School of Computing and Information Systems
COMP20007 Design of Algorithms
Subject Guide, Semester 1, 2020

Staff

Lecturer & Coordinator
Lars Kulik & Daniel Beck
`http://lars.kuliks.net` & `https://cis.unimelb.edu.au/people/d-beck`
Email: lkulik@unimelb.edu.au & d.beck@unimelb.edu.au
Consultation by appointment – please contact us by email.

Head tutor
Tobias Edwards
Email: tobias.edwards@unimelb.edu.au, regular consultation in the Doug McDonell building. We will advise the times and the room soon.

Subject information

Textbook. There is no prescribed textbook for this subject. Most of the topics in this subject are covered by Levitin and Skiena.

- Anany Levitin. *Introduction to the Design and Analysis of Algorithms* Pearson, 2012.
- Steven S Skiena. *The Algorithm Design Manual* Springer, 2008.[1]
- You may be interested in purchasing a copy of T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein. *Introduction to Algorithms*, 2009, MIT Press. This covers much more than we will use in the course, but it is an outstanding reference for a lifetime of problem-solving with computers.

Syllabus. The indicative scope of the subject is as follows (we might cover a few topics such as compression only briefly). We will cover

- Array algorithms
- Complexity analysis: O, Ω, and Θ, the Master theorem
- Divide & conquer
- Graphs, DFS, BFS, Dijkstra
- Dynamic programming, string matching
- Hashing
- Search trees
- Minimum spanning trees, greedy algorithms
- Compression
- Decidability, NP-completeness

Please note that there are no lectures on Tuesday, 14 April and Thursday, 16 April (Easter break). There are also no tutorials on Friday, 10 April (Good Friday public holiday) and during the Easter break.

[1] Please download your own copy, and do not share by email or other means.

Prior knowledge. It is assumed that you have taken COMP10002 (or equivalent) before this subject. You should already know

- How to code in C in a UNIX-style environment, including: arrays, `#define`, `struct`, pointers and `malloc`, `make`, and so on;
- Basic big-O notation, and the concept of counting fundamental operations as a method of analysing algorithms; and
- Elementary data structures including: arrays, stacks, queues, linked lists, and binary search trees.

Assessment

Assignments (30%) There will be two assignments (available on LMS).

- Assignment 1 (worth 10%), (week 5 or 6).
- Assignment 2 (worth 20%), (Monday in week 12).

Late submissions. Project work is due at a time that will be made explicit on the project specification. The late penalty is 20% of the available marks for that project for each day (or part thereof) overdue. Requests for extensions on medical grounds will need to be supported by a medical certificate. In general, extensions will not be granted if the interruption covers less than 10% of the project duration. Remember that labs are often heavily loaded near project deadlines, and unexpected outages can occur; in general, these will not be considered as grounds for an extension.

Students who experience difficulties due to personal circumstances are encouraged to make use of the appropriate University student support services, and to contact the lecturer, at the earliest opportunity. No student will have more than one submission marked for each project. If you make both on-time and late submissions, please see the lecturer as soon as possible to determine which submission will be assessed.

The University provides a range of support services for students experiencing personal difficulties, and in some cases can assist with management of study load. Please make use of these services if you have any concern about the impact of outside factors on your ability to undertake your studies.

Mid-semester test. Normally, there would be a forty-minute mid-semester test worth 10% of your final mark, in week 5 or 6. However, due to the travel restrictions in relation to the coronavirus (COVID-19), we will not be holding a mid-semester test this semester. Instead, we will be offering a take-home mid-semester test for all students that we will be giving feedback on. However, it will not count towards your grades. “This means the final exam will be worth 70% of your final grade, rather than 60%. We encourage all students to hand in their take-home mid-semester tests for feedback and to gauge their learning. The details will be made available closer to the date.

Examination (70%). There will be a 3-hour written exam at the end of semester.

Hurdles. To pass the subject, students must obtain at least:

- 50% in assignments (total $\geq 15/30$); and
- 50% in the exam (total $\geq 35/70$);
- Hence at least 50% overall.

Academic honesty. All work is to be done on an individual basis. In cases of cheating, both parties–receiver and giver–will be referred to the School of Engineering under the University Discipline procedures. For further information, see `http://academichonesty.unimelb.edu.au/`. Note that C source code counts as “written work” under the University’s definition of plagiarism.

Where it is stated on the assignment specification, you can use code from the Web in assignment submissions (such as Stack Overflow and Wikipedia, which are a rich resource for this subject material), but proper attribution (citation) must be given. State clearly in comments in your source code as close as feasible to the copied code where it was taken from, and why. If you use copied source code and it is not attributed, you will be referred to the School of Engineering under the University Discipline procedures.

Communication and attendance

Lectures. You are expected to attend the two lectures per week. It is intended that the lecture slides will be available on the LMS in advance of the lecture. However, these advance versions are **drafts**. The lecturer may edit the slides and notes after they are delivered. Lectures will be recorded using the university's lecture-capture facility; these recordings will be available to students via the LMS. You should regard these, however, as a safety net: attending lectures in person may be a better educational experience. Sometimes, due to technical failures, recordings are incomplete or unavailable.

You may not make recordings in lectures or tutorials without the written permission of the staff member or of any person who is identifiable in the recording.

Workshops. Each week, *starting on Monday 9th March*, you will be expected to attend one of the two-hour workshops. The workshop will contain about one hour of theory work (predominately questions from the textbook), and one hour of coding in C. It is critical that you attempt the set exercises **before** arriving at the workshop. A tutor will be on hand to assist you with your work towards an answer, but in general will not hand out answers. Note further that this is not a "rote learning" subject. You cannot simply memorise algorithms and expect to achieve in the tests and exam, which will examine your ability to absorb, synthesise, and use algorithmic techniques.

Please note that we will be also offering some online workshops from week 2 onwards. We will post the details soon on the LMS.

Seeking assistance. The LMS should be your first resource when seeking help. Check the "Announcements" Section in particular; perhaps it already has the answer to your problem. If the problem is not of a confidential nature, please post a question to "General Discussion" forum; other students may have the same question as you and would benefit from the answer. If you have a personal or confidential concern, please contact the lecturer or coordinator directly. If you contact staff directly and we feel that your message is better sent to the forum, we may answer by asking you to post it to the forum. Most face-to-face enquiries should be handled during or immediately after the lectures or in the Office Hours. Other consultations will be held by appointment only.

Use of email Communications with staff should be from your University email address; include the subject code COMP20007 in the subject line to help us understand the context of your message. Mail from other email services (gmail, hotmail, etc.) may be discarded without being read, but if for any reason you email us from such a service you must include your full name, your student number, and the subject code.

The subject web page. All subject information is posted on the LMS for COMP20007; you can access the LMS at `http://www.lms.unimelb.edu.au`. You will be expected to visit the LMS at least once a week and take note of the information posted there. Students without off-campus Internet access should routinely make use of University computing facilities to view the subject web page. Answers to questions will often be posted on the LMS, and it includes discussion forums for subject topics.

Workload. It is expected that you will spend between 10 – 12 hours per week on this subject. As a *minimum* the following is expected.

- 2×1 hour of lectures
- 2×1 hours of lecture revision
- 2 hours of workshop preparation
- 2 hours of workshop
- 1 hour of finishing off workshop
- 30 hours of assignment work ($30/12 = 2.5$ hours per week)