

Introduction to Algorithms and Data Structures

Exam preparation

Mary Cryan

School of Informatics
University of Edinburgh

The structure of Inf2-IADS assessment

cwk1: “Smart Mergesort and Perfect Hashing” (worth 20%)

cwk2: “Algorithms for Independent Set” (worth 20%)

Exam! Worth 60%

(note that Quizzes are for practice/self-testing only, not for credit)

The Exam

- ▶ “notes permitted” exam (but this means 1 sheet of A4 (both sides) of pre-prepared notes)
- ▶ 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

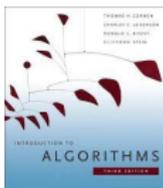
2 hours long.

- 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 (except for a few **flagged** “unexaminable” parts from John’s lectures).
- ▶ The suggested reading from [CLRS] *or* [AI] (Roughgarden) for any lectures which have this listed.



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

- ▶ 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

- 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 Greedy Algorithms

L18 Introduction to Dynamic Programming

L19 Seam-Carving by Dynamic Programming

L20 Edit Distance by Dynamic Programming

L18-L20 reading: [CLRS] Chap 15: "Intro", 15.3, 15.4 (Longest Common Subsequence is related to edit dist).

L21 Probabilistic Finite State Machines and the Viterbi Algorithm

semester 2 cont'd.

L22 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).

L23 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>

L24 LL(1) Predictive Parsing

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.

L25 P and NP

L26 Satisfiability and NP-completeness

L24+L25 reading: [CLRS] Chap 34: "Intro", 34.1-34.4, but ignore the "Circuit Satisfiability" in 34.3.

L27 Dealing with NP-completeness (approximation algorithms)

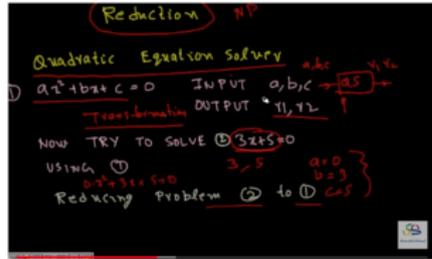
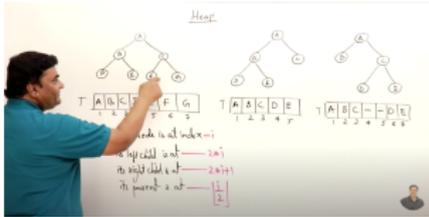
[CLRS] Chap 35: "Intro", 35.1, 35.4.

L28 Dealing with NP-completeness (more approximation, recursive backtracking)

L29 Introduction to Computability Theory

L30 Unsolvable Problems

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).

Please make sure that you understand/know our own notation/algorithms.

- ▶ Use other sources as **motivation**, as **high-level** understanding, but make sure you understand the fine details of **our** algorithms, our notation.
- ▶ In a worked example question of a finely-detailed algorithm, we will sometimes give the pseudocode in the exam question, but **you need to have done practice** in advance.

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

Previous exam papers

- ▶ Available at Library Exam papers site (and on our opencourse page)
- ▶ Exams from 2020/21, 2021/22, 2022/23, 2024/25 were based on the same syllabus as ours
- ▶ The May 2023 has some differences, especially on the semester 2 content. Do not stress if you see something unfamiliar in that paper.
- ▶ From May 2025, we are asked (by ITO) not to publish solutions.

Stages of progress

- ▶ Understanding the slides
- ▶ Confident in doing worked examples (given pseudocode)
- ▶ Confident in writing short descriptions of Algorithms/Data Structures
 - ▶ Really understanding the details of the notation and being able to remember it.
- ▶ Comfortable with the definitions of big-O, big- Ω , big- Θ and being able to employ these to analyse the running-times of our Algorithms.
- ▶ Understanding the steps in our Mathematical derivations, and being able to re-produce many of these.
- ▶ Understanding the proofs of correctness that we cover in class.
- ▶ Being able to remember and reproduce the main steps of our proofs.

Sheet of A4

This is the “notes permitted” rubric option:

Students can take into the exam just one A4 page (two sides) of pre-prepared notes (with a proportionate increase in the number of sheets of notes taken in for students with learning profiles that mandate the provision of larger format exam papers).

- ▶ You decide which notes you want to write on your own A4 sheet. This depends on what will help **you** most. Think about this, and make a first attempt soon.
- ▶ Magnifying devices to enlarge the contents of the sheets for viewing are not permitted.
- ▶ No further notes, printed matter or books are allowed.