CS217 - Data Structures & Algorithm Analysis (DSAA)

Fall 2025

Lecture #0

Introduction

Prof Pietro S. Oliveto
Department of Computer Science and Engineering
Southern University of Science and Technology (SUSTech)

olivetop@sustech.edu.cn
https://faculty.sustech.edu.cn/olivetop

Office: Room 113

About Me

- Previously **Professor of Computer Science, Chair in Algorithms**, University of Sheffield, UK
- Moved to **SUSTech** in April 2023 (Room 113)
- Taught **Algorithms and Data Structures** in the UK for a few years, and last two years at SUSTech
- I also teach Theory of Computation (CS338) – Spring semester (selective course)
- Expertise in **time complexity of randomised search heuristics and bio-inspired computation** (evolutionary algorithms, genetic algorithms, artificial immune systems etc.)
- I am building a **Theory of AI lab** at SUSTech (Room 348B)

Webpages:

- https://faculty.sustech.edu.cn/olivetop
- https://peteroliveto.github.io

What this is about

- **Algorithms** for solving computational problems
- **Understanding** how algorithms work
- How **efficient** is my algorithm?
- How to **design** efficient algorithms?
  - Greedy algorithms
  - Divide and conquer
  - Dynamic programming
- **Data structures**, efficiency of operations, and how to use data structures efficiently.

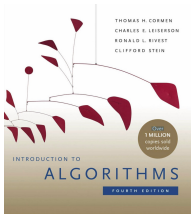
At the end of this course you'll be able to...

- Appreciate what constitutes an **efficient** and an **inefficient** solution to a computational problem
- **Analyse** the **efficiency** of an algorithm
- Evaluate and choose data structures that support **efficient algorithmic solutions**
- Identify and apply **design principles** such as greediness, divide and conquer and dynamic programming in the **design of efficient algorithms**
- Describe **efficient algorithms for fundamental computational problems**, along with their **computational complexity**.
- Difference to other programming courses (CS109 Intro to computer programming, CS205 C/C++, CS309 Object-oriented analysis and design): focus on analysing efficiency

Why care about Theory of Algorithms?

- Living in the **Age of Algorithms**
- Programming → Computer **Science**
- "*There is nothing so practical as a good theory.*"
- Can do so much more if you understand your algorithms.
- Design and analysis of algorithms is **at the heart of Computer Science**.
- Things every computer scientist is expected to know.
- **Employability** – Tencent, Huawei, Alibaba etc. require algorithms.
- Preparation for CS338 (Theory of Computation), CS208 (Algorithms Design and Analysis), and CSE5003/CS5018 (Advanced Algorithms)

Textbook



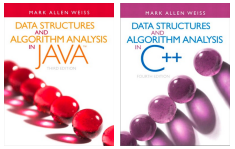
- **T. H. Cormen, C. E. Leiserson, R. L. Rivest and C. Stein (2009) *Introduction to Algorithms*, third/fourth edition (MIT Press)**
  - Available in library (English/Chinese),
  - Available online

What Cambridge tell their students:

"Students hoping to receive a computer science degree from Cambridge are expected to buy, make extensive use of, and keep as reference for their future career, one of the above fundamental textbooks: those not doing so will be severely disadvantaged.

The recommended choice is Cormen, Leiserson, Rivest and Stein which covers all topics listed and, in spite of its superb quality, is the cheapest: about 35 GBP new for over 1300 pages."

For Labs



- Mark A. Weiss (2014), *Data Structures and Algorithm Analysis in C++*, PEARSON
- Mark A. Weiss (2012), *Data Structures and Algorithm Analysis in C++*, PEARSON

Teaching

- **Lectures**
  - **Tuesday 16:20-18:10**, Lecture Hall 3, Room **306**
  - mixture of slides, notes, discussions, etc.
  - Blackboard: https://bb.sustech.edu.cn (check regularly)
  - WeChat group
- **Tutorial / Lab sessions**
  - **Tuesday 19:00-20:50**, Lecture Hall 3, Room **501**
  - **Marked exercises**: algorithm design & analysis + programming
  - **After submission solutions will be presented**
  - **Ask questions**, get **feedback**, get help for current exercise sheet

Assessment

- Marked Lab exercise sheets : **40%** of module mark
  - Help you **prepare for final exam**
  - Algorithm design and analysis + programming tests
  - Find out in labs whether your solution was correct.
  - Exercise deadline day will be on each sheet; Submit by **Tuesday at 4pm**
  - Solutions will be released after sheets were discussed in labs.
- Final exam: **60%** of module mark
  - Written test: 2 hours
- We'll provide a **revision class** and a **mock exam**.

Grading Policy

- Strict late assignment policy. Late Lab submissions within:
  - 6 hours: 25% mark deduction
  - 24 hours: 50% mark deduction
  - >24 hours: ALL marks lost
- Write your assignment on your own
- ZERO tolerance on plagiarism (Software will be used to detect plagiarism and cases reported to the university)
- You are not allowed to use Chat-gpt to do the theory or programming exercises

Tentative Schedule

No	Topic
1	Getting Started
2	Asymptotic Notation
3	Divide & Conquer
4	Heaps & In Place Sorting
5	Quicksort: Average-case Analysis
6	Randomisation & Lower Bounds
7	Linear Time Sorting
8	Elementary Data Structures

No	Topic
9	Binary Search Trees
10	Balanced Trees
11	Dynamic Programming
12	Greedy Algorithms
13	Elementary Graph Algorithms (I)
14	Elementary Graph Algs (II)
15	Minimum Spanning Trees & Shortest paths
16	Revision Class

Expectations on you

- In class:
  - Attend lectures and participate (ask questions, take notes)
  - Attend labs and participate (ask questions, you can also present solutions)
- Outside class:
  - **READ THE BOOK**
  - Recap slides/notes after each lecture
  - Do all exercise sheets
  - Use the discussion board on Blackboard or WeChat to ask questions
- Office hour (Office 113 - just drop in):
  - **Friday 14:00-16:00**
  - or by appointment: olivetop@sustech.edu.cn

What a student says

- Quote from Lyes Bouakaz, former Computer Science/ Student Ambassador for Learning & Teaching (University of Sheffield, UK)

"Your top tip for studying success:

Make sure that you understand everything you have been taught before the end of each week. It makes revision much easier and quicker, and the whole exam period is less stressful if you really are revising rather than trying to learn things you didn't understand before!"