COURSE UNIT (MODULE) DESCRIPTION

Course unit (module) title	Code
Algorithm Design and Analysis	

Lecturer(s)	Department(s) where the course unit (module) is delivered
Coordinator: Martynas Sabaliauskas **Other(s):** -	Vilnius University Faculty of Mathematics and Informatics

Study cycle	Type of the course unit (module)
First	Compulsory

Mode of delivery	Period when the course unit (module) is delivered	Language(s) of instruction
Face-to-face or remote	3rd semester	Lithuanian / English

Requirements for students	
Prerequisites: none Knowledge of foundations of Discrete Mathematics and Calculus (or course in Higher Mathematics), ability to write program codes.	**Additional requirements (if any): none**

Course (module) volume in credits	Total student’s workload	Contact hours	Individual work
5 ECTS	125	48	77

Purpose of the course unit (module): programme competences to be developed		
Purpose of the course unit – to develop student’s ability to design efficient algorithms for real world discrete problems as well as to estimate the complexity of algorithms and problems.		
Learning outcomes of the course unit (module)	**Teaching and learning methods**	**Assessment methods**
Analyze algorithms and estimate their complexity.	Lecture. Case study. Problem-oriented teaching. Individual reading. Problem solving.	Homeworks (written). Programming project (program code, experiments, typed report and presentation). Exam (written).
Analyze complexity of real world problems.		
Design efficient algorithms in practise.		
Distinguish between tractable and intractable problems.		

Content: breakdown of the topics	Contact hours							Self-study work: time and assignments	
	Lectures	Tutorials	Seminars	Exercises	Laboratory work	Internship/work placement	**Contact hours**	**Self-study hours**	**Assignments**
1. Introduction to algorithm analysis. Algorithms and their properties. Sorting Algorithms. Measuring complexity of algorithms and problems. Counting techniques useful in algorithm analysis. Growth of functions. Upper and lower complexity bounds for sorting.	4			8			**12**	**12**	Individual reading, problem solving, case study.
2. Combinatorial objects, their properties and their presentation by different data structures. Integers, sets, sequences, trees and graphs.	2			4			**6**	**6**	
3. Basic techniques for the design and analysis of efficient algorithms: divide-and-conquer, dynamic programming, backtracking, branch-and-bound,	4			8			**12**	**12**	

greedy and heuristic algorithms. Analysis of Matrix Multiplication, Knapsack and Travelling Salesman problems.									
4. Main graph algorithms and their analysis. Depth-first search and Breadth-first search. Minimum Spanning Tree problem. Algorithms of Prim and Kruskal. Shortest Path problem. Floyd–Warshall algorithm. Euler and Hamilton graphs.	4			8			**12**	**12**	
5. Complexity classes P and NP. Reduction techniques and NP-complete problems. Problems CIRCUIT-SAT, SAT, CLIQUE, VERTEX COVER, HAMILTON and TSP. Approximation algorithms. Approximation schemes FPTAS and PTAS.	2			4			**6**	**6**	
6. Laboratory works								**16**	Project: algorithm implementation, analysis, testing and written presentation.
Prepare for exam								**13**	
Total	**16**			**32**			**48**	**77**	

Assessment strategy	**Weight,%**	**Deadline**	**Assessment criteria**
Four laboratory works: to implement a specific algorithm, make experiments, evaluate the complexity theoretically and practically, and prepare a presentation and program demonstration.	40	During the semester	Each of the four laboratory works is evaluated separately, with a maximum score of 1 point. It is possible to earn up to 4 points for completing all four laboratory assignments. Each programming task is defended by demonstrating it and answering the questions posed.
Exam	60	During the exam	Before the exam, the student must know all course topics and the required literature sources. The exam is graded up to 6 points, based on written answers to 6 questions. The final grade is calculated by summing the scores from the programming assignments and the exam.

External exam
External students have no opportunity to pass the exam.

Author	**Year of publication**	**Title**	**Issue of a periodical or volume of a publication**	**Publishing place and house or web link**
Compulsory reading				
T.H. Cormen, C.E. Leiserson, R.L. Rivest and C. Stein	2022	Introduction to Algorithms, Fourth Edition 4th ed.		MIT Press, Cambridge, Massachusetts
V. Dičiūnas	2005	Algoritmų analizės pagrindai		**http://www.mif.vu.lt/katedros/cs/Asmen/algoritmu_analize.pdf**
Optional reading				
Sandeep Sen and Amit Kumar	2019	Design and Analysis of Algorithms: A Contemporary Perspective		Cambridge University Press
E.M. Reingold, J. Nievergelt and N. Deo	1977	Combinatorial Algorithms: Theory and Practice		Prentice-Hall
R. Čiegis	2007	Duomenų struktūros, algoritmai ir jų analizė		Vilnius: Technika