COMP10002
Foundations of Algorithms

Semester Two, 2017

Now What?

Finishing up

More Computing?

Exam Overview

Choose Your Own Fun

More Computing?

Exam Overview

Choose Your Own Fun

More computing?

The most obvious next subject is comp20007 Design of Algorithms, offered in semester 1 each year.

It is the pathway through to further languages (Java, in swen20003), and technologies (databases, in info20003; data handling, in comp20008).

The pathway through comp20003 Algorithms and Data Structures (offered semester 2) is also an option, but has some overlap in content with comp10002.

Subject comp20005 Engineering Computation is now restricted against comp10002. It has substantial overlap, and you can't take it.

More computing?

COMP10001 → COMP10002 → COMP20007 → ...
or
ENGR10003 → COMP20005 → COMP20003 → ...

If you are heading off to a Maths, Physics, Bio, Geomatics, whatever, major, computing skills are still critical. Use your second- and third-year elective slots wisely.

Want to combine two disciplines? Consider adding the Diploma in Informatics, to fit more in to your BSc.

The Exam

The exam is based primarily around three themes:

- Writing functions to show your command of control structures in C: selection, iteration, and abstraction;
- Writing declarations and functions to show your command of data structures in C: arrays, structs, linked structures, and combinations of them;
- Demonstrating your understanding of the fundamentals of algorithm design and analysis, including asymptotic execution times under different models, and "good" versus "bad" algorithms.

C control structures

Basic litmus tests:

- Integer number representations, general idea of floating point representations;
- Simple arithmetic and operations (function to sum a sequence);
- Loops and selection on scalars (prime numbers and etc);
- Pointer arguments to functions;
- Recursion.

C data structures

Basic litmus tests:

- Arrays, including functions that process arrays;
- Hierarchical construction of types and declarations (using constants, structs, and arrays);
- Manipulation of structs in functions via pointers;
- Broad principles of recursive structures using pointers: lists and trees.

Fundamentals of algorithms

Basic litmus tests:

- Functional growth rates and principles of analysis;
- Dictionary data structures, including arrays, lists, and binary search trees, and their tradeoffs and costs;
- Binary and linear search;
- At least one $O(n \log n)$-time sorting algorithm;
- Problem solving techniques and their various application domains.

To score well

Add in:

- Details of number representations;
- Program correctness arguments, and formulation of loop invariants;
- Pattern matching and string search algorithms;
- Functions over recursive structures (lists and trees);
- Details of Quicksort, Mergesort, Heapsort, including of their analysis and relative merits;
- Details of all dictionary structures, including hashing;
- Priority queues, including implementation via a heap.

To score really well

There will be at least one challenge that will involve on-the-spot creativity or insight.

Maybe a problem we haven't discussed, or a related problem that can be solved by adapting something you have been shown for a different purpose.

Maybe an analysis of a new algorithm, or some aspect of something covered only in passing in the lectures.

Maybe to work backward from an analysis, to derive an algorithm that meets it, putting together your knowledge from different parts of the subject.

Sample exam

The sample exam shows these various expectations.

- Questions 1 and 2 are "litmus";
- Questions 3(a), 3(b), 3(c), 4(b), 5(a), 6(a), 6(b), 6(e) are "litmus";
- Questions 3(d), 4(a), 5(b), 6(c), 6(d), are "score well";
- Question 5(c) and 7 are "score very well".

Don't allow yourself to be frightened by "score very well" questions unless you are hoping to score very well!

Exam preparation

- Start well in advance (now!);
- Lay out a study schedule that balances your exam timetable against particular subject needs;
- Put a copy of your exam timetable in a place where others will see it too;
- Scale back your social activities and other commitments;

Exam preparation

- Do (some of) your study with other people committed to passing;
- Read the whole book again (or for the first time), and not just the lecture slides.
- Listen to the lectures again. (Might make more sense the second time round...)

Exam preparation

- Revisit as many of the chapter exercises as you can manage;
- Read the LMS site from top to bottom;
- Review the on-line workshop solutions;
- Study the sample solutions to the projects;
- Do as much programming as you can, writing programs that work.

Exam preparation

- Revisit as many of the chapter exercises as you can manage;
- Read the LMS site from top to bottom;
- Review the on-line workshop solutions;
- Study the sample solutions to the projects;
- Do as much programming as you can, writing programs that work.
- Eat well, sleep well, and make sure you get regular sunshine and exercise.

Exam technique

- Highlight the key words in each question. Look out for the words write a function (can write more than one);
- Watch out also for you may, you should, and you may not;
- Draft answers using the blank pages;
- Use parts of one question to help you answer another, especially any program fragments;
- If you cannot write the C code, write answers in English describing the approach you planned to use.

Exam technique

- Be brief and to the point in any written answers;
- Be neat. Use a black or blue pen, or a **dark** pencil. No red or green or purple! No liquid paper, or pencil rubbings dandruff;
- Ask yourself "what competency is being tested with this question?", then try and deliver;
- See if you can include a surprise at some stage – an extension, or insightful comment to make the marker smile, or even a new computing joke.

Exam technique

If you think you have found a mistake in the examination paper (it does happen), state what your assumptions going forward are, and what the alternative assumptions might have been.

Then continue on the basis of your assumptions.

Don't get agitated about the problem, and don't just freeze and do nothing. We will assess your response to the unexpected situation, even though that situation may not have been planned.

Assessment

Hurdles: No one is prevented from sitting the exam. So if you fall into a perilous state because of your assignment marks, you should make a considered decision whether to abandon, and focus on other subjects (3/3 attempted is better than 2/4); or whether to try and pull it out of the fire via the exam.

The higher your exam mark, the harder it is for us to fail you, regardless of your project and test marks. Decisions to fail students get made very carefully, based on litmus abilities.

Questions?

Yes, there will be a sample solution to the sample exam, and workshop solutions posted, watch the Exam FAQ page.

Yes, Assignment 2 marks will be finalized and made available before the exam, hopefully by the end of next week.

A Q&A revision session will be held 2:15pm Friday 10th, see the FAQ page.

What else?

Subject experience survey

If not already done, please take five minutes to complete the on-line form, linked from the LMS.

The written comments will get read (after the exam results are loaded), and the feedback that is provided will be acted on if it is possible to do so.

But for that to happen, we need you to provide thoughtful feedback. You can see a summary of and response to last year's survey on the LMS.

Goodbye to the Viewers At Home

Nothing in the rest of the lecture is relevant to the exam. It is "for fun", some algorithmic entertainment for the people who have been coming to lectures.

Good luck with all of your exams...

Goodbye to the Viewers At Home

Nothing in the rest of the lecture is relevant to the exam. It is "for fun", some algorithmic entertainment for the people who have been coming to lectures.

Good luck with all of your exams...
And remember, Algorithms are Fun!