I054	**Data Structures and Algorithms II**	L	S	E	ECTS
		3	0	2	7

Course objectives. The main course objective is to introduce advanced data structures and graph algorithms with their applications to the solving of a wide spectrum of different computational problems. At the end of the course, students are introduced to selected topics on algorithms in number theory and computational geometry. An efficient implementation of data structures and algorithms, by using some programming language, is also one of the most important objectives.

Course prerequisites. Data Structures and Algorithms I

Syllabus.

1. **Advanced Data Structures.** Red-Black trees. Augmenting Data Structures. Fenwick trees. B-Trees. Binomial Heaps. Fibonacci Heaps. Data structures for Disjoint Sets.
2. **Graph Theory Fundamentals.** Basic terms and definitions. Paths, walks and cycles. Connectivity. Trees and forests. Bipartite graphs. Euler tours. Hamiltonian cycles. Matchings. Planar graphs. Colouring. Flows and circulations.
3. **Elementary Graph Algorithms.** Representations of graphs. Breadth-first search. Depth-first search. Topological sort. Connected components. Minimum Spanning Trees. The algorithms of Prim and Kruskal. Single-Source Shortest Path problem. The Bellman-Ford algorithm. Single-source shortest paths in acyclic graphs. Dijkstra's algorithm.
4. **Advanced Graph Algorithms.** All-Pairs Shortest path problem. The Floyd-Warshall algorithm. Johnson's algorithm for sparse graphs. Maximum Flow problem. Flow networks. The Ford-Fulkerson method. Edmonds Karp's algorithm. Minimum Cut problem. Maximum bipartite matching.
5. **Selected Topics.** Linear Programming. Parallel algorithms. Polynomial and the Fast Fourier Transform. Number-Theoretic algorithms. String Matching algorithms. Computational Geometry algorithms.

EXPECTED LEARNING OUTCOMES

No.	**LEARNING OUTCOMES**
	To recognize and explain basic terms related to advanced dana structures and algorithms.
	To understand a theoretical analysis of correctness, time and space complexity of graph algorithms that are written in pseudo-code.
	To recognize and apply elementary graph algorithms to efficient solving of concrete computational problems.
1.	To efficiently implement algorithmic solutions in different programming languages.

COUPLING OF THE EXPECTED LEARNING OUTCOMES, TEACHING PROCESS ORGANIZATION AND THE EVALUATION OF THE TEACHING OUTCOMES

TEACHING PROCESS ORGANIZATION	ECTS	EXPECTED LEARNING OUTCOMES **	STUDENT ACTIVITY *	EVALUATION METHOD	SCORE	
					min	max
Lecture attendance	1	1-5	Class attendance, discussion, solving the problems individually and in a team	Lists with signatures, observing the activity during the lectures	0	10
Homework	1	1-4	Solving the problems individually	Grading	17	30
Repeated exams	2	1-4	Preparation for the written exam	Grading	17	30
Final exam	2	1-4	Revising	Oral exam	16	30
TOTAL	7				50	100

Teaching methods and student assessment. Lectures contain a deep and systematic overview of advanced data structures and algorithms. During exercises studenta are expected to solve given programming problems by using acquired knowledge. The correctness, time and space complexity of implemented algorithms are the most important elements. At the end of each practice session students individually solve short quizzes. During the semester, students solve homework assignments that contain programming problems. The assessment of theoretical knowledge is done by written examinations. If students achieve satisfactory results in homework and written exams, they are not obliged to take final written and oral exams. A seminar or final project can contribute to the final grade.

Can the course be taught in English: Yes

Basic literature:

1. T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, Introduction to Algoritms, 3Ed, MIT Press, 2009.

Recommended literature:

1. M. T. Goodrich, R. Tamassia, D. M. Mount, Data Structures and Algorithms in C++, Wiley, 2010.
2. A. Drozdek, Data Structures and Algorithms in C++, Cengage Learning, 2012.
3. R. Sedgewick, K. Wayne, Algorithms, Addison-Wesley Professional, 2011.
4. M. J. Atallah, Algorithms and Theory of Computation Handbook, CRC Press, 1998.
5. P. Brass, Advanced Data Structures, Cambridge University Press, 2008.
6. R. Diestel, Graph Theory, 2nd edition, Springer, 2000.
7. R. Sedgewick, Algorithms in C++, Parts 1-4 Fundamentals, Data Structure, Sorting, Searching, Third Edition, Addison-Wesley Professional, 1998.
8. R. Sedgewick, Algorithms in C++, Part 5: Graph Algorithms, Third Edition, Addison-Wesley Professional, 2002.
9. J. Šribar, B. Motik: Demistificirani C++, 4. dopunjeno izdanje usklađeno sa standardom C++11/C++14, Element, Zagreb, 2014.