Data Structures and Algorithms II

Module Description

Module code	CS232-F27	**Credits**	7 ECTS
Programme	Foundation programme	**Level**	Semester 3
Semester	Fall 2027	**Prerequisites**	CS103-S27 Data Structures and Algorithms I
Module leader	Dr. Ali Khudiyev	**Email**	ali.khudiyev@wosm.academy
Office hours	Monday–Friday, 10:00–17:00 (Baku time, UTC+4)		

1. Module Overview

Data Structures and Algorithms II continues from Data Structures and Algorithms I. Where the first module concentrated on data structures and sorting, this module concentrates on *algorithm design*: graph algorithms, and the general techniques of divide and conquer, greedy algorithms, dynamic programming and network flow.

The last part of the module looks at problems for which no efficient algorithm is known. You will learn to recognise NP-hard problems through reductions and to cope with them using approximation, randomisation and systematic search. This prepares you for Theory of Computation in Semester 4. Programming is done in C, as in Data Structures and Algorithms I.

2. Aims

- Teach the main paradigms of algorithm design and how to justify algorithms built with them.
- Give you the standard algorithms on graphs and networks and practice in modelling problems as graphs.
- Introduce amortised analysis and advanced data structures used inside efficient algorithms.
- Introduce intractability and practical strategies for hard problems.

3. Learning Outcomes

After completing this course, you will be able to:

LO1. Model problems as graphs and apply and analyse algorithms for traversal, topological sorting, connectivity, shortest paths and minimum spanning trees.

LO2. Design algorithms with divide and conquer, greedy choice and dynamic programming, and justify them with recurrences, exchange arguments and optimal substructure.

LO3. Formulate problems as network flow and bipartite matching problems and solve them.

LO4. Analyse algorithms and data structures, such as union–find, with aggregate, accounting and potential-method amortised analysis.

LO5. Prove problems NP-complete by polynomial-time reduction and choose a suitable strategy for them: approximation, randomisation, or exact search.

4. Syllabus

Unit	Topic
1	Graphs: representations, breadth-first and depth-first search
2	Applications of DFS: topological sorting, strongly connected components
3	Shortest paths I: Dijkstra's algorithm
4	Shortest paths II: Bellman–Ford; all-pairs shortest paths
5	Minimum spanning trees: Prim and Kruskal; union–find
6	Amortised analysis: aggregate, accounting and potential methods
7	Divide and conquer: Karatsuba, Strassen, closest pair of points
8	Greedy algorithms: exchange arguments, scheduling, Huffman coding
9	Dynamic programming I: principles; sequence problems
10	Dynamic programming II: knapsack, edit distance, shortest paths revisited
11	Network flow: Ford–Fulkerson, Edmonds–Karp, max-flow min-cut theorem
12	Applications of flow: bipartite matching and related problems
13	String algorithms: Knuth–Morris–Pratt, Rabin–Karp, suffix arrays
14	Reductions, P, NP and NP-completeness
15	Approximation algorithms
16	Randomised algorithms; backtracking and branch and bound
17	Course Review & Exam

5. Teaching and Learning Methods

The module is self-paced. All materials are provided on Moodle. Each unit has a lecture, exercises on designing and proving algorithms, and programming tasks in C.

Activity	Hours
Lectures	34
Labs / Tutorials	36
Independent study	130
Total	**200**

6. Assessment

Your knowledge will be assessed throughout the course with quizzes, programming assignments and the final exam. The exam checks whether you can choose a suitable design technique for an unfamiliar problem, develop an algorithm with it, and argue its correctness and running time.

Component	Weight	Due	LOs
Quizzes	20%	Throughout	LO1–LO5
Programming assignments	40%	Throughout	LO1–LO3
Final exam	40%	End of module	LO1–LO5
Total	**100%**		

7. Reading List

Essential

- T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein: *Introduction to Algorithms*, 4th ed., MIT Press, 2022.
- J. Kleinberg, É. Tardos: *Algorithm Design*, Addison-Wesley, 2005.

Recommended

- S. Dasgupta, C. H. Papadimitriou, U. V. Vazirani: *Algorithms*, McGraw-Hill, 2006.
- S. S. Skiena: *The Algorithm Design Manual*, 3rd ed., Springer, 2020.

8. Materials and Licence

Course materials are published under the Creative Commons Attribution-ShareAlike 4.0 licence unless stated otherwise. See wosm.academy/licence for details.

9. Policies

- **Attendance:** Attendance is not mandatory. The module is self-paced, so you can study the materials when it suits you.
- **Late submissions:** Work submitted after the deadline is penalised.
- **Academic integrity:** All submitted work must be your own. See the Code of Conduct.