithms to solve computational problems;
- appreciate that seemingly difficult problems can have very simple elegant algorithmic solutions (and vice versa);
- value correctness arguments for algorithms; and
- value the importance of simplicity and efficiency.

Demonstrated the communication skills necessary to:

- solve a problem by discussing possible approaches and solutions as a team; and
- clearly communicate (the specification of) a computational problem, its algorithmic solution and arguments for correctness and efficiency.

Graduate Attributes

Monash prepares its graduates to be:

1. responsible and effective global citizens who:

a. engage in an internationalised world
b. exhibit cross-cultural competence
c. demonstrate ethical values

critical and creative scholars who:

a. produce innovative solutions to problems
b. apply research skills to a range of challenges
c. communicate perceptively and effectively

Assessment Summary

Examination (3 hours): 60%; In-semester assessment: 40%

Assessment Task	Value	Due Date
Assignment 1	10%	Monday 8 August 2011
Assignment 2	15%	Monday 5 September 2011
Assignment 3	15%	Monday 3 October 2011
Examination 1	60%	To be advised

Teaching Approach

Lecture and tutorials or problem classes

This teaching and learning approach provides facilitated learning, practical exploration and peer learning.

Feedback

Our feedback to You

Types of feedback you can expect to receive in this unit are:

- Informal feedback on progress in labs/tutes
- Graded assignments with comments

Your feedback to Us

Monash is committed to excellence in education and regularly seeks feedback from students, employers and staff. One of the key formal ways students have to provide feedback is through SETU, Student Evaluation of Teacher and Unit. The University's student evaluation policy requires that every unit is evaluated each year. Students are strongly encouraged to complete the surveys. The feedback is anonymous and provides the Faculty with evidence of aspects that students are satisfied and areas for improvement.

For more information on Monash's educational strategy, and on student evaluations, see:
http://www.monash.edu.au/about/monash-directions/directions.html
http://www.policy.monash.edu/policy-bank/academic/education/quality/student-evaluation-policy.html

Previous Student Evaluations of this unit

If you wish to view how previous students rated this unit, please go to https://emuapps.monash.edu.au/unitevaluations/index.jsp

Recommended Resources

Unit Schedule

Week	Activities	Assessment
0		No formal assessment or activities are undertaken in week 0
1	Introduction to the unit and the type of problems	
2	Understanding and modelling the problem	
3	Invariants in problems and data	Assignment 1 due Monday, 8 August 2011
4	Decomposition of problems and applying Brute Force to solve problems	
5	Using abstraction, symmetry, heuristics and divide and conquer to simplify problems	
6	Recursion	
7	Backtracking	Assignment 2 due Monday, 5 September 2011
8	Dynamic Programming	
9	Fundamentals	
10	Abstract Data Types and Correctness	Assignment 3 due Monday, 3 October 2011
11	Complexity	
12	Limitations of algorithms	
	SWOT VAC	No formal assessment is undertaken SWOT VAC
	Examination period	LINK to Assessment Policy: http://policy.monash.edu.au/policy-bank/academic/education/assessment/assessment-in-coursework-policy.html

*Unit Schedule details will be maintained and communicated to you via your MUSO (Blackboard or Moodle) learning system.

Assessment Requirements

Assessment Policy

To pass a unit which includes an examination as part of the assessment a student must obtain:

- 40% or more in the unit's examination, and
- 40% or more in the unit's total non-examination assessment, and
- an overall unit mark of 50% or more.

If a student does not achieve 40% or more in the unit examination or the unit non-examination total assessment, and the total mark for the unit is greater than 50% then a mark of no greater than 49-N will be recorded for the unit

Assessment Tasks

Participation

- **Assessment task 1**

  Title:
  Assignment 1
  Description:
  This assignment will aim to help you understand how to go about finding algorithms to solve problems.
  Weighting:
  10%
  Criteria for assessment:
  Detailed assessment criteria will be issued along with the assignment.

  1. All assumptions should be stated.
  2. All algorithms must meet the problem specification.
  3. Students should be able to answer questions about their own work.

  Due date:
  Monday 8 August 2011

- **Assessment task 2**

  Title:
  Assignment 2
  Description:
  This assignment will aim to help you the importance of fundamental concepts, such as invariants, divide and conquer, induction in developing algorithms
  Weighting:
  15%
  Criteria for assessment:
  Detailed assessment criteria will be issued along with the assignment.

  1. All assumptions should be stated.
  2. All algorithms must meet the problem specification.
  3. Students should be able to answer questions about their own work.

  Due date:

Monday 5 September 2011

- **Assessment task 3**

Title:
Assignment 3
Description:
This assignment will help you understand different search techniques. It will also help you communicate and reason about algorithms.
Weighting:
15%
Criteria for assessment:
Detailed assessment criteria will be issued along with the assignment.

1. All assumptions should be stated.
2. All algorithms must meet the problem specification.
3. Students should be able to answer questions about their own work.

Due date:
Monday 3 October 2011

Examinations

- **Examination 1**

Weighting:
60%
Length:
3 hours
Type (open/closed book):
Closed book
Electronic devices allowed in the exam:
None

Assignment submission

It is a University requirement (http://www.policy.monash.edu/policy-bank/academic/education/conduct/plagiarism-procedures.html) for students to submit an assignment coversheet for each assessment item. Faculty Assignment coversheets can be found at http://www.infotech.monash.edu.au/resources/student/forms/. Please check with your Lecturer on the submission method for your assignment coversheet (e.g. attach a file to the online assignment submission, hand-in a hard copy, or use an online quiz).

Extensions and penalties

Submission must be made by the due date otherwise penalties will be enforced.

You must negotiate any extensions formally with your campus unit leader via the in-semester special consideration process:
http://www.infotech.monash.edu.au/resources/student/equity/special-consideration.html.

Returning assignments

Students can expect assignments to be returned within two weeks of the submission date or after receipt, whichever is later

Other Information

Policies

Monash has educational policies, procedures and guidelines, which are designed to ensure that staff and students are aware of the University's academic standards, and to provide advice on how they might uphold them. You can find Monash's Education Policies at: http://policy.monash.edu.au/policy-bank/academic/education/index.html

Key educational policies include:

- Plagiarism (http://www.policy.monash.edu/policy-bank/academic/education/conduct/plagiarism-policy.html)
- Assessment (http://www.policy.monash.edu/policy-bank/academic/education/assessment/assessment-in-coursework-po
- Special Consideration (http://www.policy.monash.edu/policy-bank/academic/education/assessment/special-consideration-policy.h
- Grading Scale (http://www.policy.monash.edu/policy-bank/academic/education/assessment/grading-scale-policy.html)
- Discipline: Student Policy (http://www.policy.monash.edu/policy-bank/academic/education/conduct/student-discipline-policy.html)
- Academic Calendar and Semesters (http://www.monash.edu.au/students/key-dates/);
- Orientation and Transition (http://www.infotech.monash.edu.au/resources/student/orientation/); and
- Academic and Administrative Complaints and Grievances Policy (http://www.policy.monash.edu/policy-bank/academic/education/management/complaints-grievance-policy.
- Codes of Practice for Teaching and Learning (http://www.policy.monash.edu.au/policy-bank/academic/education/conduct/suppdocs/code-of-practice-tea

Student services

The University provides many different kinds of support services for you. Contact your tutor if you need advice and see the range of services available at www.monash.edu.au/students The Monash University Library provides a range of services and resources that enable you to save time and be more effective in your learning and research. Go to http://www.lib.monash.edu.au or the library tab in my.monash portal for more information. Students who have a disability or medical condition are welcome to contact the Disability Liaison Unit to discuss academic support services. Disability Liaison Officers (DLOs) visit all Victorian campuses on a regular basis

- Website: http://adm.monash.edu/sss/equity-diversity/disability-liaison/index.html;
- Telephone: 03 9905 5704 to book an appointment with a DLO;
- Email: dlu@monash.edu
- Drop In: Equity and Diversity Centre, Level 1 Gallery Building (Building 55), Monash University, Clayton Campus.

Reading List:

1. Levitin, A., *Introduction to the Design and Analysis of Algorithms (2nd Edition)*, Addison-Wesley, 2006
2. Harel, D. with Y. Feldman, *Algorithmics: The Spirit of Computing,* 3rd ed., Pearson Education Limited, 2004.
3. Michalewicz, Z. and M. Michalewicz, *Puzzle-Based Learning: An introduction to critical thinking,*

mathematics, and problem solving, Hybrid Publishers, 2008.
4. Polya, G., *How to solve it; a new aspect of mathematical method,* 2nd ed., Garden City, N.Y., Doubleday, 1957
5. Bentley, J., *Programming Pearls,* Addison-Wesley, 1986
6. Bentley, J., *More Programming Pearls: confessions of a coder,* Addison-Wesley, 1988
7. Skiena, S., *The Algorithm Design Manual, TELOS--the Electronic Library of Science,* 1998
8. Cormen, T., C.E. Leiserson, R.L. Rivest, and C. Stein, *Introduction to Algorithms*, The MIT Press, 1990