Introduction to Algorithms and Data Structures

Lecture 30: Revision lecture

Mary Cryan

School of Informatics
University of Edinburgh

The structure of Inf2-IADS assessment

cwk1: "Search Engine for a Large Text Corpus" (worth 10%)

cwk2: "Memory recycling, and the knapsack problem" (worth 10%)

cwk3: "Heuristics for the Travelling Salesman Problem" (worth 10%)

Quizzes: Quizzes 1-5 altogether compose 10% (2% each)

Blogs: worth 10%

Exam! Worth 50%

The Exam

- ▶ "Open book" exam (so there will be little "bookwork")

- ▶ Focused on the theoretical side.

- ▶ Pseudocode:
    - ▶ May be given pseudocode and asked to interpret/analyse it
    - ▶ May be asked to write pseudocode, we won't be strict about syntax

- ▶ Python:
    - ▶ May be given sections of Python and asked to interpret/analyse
    - ▶ May be asked to write little bits of Python

- ▶ How hard/novel the questions?
    - ▶ Some worked examples (more straightforward)
    - ▶ Some "unseen" problem-solving (probably a bit easier than tutorial-sheet Qs ... unless variants of tutorial Qs)
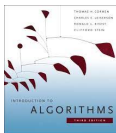
Exam structure

2 hours long.

Part A (50%)
- 5 Part A questions, each worth 10%
- Each of the part As is compulsory.
- Usually focused around one topic

Part B (50%)
- 3 Part B questions, choose your preferred 2
- Each of the Part Bs is worth 25%.
- Questions might be more broadly based than the Part As (maybe drawing on different parts of the course. But not always)

Questions will be reasonably "distributed" between semesters, topics ...

What should I know?

► All of the lecture slides/videos (including L24, L25, L26)

(except for a few flagged "unexaminable" parts from John's lectures).

► The suggested reading from [CLRS] for any lectures which have this listed.

(University of Edinburgh library delivers electronic access to this book).

► Sometime there are suggested readings from Goodrich, Goldwasser and Tamassia "Data Structures and Algorithms in Python" [GGT]

► All tutorial questions and their solutions.

(these are also part of the core content)

► You could be asked coursework-related questions (not "write a Python program to ..." but questions about concepts, algorithms, definitions).

semester 1 details

Always "lecture and video"

> L1 Overview of course content
> [CLRS] Chapter 1
>
> L2 Inefficient vs. efficient algorithms
> [CLRS] 2.1, 2.3 and 31.6 (2nd half)
>
> L3 Asymptotics: o and ω
>
> L4 More asymptotics: Big O, Omega and Theta
> L3+L4 reading: [CLRS] Chapter 3 OR [GGT] Sections 3.3, 3.4
>
> L5 Asymptotics for Insertsort and Mergesort
> [CLRS] 2.2

semester 1 cont'd.

L6 Representation of program data in memory
[CLRS] Chap 10, especially 10.2 and 10.3

L7 Classic datatypes: lists, stacks, queues
[CLRS] Chap 10, also 17.4.

L8 Sets, dictionaries and hashing
[CLRS] Chap 11 (the only **Theorem/proof** needed is Thm 11.1)

L9 Balanced trees
[CLRS] 12.1-12.3, 13.1-13.3

L10 Divide-conquer-combine and the Master Theorem
[CLRS] Chap 4: Intro and 4.3–4.5

semester 1 cont'd.

L11 The Heap data structure

L12 BuildHeap and HeapSort
   L11+L12 reading: [CLRS] Chap 6: 6.1-6.4

L13 Quicksort
   [CLRS] Chap 7: 7.1, 7.2 and 7.4

L14 Graphs 1: graphs, Breadth-first search, Depth-first search

L15 Graphs II: DFS and Topological Sort
   L14+L15 reading: [CLRS] Chap 22: 22.1-22.4

semester 2

▶ Semester 2 lectures are more "mixed" in topics than semester 1.

▶ We don't always have book chapters for the s2 lectures (or sometimes the book chapters have different emphasis).

▶ Make sure you know the Lecture slides, tutorials and solutions ... and watch the videos.

▶ Problems understanding notation?

   Maybe start a Piazza thread to discuss this, and get advice from lecturers.

The Quizzes are a good way to "revise" your knowledge, get you thinking.

semester 2 details

L16 Dijkstra's Algorithm

L17 Introduction to Dynamic Programming

L18 Seam-Carving and Edit Distance by Dynamic Programming
   L17+L18 reading: [CLRS] Chap 15: "Intro", 15.3, 15.4 (Longest
   Common Subsequence is related to edit dist)..

L19 All-pairs shortest paths by Dynamic programming
   [CLRS] Sections 25.1 and 25.2

L20 Probabilistic Finite State Machines and the Viterbi Algorithm

semester 2 cont'd.

L21 Context-free Languages and Grammars

Reading: Section 2.1 of "Introduction to the Theory of Computation (3rd ed)" by Sipser (Online access via UoE library).

L22 Parsing for Context-free Languages: The CYK Algorithm

Reading: Sections 13.1, 13.2 of "Speech and Language Processing, 3rd ed." by Jurafsky and Martin: https://web.stanford.edu/~jurafsky/slp3

L23 LL(1) Predictive Parsing

"Part 6" of video not examinable.

Reading: Sections 3.1, 3.2 of "Modern Compiler Implementation in C", by Appel and Ginsburg (Online access via UoE library).

semester 2 cont'd.

Other sources
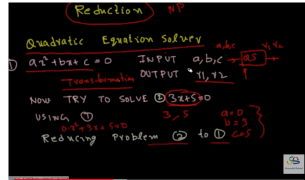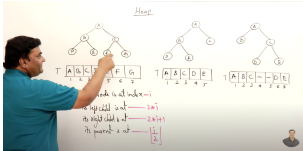
Some videos online give very nice "hand-wavy" chats about technical subjects
... but for Inf2-IADS we want technical details, rigorous arguments, proofs.

Other videos are of a similar level of technicality to ours (or do worked examples).
That is great, just make sure that you understand/know our own notation.

(so use these sources as motivation, then back to our slides/videos to learn the
details)

Practice

Quizzes:
Good way to test your coverage/understanding of the material.

Tutorials:
Probably closest (though a bit harder) to Exam questions.
And also core content!

CLRS problems and exercises

Inf2B (pre 18/19) and Inf2A (pre 18/19) exam papers

- ▶ The Algs+DataStructs half of Inf2B papers is our semester 1 material.

- ▶ Our L21-L23 topics (Grammars, Parsing, LL(1)) were part of the old Inf2A course, so there are (some) relevant questions in old Inf2A papers.

Our sample exam paper!
Now on the course webpage
2020/21 exam paper!
Available at Library Exam papers site