MONASH University

FIT2004
Algorithms and data structures

Unit Guide

Semester 2, 2009

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 : 13 Jul 2009

Table of Contents

FIT2004 Algorithms and data structures - Semester 2, 2009

Chief Examiner:

Associate Professor Bernd Meyer
Associate Professor
Phone: +61 3 990 52240
Fax: +61 3 990 31077

Lecturer(s) / Leader(s):

Clayton

Associate Professor Bernd Meyer
Associate Professor
Phone: +61 3 990 52240
Fax: +61 3 990 31077

Malaysia

Dr Mohammed Belkhatir

Introduction

Welcome to FIT2004 "Algorithms and Data Structures". This unit comprises a vital part of the core knowledge of Computer Science and Algorithms: the design of efficient algorithms for non-trivial problems and the analysis of their complexity and correctness. The inclusion of algorithm and data structure design is what fundamentally distinguishes programming from coding. The concepts discussed in this unit are mostly treated on an abstract level and a reasonable level of profficiency with coding is assumed, so that students will be able to implement these abstract concepts in concrete programs.

Unit synopsis

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.

Learning outcomes

Upon successful completion of this unit, students will have:

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

Upon successful completion of this unit, students will be able to carefully design and/or analyse the algorithms they are using in order to verify important properties such as correctness, termination, and complexity.

Upon successful completion of this unit, students will be able to:

1. Identify the key features of a brief informal problem description and abstract the underlying formal problem;
2. Create their own data structures;
3. Create a new algorithm to solve a new problem;
4. Make a formal argument about desirable properties of the solution;
5. Adapt an existing algorithm and/or data-structure where that is possible and appropriate;
6. Implement a non-trivial algorithm efficiently;

Upon successful completion of this unit, students will be able to make a 'formal argument' that an algorithm and/or data-structure has a given property, such as correctness, termination or complexity.

Contact hours

2 hr lecture/week, 1 hr tutorial/fortnight, 3 hr lab/fortnight

Workload

2 hours of lecture

1 hour of tutorial (fortnightly)

3 hours of laboratory (fortnightly)

4 hours reading

4 hours laboratory preparation

Unit relationships

Prerequisites

One of CSE1303, FIT1008, FIT1015 and two of MAT1841, MAT1830, MTH1020 or MTH1030 or MTH1112

Prohibitions

CSC2040, CSE2304, DGS2131, FIT2009, RDT2131

Relationships

The unit is a second year core unit in the Bachelor of Computer Science and the Bachelor of Software Engineering. It may be taken as an elective in other programs where you have satisfied the prerequisites and course rules permit.

You may not study this unit and CSC2040, CS2304, DGS2131, RDT2131 in your degree.

Teaching and learning method

Lectures will be used to present new concepts, compare different approaches, analyse their advantages and disadvantages, and propose some general questions. The aim is to give students a first look at the concepts and challenge them to think further. Tutorials and practicals will be used to link the theory with practice and deepen the students understanding and practical abilities.

Timetable information

For information on timetabling for on-campus classes please refer to MUTTS, http://mutts.monash.edu.au/MUTTS/

Tutorial allocation

On-campus students should register for tutorials/laboratories using the Allocate+ system: http://allocate.cc.monash.edu.au/

Unit Schedule

Week	Topic	Tutorials	Labs	Key dates
1	Specification & Abstract Data Types	---		
2	Proofs & Induction	T1 ADTs		
3	Complexity Analysis I	T2 Proofs & Induction		
4	Complexity Analysis II	---	P1 ADTs	Assignment 1 due August 10
5	Pattern Matching	T3 Complexity Analysis		
6	Dynamic Programming	---	P2 Complexity Analysis	Assignment 2 due August 24
7	Dynamic & Balanced Trees	T4 Dynamic Programming		
8	Amortized Analysis	---	P3 Dynamic Programming	Assignment 3 due September 7
9	Multi-way Trees	T5 Trees		
10	Graphs	---	P4 Trees	Assignment 4 due September 21
Mid semester break				
11	Path Problems	T6 Graphs		
12	Flow Problems	---	P5 Graphs & Graph Algorithms	Assignment 5 due October 16
13	Revision	T7 Graph Algorithms		

Unit Resources

Prescribed text(s) and readings

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

Text books are available from the Monash University Book Shops. Availability from other suppliers cannot be assured. The Bookshop orders texts in specifically for this unit. You are advised to purchase your text book early.

Recommended text(s) and readings

Michael Goodrich and Roberto Tamassia.
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

Required software and/or hardware

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

Equipment and consumables required or provided

Students studying off-campus are required to have the minimum system configuration specified by the Faculty as a condition of accepting admission, and regular Internet access. On-campus students, and those studying at supported study locations may use the facilities available in the computing labs. Information about computer use for students is available from the ITS Student Resource Guide in the Monash University Handbook. You will need to allocate up to **5** hours per week for use of a computer, including time for newsgroups/discussion groups.

Study resources

Study resources we will provide for your study are:

The FIT2004 web site on MUSO, where lecture slides, weekly tutorial requirements, assignment specifications, sample solutions and supplementary material will be posted.

Assessment

Overview

Tutorials and practicals: 30%
Examination (3 hours): 70%.

Faculty 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 44% then a mark of no greater than 44-N will be recorded for the unit.

In a addition to a three hour closed book examination, the lab work in this unit is assessed(and is supposed to be prepared at home before each lab). To pass the unit you must:

- attempt at least 4 assignments (assessed in the labs)
- achieve no less that 50% of the possible marks in the non-examination assessment
- achieve no less than 50% of the possible marks in the examination

Assignment tasks

Assignment coversheets

Assignment coversheets are available via "Student Forms" on the Faculty website:
http://www.infotech.monash.edu.au/resources/student/forms/
You MUST submit a completed coversheet with all assignments, ensuring that the plagiarism declaration section is signed.

Assignment submission and return procedures, and assessment criteria will be specified with each assignment.

- **Assignment task 1**

  Title:
  ADTs, Proofs & Induction
  Description:
  specific tasks and marking criteria will be distributed at the appropriate time during the semester
  Weighting:
  6 %
  Due date:
  August 10

- **Assignment task 2**

  Title:
  Complexity Analysis
  Description:
  specific tasks and marking criteria will be distributed at the appropriate time during the semester
  Weighting:
  6 %
  Due date:
  August 24

- **Assignment task 3**

  Title:
  Dynamic Programming
  Description:
  specific tasks and marking criteria will be distributed at the appropriate time during the semester
  Weighting:
  6 %
  Due date:
  September 7

- **Assignment task 4**

  Title:
  Trees
  Description:
  specific tasks and marking criteria will be distributed at the appropriate time during the semester
  Weighting:
  6 %
  Due date:
  September 21

- **Assignment task 5**

  Title:
  Graphs & Graph Algorithms
  Description:
  specific tasks and marking criteria will be distributed at the appropriate time during the semester
  Weighting:
  6 %
  Due date:
  October 16

Examination

- **Weighting:** 70%
  Length: 3 hours
  Type (open/closed book): Closed book

See Appendix for End of semester special consideration / deferred exams process.

Due dates and extensions

Please make every effort to submit work by the due dates. It is your responsibility to structure your study program around assignment deadlines, family, work and other commitments. Factors such as normal work pressures, vacations, etc. are not regarded as appropriate reasons for granting extensions. Students are advised to NOT assume that granting of an extension is a matter of course.

Students requesting an extension for any assessment during semester (eg. Assignments, tests or presentations) are required to submit a Special Consideration application form (in-semester exam/assessment task), along with original copies of supporting documentation, directly to their lecturer within two working days before the assessment submission deadline. Lecturers will provide specific outcomes directly to students via email within 2 working days. The lecturer reserves the right to refuse late applications.

A copy of the email or other written communication of an extension must be attached to the assignment submission.

Refer to the Faculty Special consideration webpage or further details and to access application forms: http://www.infotech.monash.edu.au/resources/student/equity/special-consideration.html

Late assignment

Assignments received after the due date will be subject to a penalty of **10% per day of late submission. No submissions will be accepted later than 1 week after the due date.**

Return dates

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

Appendix

Please visit the following URL: http://www.infotech.monash.edu.au/units/appendix.html for further information about:

- Continuous improvement
- Unit evaluations
- Communication, participation and feedback
- Library access
- Monash University Studies Online (MUSO)
- Plagiarism, cheating and collusion
- Register of counselling about plagiarism
- Non-discriminatory language
- Students with disability
- End of semester special consideration / deferred exams