University of Gdańsk

Subject card

Subject name and code	Algorithms and Data Structures, PG_00155275						
Field of study	Mathematical Modeling and Data Analysis						
Date of commencement of studies	October 2025		Academic year of realisation of subject	2026/2027			
Education level	Bachelor's studies		Subject group	Obligatory subject group in the field of study			
Mode of study	full-time studies		Mode of delivery	at the university			
Year of study	2		Language of instruction	Polish			
Semester of study	3		ECTS credits	4.0			
Learning profile	academic		Assessment form	credit			
Conducting unit							
Name and surname of lecturer (lecturers)	Subject supervisor		dr hab. Błażej Szepietowski				
	Teachers						
Lesson types	Lesson type	Lecture	Tutorial	Laboratory	Project	Seminar	SUM
	Number of study hours	30.0	0.0	30.0	0.0	0.0	60
	E-learning hours included: 0.0						
Learning activity and number of study hours	Learning activity	Participation in didactic classes included in study plan		Participation in consultation hours		Self-study	SUM
	Number of study hours	60		5.0		35.0	100
Subject objectives	The aim of the course is to familiarize the student with basic algorithms and data structures, methods of proving the correctness and determining the time complexity of algorithms, and constructing effective algorithms.						

Learning outcomes

Course outcome	Subject outcome	Method of verification
[MMiADL3_W09] knows and understands the basics of computational and programming techniques supporting mathematician's work and understands their limitations	knows the basic methods for designing efficient algorithms and the fundamental data structures, understands the limitations of applying particular data structures and algorithms	[SW4] test/exam - oral or written
[MMiADL3_U11] knows how to arrange and analyse an algorithm in accordance with the specification and save it in the selected programming language	is able to develop a correct algorithm for a given computational problem, analyze its time complexity and implement it in a chosen programming language.	[SU5] implementation of a problem task
[MMiADL3_U13] knows how to use computer programmes in the field of data analysis	is able to apply implemented algorithms and data structures to analyze information and solve computational problems, as well as use programming tools to facilitate this analysis.	[SU5] implementation of a problem task
[MMiADL3_K03] is ready to work in a team; understands the need for systematic work on all projects that have a long-term character	is ready to collaborate effectively within a team during the implementation of a project task	[SK8] observation of student's independent or team work
[MMiADL3_U12] can compile, run and test a self-written computer programme	uses the selected programming environment in the process of implementing, debugging and testing applications	[SU8] observation of student's independent or team work
[MMiADL3_U10] is able to recognise problems, including practical issues, that can be solved algorithmically; can make a specification of such a problem	is able to analyze a problem, identify its computational aspects that can be solved algorithmically,	[SU5] implementation of a problem task
[MMiADL3_U09] is able to use the learned software package or the learned programming language to solve selected problems from the known fields, in particular from mathematical analysis, linear algebra and statistics	can use the acquired programming language to solve selected computational problems by selecting appropriate data structures	[SU5] implementation of a problem task

Subject contents

1. Basic data structures: lists, stacks, queues, trees and their implementations.
2. Analysis of algorithms, their semantic correctness, pessimistic and average time complexity.
3. Sorting by comparison. Algorithms with quadratic complexity (insertion-sort), with linear-logarithmic complexity (merge-sort, heap-sort), with average linear-logarithmic complexity (quick-sort). Lower bound on the pessimistic time complexity.
4. Binary heaps and applications. Priority queue.
5. Sorting in linear time.
6. Data structures for dictionary operations (insert, delete, search): hash tables, binary search trees, red-black trees, B-trees.
7. Methods of constructing effective algorithms: divide and conquer, dynamic programming, greedy strategy.
8. English nomenclature of the course

Prerequisites and co-requisites

The student knows methods of calculating and estimating sums, basic definitions and notations related to sets, relations, graphs and trees, basic combinatorial concepts: permutation, combination, etc., definitions and theorems of basic probability theory.

Assessment methods and criteria

Subject passing criteria	Passing threshold	Percentage of the final grade
projects	51.0%	50.0%
test	51.0%	50.0%
Observation of the student's attitude	51.0%	0.0%

Recommended reading

Basic literature	T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, *Wprowadzenie do algorytmów*, WNT 2007,
Supplementary literature	L. Banachowski, K. Diks, W. Rytter, *Algorytmy i struktury danych*, WNT 2004, N. Wirth, *Algorytmy+ struktury danych= programy*, WNT 2006,
eResources addresses	

Example issues/ example questions/ tasks being completed	1. Sort the given sequence of functions by their asymptotic growth rate. 2. Illustrate the subsequent steps of the BUILD-MAX-HEAP procedure that transform the given array into a max-heap. 3. Demonstrate the execution of the QUICKSORT algorithm for the given input sequence. 4. Draw the BST tree constructed by inserting nodes with the following keys into an initially empty tree. 5. Construct the Huffman code tree for a given alphabet and compute its cost. 6. Write a program that checks the correctness of a parenthetical expression provided by the user. 7. Prove the correctness of the Insertion Sort algorithm using a suitable loop invariant. 8. Write pseudocode for a function that recursively calculates the height of a node in a binary tree.
Work placement	Not applicable

Document generated electronically. Does not require a seal or signature.