University of Gdańsk

Subject card

Subject name and code	Algorithms and Data Structures, PG_00204159						
Field of study	Informatics						
Date of commencement of studies	October 2026		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	1		Language of instruction		Polish		
Semester of study	2		ECTS credits		5.0		
Learning profile	practical		Assessment form		exam		
Conducting unit	Institute of Informatics -> Faculty of Mathematics, Physics and Informatics -> Rector						
Name and surname of lecturer (lecturers)	Subject supervisor		dr Paweł Pączkowski				
	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		0.0		65.0	125
Subject objectives	Familiarizing students with classical algorithms and data structures used to effectively solve typical programming tasks, methods of implementing the studied algorithms, analysis of the time complexity of these algorithms and justification of their correctness						

Learning outcomes

Course outcome	Subject outcome	Method of verification
[INFPL3_U08] is able to assess the usefulness of various paradigms and programming tools for solving various types of problems	can program the studied algorithms based on their description in the form of pseudocode	[SU5] implementation of a problem task
[INFPL3_K02] is ready to recognize the importance of knowledge in solving cognitive problems and practical and seeking opinions experts in case of difficulties with independent problem solving	can formulate statements on algorithms and data structures and understands the necessity of further education	[SK5] implementation of a problem task
[INFPL3_W04] knows and understands advanced issues in programming, algorithms and computational complexity, programming languages and paradigms, as well as the complex relationships between these areas	knows classical data structures (letters, stacks, trees, tables with hashing) and operations on them knows selected effective sorting algorithms knows the facts about time complexity of sorting algorithms as well as algorithms for searching, inserting and deleting in selected data structures	[SW4] test/exam - oral or written
[INFPL3_U01] can apply mathematical knowledge to formulate, analyse and solve tasks related to computer science, design and analyze algorithms in terms of their correctness and computational complexity	can explain, using an example, the operation of selected classical algorithms can give definitions of selected commonly used data structures and illustrate them with examples (stacks, queues, heaps, trees, hash tables)	[SU4] test/exam - oral or written

Subject contents

- Introductory concepts: semantic correctness, pessimistic and expected time complexity, asymptotic time complexity notation.
- Sorting by comparison. Algorithms with quadratic complexity, linear-logarithmic complexity (heapsort), and expected linear-logarithmic complexity (quicksort). Lower bound on pessimistic and expected time complexity.
- Basic data structures: lists, stacks, queues, priority queues. Implementations using arrays and linked structures.
- Data structures for insertion, deletion and searching and time complexity of these operations: hash tables, binary search trees, balanced trees.
- Strategies for creating effective algorithms: divide and conquer, dynamic programming, greedy strategy - presented using examples.
- Amortized cost analysis.
- Familiarizing students with the terminology in English.

Prerequisites and co-requisites

Programming skills, the knowledge of mathematical apparatus at the level of Discrete Mathematics lecture

Assessment methods and criteria

Subject passing criteria	Passing threshold	Percentage of the final grade
exam	51.0%	50.0%
computer programs on laboratory	51.0%	35.0%
tests	51.0%	15.0%

Recommended reading

Basic literature	• T. H. Cormen, C. E. Leiserson , R. L. Rivest, C. Stein, Wprowadzenie do algorytmów, Wydawnistwo Naukowe PWN 2012. • L. Banachowski, K. Diks, W. Rytter, Algorytmy i struktury danych, WNT 2011.
Supplementary literature	no recommendations
eResources addresses	

Example issues/ example questions/ tasks being completed

- Provide the definition of a binary heap, illustrated by an example.
- Write and test a program that sorts an array of integers using the Heap-sort algorithm.

Work placement

Not applicable