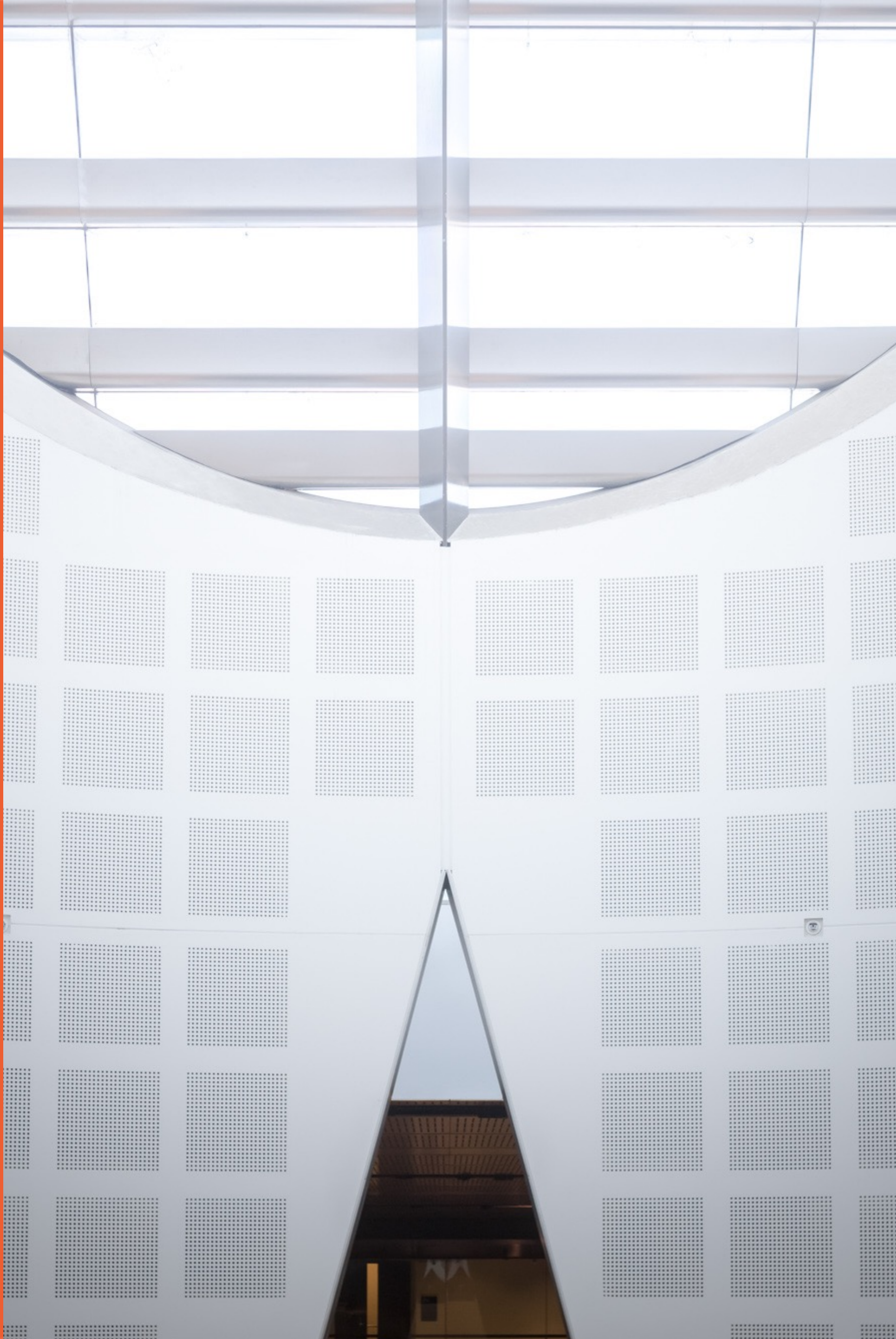
# COMP2123

## Week 13: Recap and Exam Review

A/Prof Julian Mestre

School of Computer Science

THE UNIVERSITY OF SYDNEY



# Quick announcements

Fill out online Unit of Study Survey

- https://student-surveys.sydney.edu.au/students/
- Use the free text to help us make this better for next years students. “Pay it forward”

Examples of changes based on past year’s feedback:

- wrote the guide on how to approach algorithmic problems
- designed programming exercises

# Week 13 Quiz

Quiz 10 will be about the final.

It's available until May 28th

# Looking back

We covered a lot of ground!

Lecture Slides

| | |
|---|---|
| Week 1 - Welcome slide pack | PDF |
| Week 1 - analysis | PDF |
| Week 1 - introduction | PDF |
| Week 2 - lists | PDF |
| Week 3 - trees | PDF |
| Week 4 - binary search trees | PDF |
| Week 5 - priority queues | PDF |
| Week 6 - hashing | PDF |
| Week 7 - graph traversals | PDF |
| Week 8 - shortest paths and minimum spanning trees | PDF |
| Week 9 - greedy | PDF |
| Week 10 - divide and conquer i | PDF |
| Week 11 - divide and conquer ii | PDF |
| Week 12 - algorithms_in_practice | PDF |
| Week 12 - empirical analysis | IPYNB |
| Week 12 - mock coding interview | IPYNB |

# Core concept 1: Abstraction layers

Abstract Data Type

↓

Data Structure

↓

Computer code

Problem definition

↓

Algorithm

↓

Computer code

# Core concept 2: Algorithm analysis

A principled framework for evaluating algorithms:

- measuring performance of resource use
- proving correctness

These should inform your design and implementation choices

# Learning outcomes

1. Proficiency in organising, presenting and discussing professional ideas [...]
2. Using mathematical methods to evaluate the performance of an algorithm.
3. Using notation of big-O to represent asymptotic growth of cost functions.
4. Understanding of commonly used data structures, including lists, stacks, queues, priority queues, search trees, hash tables, and graphs. This covers the way information is represented in each structure, algorithms for manipulating the structure, and analysis of asymptotic complexity of the operations.
5. Understanding of basic algorithms related to data structures, such as algorithms for sorting, tree traversals, and graph traversals.
6. Ability to write code that recursively performs an operation on a data structure.
7. Experience designing an algorithmic solution to a problem, coding it, and analysing its complexity.
8. Ability to apply basic algorithmic techniques (e.g. divide-and-conquer, greedy) to given design tasks.

# Beyond this unit of study

SCS offers many algorithmic units:
- COMP2022 Models of Computation (S2)
- COMP3027 Algorithm Design (S1)
- COMP3530 Discrete Optimisation (S2)
- COMP4445 Computational Geometry (S1)

New units coming soon:
- COMP4530 Discrete Optimisation (S2)
- COMP4270 Randomised Algorithms (S2)

Sydney Algorithms and Computation Theory group:
- weekly reading group on Algorithm research
- do a research project with us
- we are always looking for bright honours students

# What is examinable?

Everything from the lectures, the referenced sections of the textbooks, the tutorials, the quizzes, the assignments. Exceptions to this rule:

- when explicitly labeled as non-examinable.
- probabilistic analysis of randomized algorithms

In general, if it happened during this unit, you are expected to know about it!

Focus on the things we put most emphasis on, as seen in tutorials and assignments

# Final Exam Structure

2.5 hours writing plus 10 minutes reading

5 questions worth in total 100 points

Worth 60% of overall COMP2123 grade

Final exam has a 40% barrier

Restricted Open Book:

- You can bring handwritten or printed material (no limit)
- No electronic devices or any kind are allowed

# Do's and Don'ts for CC students

Exam is in person.

Check you exam timetable for details on the venue!

Restricted open book exam:

- Can refer to slides, tutorial solutions, assignment solutions, books used in the unit
- Making a **2-page summary is highly recommended**
- **Never** copy text verbatim from anywhere, including the slides (this is grounds for academic dishonesty case). If you refer to anything from the permitted material, write in your own words

Start your submission with your student ID

- Don't include your name

# Do’s and Don’ts for RE students

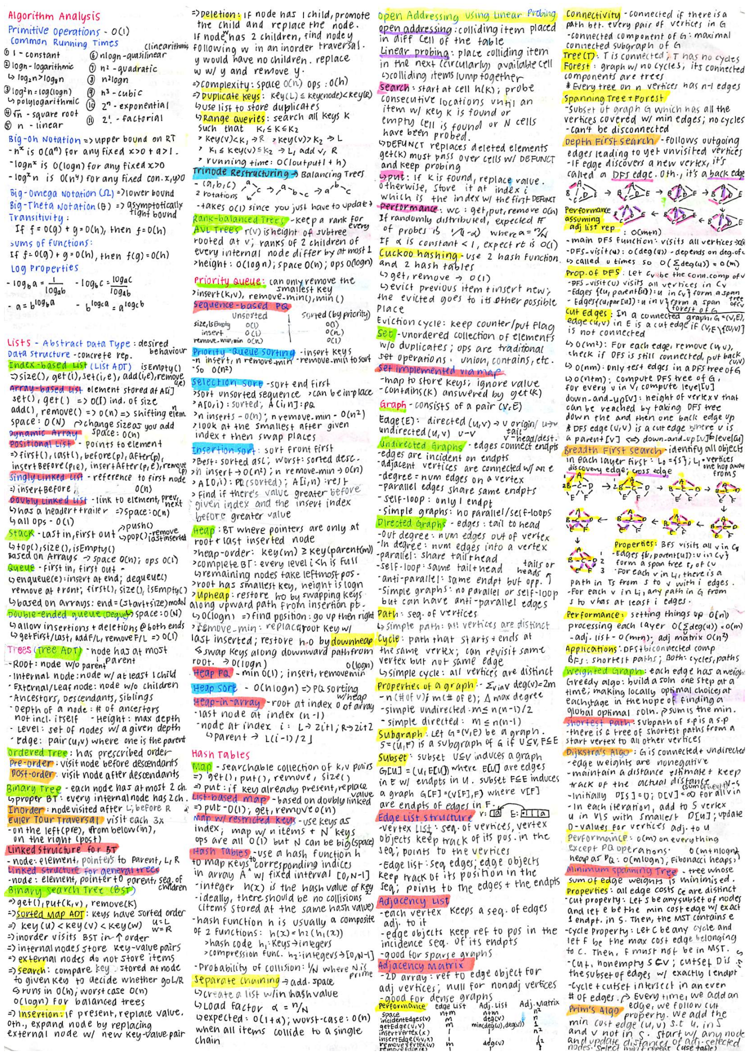
Exam is online (Canvas quiz) and supervised using ProctorU (Live+).

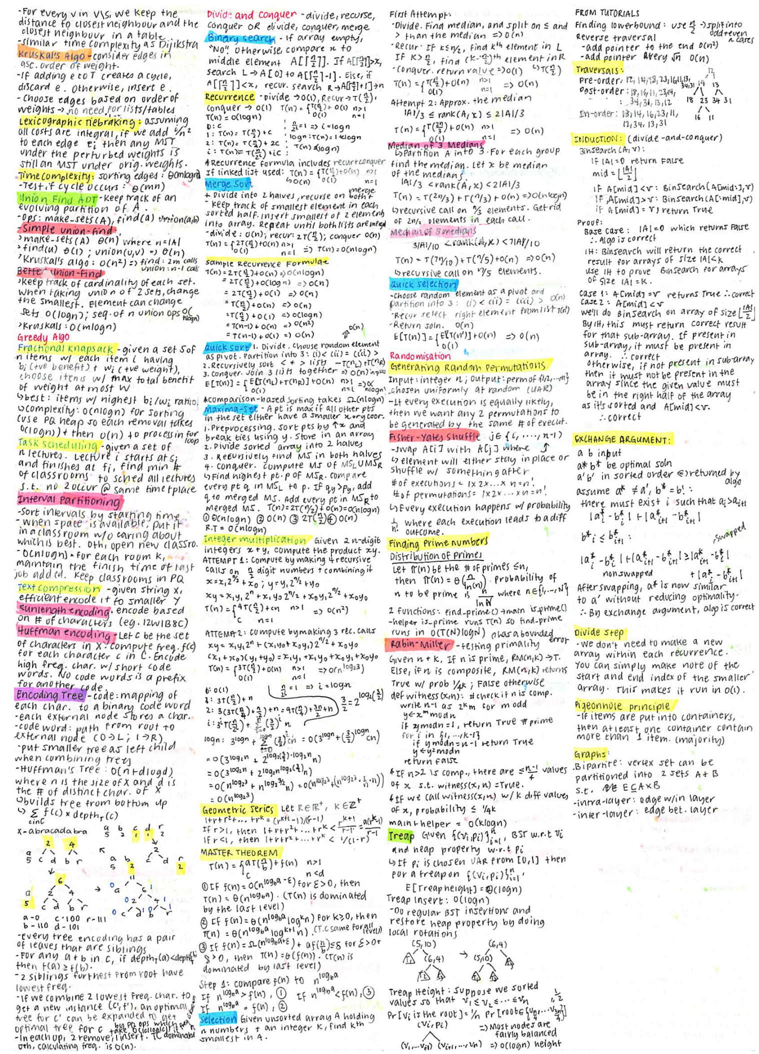
Check your internet connection and webcam.

Restricted open book exam:

- Can refer to slides, tutorial solutions, assignment solutions, books used in the unit
- Making a **2-page summary is highly recommended**
- **Never** copy text verbatim from anywhere, including the slides (this is grounds for potential academic dishonesty case). If you refer to anything from the permitted material, write in your own words

# Neatest summary page I came across





# Problem 1

20 points

Analysis of given algorithms

Easy problem. Make sure you nail it!

# Problem 2

20 points

Analysis of given algorithms

Easy problem. Make sure you nail it!

# Problem 3

20 points

Design or modify an ADT/algorithm that solves a problem

Medium difficulty problem.

Remember to:

- Describe your approach
- Prove correctness
- Analyze complexity (if there's a space requirement, don't forget to analyze this as well)

# Problem 4

20 points

Design or modify an ADT/algorithm that solves a problem

Medium difficulty problem.

Remember to:
- Describe your algorithm
- Prove correctness
- Analyze complexity

# Problem 5

20 points

Design or modify an ADT/algorithm that solves a problem

Hard problem.

Remember to:

- Describe your algorithm
- Prove correctness
- Analyze complexity

# Problem 3, 4, and 5

Check if you're supposed to use a specific technique:

- "design a greedy algorithm"
- "design a divide and conquer algorithm"

(Using a different technique will cost you a significant number of marks, but may still be better than a poorly explained incorrect attempt)

Let the running time requirement guide you:

- If we ask O(1) time, this limits your options considerably
- If we ask O(n) time, you can't sort the input

# Exam technique

Read all questions to see which ones you can answer quickly

Plan how you will allocate time (wisely)

Start with easy problems and move to harder ones

Write clearly and efficiently

- Start with outline/bullet points, then expand if you have time
- No need for fancy style or overly formal

# Pragmatic Advice (for CC students)

- It's a good idea to check the exam venue ahead of time
- Plan to arrive ahead of time (don't rely on public transport running smoothly on the day of the exam)
- Bring water, spare pens, and ID
- Have clothing in layers
- Start by writing your student ID. Do not write your name on the exam (marking is anonymous)
- Breathe and relax
- Follow the instruction of the invigilator

More info:

https://www.sydney.edu.au/students/exams/in-person.html

# Pragmatic Advice (for RE students)

- Be alone in your room
- Let housemates know when your exam is to avoid distractions
- Bring water
- Have clothing in layers
- Breathe and relax
- Make sure your webcam is working
- Follow the instructions in Canvas

More info:

https://www.sydney.edu.au/students/exams/online.html

# Quick announcements

Fill out online Unit of Study Survey

- https://student-surveys.sydney.edu.au/students/
- Use the free text to help us make this better for next years students. “Pay it forward”

Examples of changes based on past year’s feedback:

- wrote the guide on how to approach algorithmic problems
- designed programming exercises