University of Gdańsk

Subject card

Subject name and code	Algorithms and Data Structures p.1, PG_00143491						
Field of study	Informatics						
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 Subject group related to scientific research 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	7.0			
Learning profile	academic		Assessment form				
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		115.0	175
Subject objectives	To familiarize 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
[INFL3_U03] can design and analyze algorithms for their correctness and computational complexity using appropriate algorithmic techniques and data structures	can explain, using an example, the operation of selected classical algorithms can provide definitions of selected commonly used data structures and illustrate them with an example (stacks, queues, heaps, trees, hash tables) can program the studied algorithms using their description in the form of pseudocode	[SU4] test/exam - oral or written [SU5] implementation of a problem task
[INFL3_U02] can precisely formulate questions to deepen one's understanding of a given topic or find missing elements of reasoning	can formulate statements about algorithms and data structures and understands the need for further education	[SU5] implementation of a problem task
[INFL3_W03] has structured, theoretically grounded general knowledge in the field of algorithms and data structures, formal languages, automata theory and computational complexity, and artificial intelligence	knows classical data structures (lists, stacks, trees, hash tables) and operations on these structures knows selected effective sorting algorithms knows facts about the time complexity of sorting, searching, inserting and deleting algorithms	[SW4] test/exam - oral or written

Subject contents

- Introductory concepts: semantic correctness, pessimistic and expected time complexity, asymptotic 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.
- Linear time sorting
- The selection problem
- Basic data structures: lists, stacks, queues, priority queues. Implementations using arrays and link structures.
- Data structures for insertion, deletion and searching: hash tables, binary search trees, balanced trees (red-black trees)
- Amortized cost analysis
- Familiarizing students with the terminology in English

Prerequisites and co-requisites

Discrete Mathematics, Programming Languages

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

Assessment methods and criteria

Subject passing criteria	Passing threshold	Percentage of the final grade
programs and tests assessed in laboratories	51.0%	50.0%
written exam	51.0%	50.0%

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

Work placement

Not applicable

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