FIT2004
Algorithms and data structures

Unit guide

Semester 2, 2008

Last updated : 27 Jun 2008

Table of Contents

FIT2004 Algorithms and data structures - Semester 2 , 2008

Unit leader :

Bernd Meyer

Lecturer(s) :

Clayton

- Bernd Meyer

Malaysia

- 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

ASCED: 020109 Algorithms

FIT2004 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; search algorithms; and graph algorithms. Numerical computing and geometric computing may be introduced to complement these algorithms on discrete structures.

Learning outcomes

Knowledge and Understanding

K1. Understand a formal specification.

K2. Ability to create a formal specification for an informal problem.

K3. Knowledge and understanding of algorithmic properties such as correctness, termination and complexity.

K4. Ability to, given a non-trivial algorithm, formally prove certain properties, such as correctness and termination.

K5. Ability, given a non-trivial algorithm, to determine its best- average- and worst-case, time- and space-complexity.

K6. Knowldege and understanding of reasonably complex data structures such as minimum spanning trees, and Directed and Undirected, Weighted and Unweighted Graphs.

K7. Ability to design and implement new non-trivial algorithms using complex data structures.

K8. Knowledge of and ability to use algorithmic paradigms such as divide and conquer, greedy, dynamic programming and so on.

K9. Ability to identify these paradigms in diverse algorithms.

K10. Knowledge and understanding of the issues involved in implementing a non-trivial algorithm efficiently.

Attitudes, Values and Beliefs

A1. Carefully design and/or analyse the algorithms they are using in order to verify important properties such as correctness, termination, and complexity.

FIT2004 Algorithms and data structures - Semester 2 , 2008

Practical Skills

P1. Identify the key features of a brief informal problem description and abstract the underlying formal problem.

P2. Create their own data structures.

P3. Create a new algorithm to solve a new problem.

P4. Make a formal argument about desirable properties of the solution.

P5. Adapt an existing algorithm and/or data-structure where that is possible and appropriate.

P6. Implement a non-trivial algorithm efficiently.

Relationships, Communication and TeamWork

R1. Conduct a 'formal argument' that an algorithm and/or data-structure has a given property, such as correctness, termination or complexity.

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

Before attempting this unit you must have satisfactorily completed

FIT1008 Computer Science or FIT1015 Computer Science or CSE1303 Computer Science AND 2 of MAT1841 Mathematics for Computer Science I, MAT1830 Mathematics for Computer Science II (or equivalent), MTH1020 Analysis of change, MTH1030 Techniques for modelling, MTH1112 Numbers, logic and graphs, MTH2010 Multivariable calculus. Note that students entering with CSE1303 ideally should have done some Java (such as FIT1002) although this is not a formal prerequisite.

Students beginning FIT2004 are assumed to:

- Understand basic programming features such as: iteration, nested loops, conditionals, arrays, pointers, functions, parameters, and recursion.
- Understand simple algorithm complexity (e.g., linear, quadratic, logarithmic and exponential).
- Be able to analyse the complexity of algorithms with a difficulty similar to those introduced in FIT1008 Computer Science (e.g., mergesort and quicksort)
- Be able to design new and efficient algorithms of reasonable difficulty (e.g., simple search and sort algorithms) using such programming features. A solution with a certain time- and/or space-complexity may be required.

- Be able to efficiently implement a reasonably difficult algorithm in the designated language.
- Full understand simple data structures, such as doubly-linked lists
- A high level of competency is expected from students in their ability to follow the above process, i.e. their ability to, given an English and/or mathematical description of a new problem of a suitable level of difficulty, be able to identify the underlying abstract problem, devise an efficient algorithm to solve the problem, and turn the algorithm into an efficient piece of code in the designated programming language.
- Be fluent with the terms and concepts of elementary propositional logic, predicate logic and set theory, proof, proof by contradiction and by induction.
- Understand and be able to define and manipulate basic functions such as log, exp, polynomials, arithmetic and geometric series, sum and product, permutations, derivatives.
- Manipulate and write such mathematical formulae, write computer programs involving them, give definitions for them.
- Understand computer programs, English sentences & descriptions, and mathematical formulae involving them.
- Have informal working knowledge of the syntax of arithmetic and boolean expressions, and their associate representation in the designated programming language.

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.

Continuous improvement

Monash is committed to 'Excellence in education' and strives for the highest possible quality in teaching and learning. To monitor how successful we are in providing quality teaching and learning Monash regularly seeks feedback from students, employers and staff. Two of the formal ways that you are invited to provide feedback are through Unit Evaluations and through Monquest Teaching Evaluations.

One of the key formal ways students have to provide feedback is through Unit Evaluation Surveys. It is Monash policy for every unit offered to be evaluated each year. Students are strongly encouraged to complete the surveys as they are an important avenue for students to "have their say". The feedback is anonymous and provides the Faculty with evidence of aspects that students are satisfied and areas for improvement.

Student Evaluations

The Faculty of IT administers the Unit Evaluation surveys online through the my.monash portal, although for some smaller classes there may be alternative evaluations conducted in class.

If you wish to view how previous students rated this unit, please go to
http://www.monash.edu.au/unit-evaluation-reports/

Over the past few years the Faculty of Information Technology has made a number of improvements to its courses as a result of unit evaluation feedback. Some of these include systematic analysis and planning of unit improvements, and consistent assignment return guidelines.

Monquest Teaching Evaluation surveys may be used by some of your academic staff this semester. They are administered by the Centre for Higher Education Quality (CHEQ) and may be completed in class with a facilitator or on-line through the my.monash portal. The data provided to lecturers is completely anonymous.
Monquest surveys provide academic staff with evidence of the effectiveness of their teaching and identify areas for improvement. Individual Monquest reports are confidential, however, you can see the summary results of Monquest evaluations for 2006 at http://www.adm.monash.edu.au/cheq/evaluations/monquest/profiles/index.html

Improvements to this unit

introduction of sample exams into the revision week.

restructured tutorial/practical sequence.

Unit staff - contact details

Unit leader

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

Lecturer(s) :

Associate Professor Bernd Meyer
Associate Professor
Phone +61 3 990 52240
Fax +61 3 990 31077
Dr Mohammed Belkhatir

FIT2004 Algorithms and data structures - Semester 2 , 2008

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 at all Australian campuses please refer to MUTTS, http://mutts.monash.edu.au/MUTTS/

timetables for Malaysia are also available on MUTTS

Tutorial allocation

On-campus students should register for tutorials/laboratories using Allocate+

Communication, participation and feedback

Monash aims to provide a learning environment in which students receive a range of ongoing feedback throughout their studies. You will receive feedback on your work and progress in this unit. This may take the form of group feedback, individual feedback, peer feedback, self-comparison, verbal and written feedback, discussions (on line and in class) as well as more formal feedback related to assignment marks and grades. You are encouraged to draw on a variety of feedback to enhance your learning.

It is essential that you take action immediately if you realise that you have a problem that is affecting your study. Semesters are short, so we can help you best if you let us know as soon as problems arise. Regardless of whether the problem is related directly to your progress in the unit, if it is likely to interfere with your progress you should discuss it with your lecturer or a Community Service counsellor as soon as possible.

Unit Schedule

Week	Topic	Key dates
1	Specification & Abstract Data Types	
2	Proofs & Induction I	
3	Proofs & Induction II	
4	Complexity Analysis I	Assignment 1 due August 4
5	Complexity Analysis II	
6	Dynamic Programming	Assignment 2 due August 18
7	Dynamic & Balanced Trees	
8	Amortized Analysis	Assignment 3 due September 1
9	Multi-way Trees	
10	Graphs	Assignment 4 due September 15

11	Path Problems	
Mid semester break		
12	Flow Problems	Assignment 5 due October 6
13	Revision	

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 <u>Monash University Book Shops</u>. 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.

Library access

The Monash University Library site contains details about borrowing rights and catalogue searching. To learn more about the library and the various resources available, please go to <u>http://www.lib.monash.edu.au</u>. Be sure to obtain a copy of the Library Guide, and if necessary, the instructions for remote access from the library website.

Monash University Studies Online (MUSO)

All unit and lecture materials are available through MUSO (Monash University Studies Online). Blackboard is the primary application used to deliver your unit resources. Some units will be piloted in Moodle. If your unit is piloted in Moodle, you will see a link from your Blackboard unit to Moodle (http://moodle.monash.edu.au) and can bookmark this link to access directly. In Moodle, from the Faculty of Information Technology category, click on the link for your unit.

You can access MUSO and Blackboard via the portal: http://my.monash.edu.au

Click on the Study and enrolment tab, then Blackboard under the MUSO learning systems.

In order for your Blackboard unit(s) to function correctly, your computer needs to be correctly configured.

For example:

- Blackboard supported browser
- Supported Java runtime environment

For more information, please visit: http://www.monash.edu.au/muso/support/students/downloadables-student.html

You can contact the MUSO Support by: Phone: (+61 3) 9903 1268

For further contact information including operational hours, please visit:
http://www.monash.edu.au/muso/support/students/contact.html

Further information can be obtained from the MUSO support site:
http://www.monash.edu.au/muso/support/index.html

Assessment

Unit assessment policy

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 6 assignments (assessed in the labs)
- achieve no less that 50% of the possible marks in assessed lab work
- achieve no less than 50% of the possible marks in the exam

Assignment tasks

- **Assignment Task**

  Title : ADTs, Proofs & Induction

  Description :

  specific tasks and marking criteria will be distributed at the appropriate time during the semester

  Weighting : 5 %

  Criteria for assessment :

  specific tasks and marking criteria will be distributed at the appropriate time during the semester

  Due date : August 4
- **Assignment Task**

  Title : Complexity

  Description :

  specific tasks and marking criteria will be distributed at the appropriate time during the semester

  Weighting : 5 %

  Criteria for assessment :

  specific tasks and marking criteria will be distributed at the appropriate time during the semester

  Due date : August 18
- **Assignment Task**

  Title : Dynamic Programming

  Description :

  specific tasks and marking criteria will be distributed at the appropriate time during the semester

  Weighting : 5 %

FIT2004 Algorithms and data structures - Semester 2 , 2008

Criteria for assessment :

specific tasks and marking criteria will be distributed at the appropriate time during the semester

Due date : September 1

- **Assignment Task**

Title : Trees

Description :

specific tasks and marking criteria will be distributed at the appropriate time during the semester

Weighting : 5 %

Criteria for assessment :

specific tasks and marking criteria will be distributed at the appropriate time during the semester

Due date : September 15

- **Assignment Task**

Title : Graph Algorithms

Description :

specific tasks and marking criteria will be distributed at the appropriate time during the semester

Weighting : 5 %

Criteria for assessment :

specific tasks and marking criteria will be distributed at the appropriate time during the semester

Due date : October 6

Examinations

- **Examination**

Weighting : 70%

Length : 3 hours

Type (open/closed book) : Closed book

Assignment submission

Assignments will be submitted by **mixed electronic and paper** submission**.** On-campus Students Submit the assignment in the pracs, with the appropriate cover sheet correctly filled out and attached Off Campus (OCL) students [OCL only] Mail your assignment to the Off-Campus Learning Centre with the cover sheet attached. Singapore and Hong Kong Students [Gippsland only] Mail your assignment to the Distance Education Centre with the cover sheet attached. Do not email submissions. The due date is the date by which the submission must be

received/the date by which the the submission is to be posted.

University and Faculty policy on assessment

Due dates and extensions

The due dates for the submission of assignments are given in the previous section. 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 seldom regarded as appropriate reasons for granting extensions. Students are advised to NOT assume that granting of an extension is a matter of course.

Requests for any other extension must be made to the unit lecturer or head tutor at your campus at least two days before the due date. You will be asked to forward original medical certificates in cases of illness, and may be asked to provide other forms of documentation where necessary. A copy of the email or other written communication of an extension must be attached to the assignment submission.

You are, however, generally allowed to submit your work in your allocated laboratory during the week of the due date even if you laboratory is after that due date.

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.

Assessment for the unit as a whole is in accordance with the provisions of the Monash University Education Policy at http://www.policy.monash.edu/policy-bank/academic/education/assessment/

We will aim to have assignment results made available to you within two weeks after assignment receipt.

Plagiarism, cheating and collusion

Plagiarism and cheating are regarded as very serious offences. In cases where cheating has been confirmed, students have been severely penalised, from losing all marks for an assignment, to facing disciplinary action at the Faculty level. While we would wish that all our students adhere to sound ethical conduct and honesty, I will ask you to acquaint yourself with Student Rights and Responsibilities (http://www.infotech.monash.edu.au/about/committees-groups/facboard/policies/studrights.html) and the Faculty regulations that apply to students detected cheating as these will be applied in all detected cases.

In this University, cheating means seeking to obtain an unfair advantage in any examination or any other written or practical work to be submitted or completed by a student for assessment. It includes the use, or attempted use, of any means to gain an unfair advantage for any assessable work in the unit, where the means is contrary to the instructions for such work.

When you submit an individual assessment item, such as a program, a report, an essay, assignment or other piece of work, under your name you are understood to be stating that this is your own work. If a submission is identical with, or similar to, someone else's work, an assumption of cheating may arise. If you are planning on working with another student, it is acceptable to undertake research together, and discuss problems, but it is not acceptable to jointly develop or share solutions unless this is specified by your lecturer.

Intentionally providing students with your solutions to assignments is classified as "assisting to cheat" and students who do this may be subject to disciplinary action. You should take reasonable care that your solution is not accidentally or deliberately obtained by other students. For example, do not leave copies of your work in progress on the hard drives of shared computers, and do not show your work to other students. If you believe this may have happened, please be sure to contact your lecturer as soon as possible.

Cheating also includes taking into an examination any material contrary to the regulations, including any bilingual dictionary, whether or not with the intention of using it to obtain an advantage.

Plagiarism involves the false representation of another person's ideas, or findings, as your own by either copying material or paraphrasing without citing sources. It is both professional and ethical to reference clearly the ideas and information that you have used from another writer. If the source is not identified, then you have plagiarised work of the other author. Plagiarism is a form of dishonesty that is insulting to the reader and grossly unfair to your student colleagues.

Register of counselling about plagiarism

The university requires faculties to keep a simple and confidential register to record counselling to students about plagiarism (e.g. warnings). The register is accessible to Associate Deans Teaching (or nominees) and, where requested, students concerned have access to their own details in the register. The register is to serve as a record of counselling about the nature of plagiarism, not as a record of allegations; and no provision of appeals in relation to the register is necessary or applicable.

Non-discriminatory language

The Faculty of Information Technology is committed to the use of non-discriminatory language in all forms of communication. Discriminatory language is that which refers in abusive terms to gender, race, age, sexual orientation, citizenship or nationality, ethnic or language background, physical or mental ability, or political or religious views, or which stereotypes groups in an adverse manner. This is not meant to preclude or inhibit legitimate academic debate on any issue; however, the language used in such debate should be non-discriminatory and sensitive to these matters. It is important to avoid the use of discriminatory language in your communications and written work. The most common form of discriminatory language in academic work tends to be in the area of gender inclusiveness. You are, therefore, requested to check for this and to ensure your work and communications are non-discriminatory in all respects.

Students with disabilities

Students with disabilities that may disadvantage them in assessment should seek advice from one of the following before completing assessment tasks and examinations:

- Faculty of Information Technology Student Service staff, and / or
- your Unit Coordinator, or
- Disabilities Liaison Unit

Deferred assessment and special consideration

Deferred assessment (not to be confused with an extension for submission of an assignment) may be granted in cases of extenuating personal circumstances such as serious personal illness or bereavement. Information and forms for Special Consideration and deferred assessment applications are available at http://www.monash.edu.au/exams/special-consideration.html. Contact the Faculty's Student Services staff at your campus for further information and advice.