MONASH University
Information Technology

FIT2004
Algorithms and data structures

Unit Guide

Semester 2, 2011

The information contained in this unit guide is correct at time of publication. The University has the right to change any of the elements contained in this document at any time.

Last updated: 22 Aug 2011

Table of Contents

FIT2004 Algorithms and data structures - Semester 2, 2011

This unit introduces students to problem solving concepts and techniques fundamental to the science of programming. In doing this it covers problem specification, algorithmic design, analysis and implementation. Detailed topics include analysis of best, average and worst-case time and space complexity; introduction to numerical algorithms; recursion; advanced data structures such as heaps and B-trees; hashing; sorting algorithms; searching algorithms; graph algorithms; and numerical computing.

Mode of Delivery

Clayton (Day)

Contact Hours

2 hrs lectures/wk, 3 hr laboratory/fortnight, 1 hr tutorial/fortnight

Workload

Students will be expected to spend a total of 12 hours per week during semester on this unit as follows:

- Lectures: 2 hours per week
- Tutorials/Lab Sessions:4 hours **fortnightly** (1 hour tutorial, 3 hours lab)
- Reading: 4 hours per week
- Lab Preparation: 4 hours per week.

Unit Relationships

Prohibitions

CSE2304, FIT2009, FIT2071

Prerequisites

One of FIT1008, FIT1015 or CSE1303 and 6 points of Level 1 mathematics.

Chief Examiner

Professor Geoff Webb

Campus Lecturer

Clayton

Reza Haffari

Tutors

Clayton

Phillip Abramson

Tianyu Long

Academic Overview

Learning Objectives

At the completion of this unit students will have:

- understanding of a formal specification;
- ability to create a formal specification for an informal problem;
- knowledge and understanding of algorithmic properties such as correctness, termination and complexity;
- ability to, given a non-trivial algorithm, formally prove certain properties, such as correctness and termination;
- ability, given a non-trivial algorithm, to determine its best, average and worst-case, time and space-complexity;
- knowledge and understanding of reasonably complex data structures such as minimum spanning trees, and Directed and Undirected, Weighted and Unweighted Graphs;
- ability to design and implement new non-trivial algorithms using complex data structures;
- knowledge of and ability to use algorithmic paradigms such as divide and conquer, greedy, dynamic programming and so on;
- ability to identify these paradigms in diverse algorithms;
- knowledge and understanding of the issues involved in implementing a non-trivial algorithm efficiently.

Developed attitudes that enable them to:

- carefully design and/or analyse the algorithms they are using in order to verify important properties such as correctness, termination, and complexity;
- identify the key features of a brief informal problem description and abstract the underlying formal problem.

Developed the skills to:

- create their own data structures.
- create a new algorithm to solve a new problem.
- make a formal argument about desirable properties of the solution.
- adapt an existing algorithm and/or data-structure where that is possible and appropriate.
- implement a non-trivial algorithm efficiently.

Demonstrated the communication skills necessary to:

- make a formal argument that an algorithm and/or data-structure has a given property, such as correctness, termination or complexity.

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): 70%; In-semester assessment: 30%

Assessment Task	Value	Due Date
ADTs, Proofs & Induction	6 %	Week 4
Complexity Analysis	6 %	Week 6
Dynamic Programming	6 %	Week 8
Trees	6 %	Week 10
Graphs & Graph Algorithms	6 %	Week 12
Examination 1	70%	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 without comments
- Solutions to tutes, labs and assignments

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

Required Resources

Java (latest version) installed in the labs, you can download a free copy from Sun Microsystems.

Unit Schedule

Week	Activities	Assessment
0		No formal assessment or activities are undertaken in week 0
1	Specification & Abstract Data Types	
2	Proofs & Induction	
3	Complexity Analysis I	
4	Complexity Analysis II	Assignment 1: ADTs, Proofs and Induction
5	Pattern Matching	
6	Dynamic Programming	Assignment 2: Complexity Analysis
7	Dynamic & Balanced Trees	
8	Amortized Analysis	Assignment 3: Dynamic Programming
9	Multi-way Trees	
10	Graphs	Assignment 4: 2-4 Trees, AVL Trees, Skip Lists.
11	Path Problems	
12	Flow Problems	Assignment 5: Graphs and Graph 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:
  ADTs, Proofs & Induction
  Description:
  Abstract data types, proofs and induction.
  Weighting:
  6 %
  Criteria for assessment:

  ♦ Demonstrate code where applicable
  ♦ explain your solutions

  Due date:
  Week 4

- **Assessment task 2**

  Title:
  Complexity Analysis
  Description:
  The time complexity analyss for several algorithms.
  Weighting:
  6 %
  Criteria for assessment:

  ♦ Demonstrate code where applicable
  ♦ explain your solutions

  Due date:
  Week 6

- **Assessment task 3**

  Title:
  Dynamic Programming
  Description:
  Dynamic programming for a game, and string pattern matching.
  Weighting:
  6 %
  Criteria for assessment:

  - ♦Demonstrate code where applicable
  - ♦explain your solutions

  Due date:
  Week 8

- **Assessment task 4**

  Title:
  Trees
  Description:
  2-4 Trees, AVL Trees, Skip Lists.
  Weighting:
  6 %
  Criteria for assessment:

  - ♦Demonstrate code where applicable
  - ♦explain your solutions

  Due date:
  Week 10

- **Assessment task 5**

  Title:
  Graphs & Graph Algorithms
  Description:
  Implementing graphs and several graph algorithms.
  Weighting:
  6 %
  Criteria for assessment:

  - ♦Demonstrate code where applicable
  - ♦explain your solutions

  Due date:
  Week 12

Examinations

- **Examination 1**

  Weighting:
  70%
  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

Mark Allen Weiss:
Data Structures and Algorithm Analysis in Java. 2nd ed
Addison-Wesley.

Michael Goodrich and Roberto Tamassia.

Other Information

Data Structures and Algorithms in Java, 3rd ed
John Wiley.

Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein. Introduction to Algorithms, 2nd Edition, EHT MIT Press & McGraw Hill

Duane Bailey. Java Structures: Data Structures in Java for the Principled Programmer. International Edition, May 1999, Mc Graw Hill. This book is available for free online at http://www.cs.williams.edu/~bailey/JavaStructures/Welcome.html