

CS217 - Data Structures & Algorithm Analysis (DSAA)

Revision class

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>

► From Lecture 0: At the end of this module 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**.

►What you've learned

- **Foundations**
 - Asymptotic notation ($\Theta, O, \Omega, o, \omega$)
 - How to analyse the runtime of an algorithm
 - Loops, nested loops
 - Recursive algorithms (Master Theorem, substitution)
 - Why an algorithm is correct
 - How to prove correctness (loop invariants, induction, contradiction)
- **Data structures**
 - Stacks, queues, lists
 - Trees, AVL trees
 - Heaps and priority queues

► What you've learned (2)

- **Efficient algorithms** for solving problems
 - Sorting (in lots of different clever ways)
 - Rod cutting, Fibonacci numbers
 - Activity selection, coin changing, fractional Knapsack
 - **Graph problems**
 - Searching in graphs (BFS, DFS)
 - Topological sorting
 - Strongly connected components
 - Minimum spanning trees (in two ways: Kruskal and Prim)
 - Shortest paths

► What you've learned (3)

- **Design paradigms**
 - Divide-and-conquer (MergeSort, QuickSort)
 - Dynamic Programming
 - Greedy algorithms
- Glimpse into more **advanced topics**
 - Randomised algorithms (randomised QuickSort)
 - Complexity theory (lower bound $\Omega(n \log n)$ for all comparison sorts)
 - Complex correctness proofs (e.g. for strongly connected components)
- **How to use efficient data structures** to design efficient algorithms
 - e.g. HeapSort, Prim, Dijkstra

► Your feedback

- What was good?
- What can be improved?

►Your feedback (2): Teaching evaluation

- Deadline: 26 December (24:00) (anonymous to teachers)
- **important** reference basis for teachers to do a good job in the continuous improvement of teaching effectiveness
- Two ways:
 - Web end: <https://tis.sustech.edu.cn/> -Business Processing – Evaluation Tasks –2025 Fall Semester Student Evaluation Tasks
 - WeChat end: Enter the "Southern University of Science and Technology" WeChat enterprise account- teaching quality management platform: – "My Tasks – Pending Evaluation" section.
- 3 minutes to complete (all items mandatory)
- The Teaching Affairs Department will give course selection points to students who complete it, according to each student's objectivity in the evaluation questionnaire for each course



►CS-217 exam

- When?

**Monday, 29th December (16:30 – 18:30)
Teaching Building 3 – Room 102**

► What is relevant?

- Content from **all lectures** and **all tutorials** can come up.
- A good exam will cover all bases.
- Make sure you answer all the questions!
- **Tutorial exercises** are a good preparation.
- Mock exam and solutions on module web page.

►Exam Paper



Course Name: CS217 Data Structures and Algorithm Analysis

Dept.: Department of Computer Science and Engineering

Exam Duration: 2 hours

Exam Paper Setter: Pietro Oliveto

| Question No. | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|--------------|----|----|----|----|---|---|---|---|---|----|
| Score | 25 | 25 | 25 | 25 | | | | | | |

This exam paper contains 4 questions and the score is 100 in total. (Please hand in your exam paper, answer sheet, and your scrap paper to the proctor when the exam ends.)