University of Gdańsk

Subject card

Subject name and code	Algorithms and Data Structures, PG_00143796						
Field of study	Informatics						
Date of commencement of studies	October 2024		Academic year of realisation of subject	2025/2026			
Education level	Bachelor's studies		Subject group	Obligatory subject group in the field of study			
Mode of study	part-time studies		Mode of delivery	at the university			
Year of study	2		Language of instruction	Polish polish			
Semester of study	3		ECTS credits	8.0			
Learning profile	academic		Assessment form				
Conducting unit							
Name and surname of lecturer (lecturers)	Subject supervisor		dr inż. Anna Nenca				
	Teachers						
Lesson types	Lesson type	Lecture	Tutorial	Laboratory	Project	Seminar	SUM
	Number of study hours	30.0	30.0	0.0	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	140.0	200	
Subject objectives	Introduce students to classical algorithms and data structures used to efficiently solve typical programming tasks, justify the correctness of the algorithms learned and conduct time complexity analysis of these algorithms.						

Learning outcomes

Course outcome	Subject outcome	Method of verification
[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	has a structured, theoretically supported general knowledge in the field of classical sorting algorithms and those implementing dictionary operations (insert, delete, search) and their time complexity knows examples of algorithms representing basic methods of creating efficient algorithms: divide and conquer, dynamic programming, greedy strategy has basic knowledge of analyzing the correctness and complexity of algorithms knows classical data structures: stacks, lists, queues, mounds, trees, arrays with hashing knows classical data structures (lists, stacks, trees, hash tables, balanced trees) and operations on them knows selected sorting algorithms knows facts about the time complexity of sorting algorithms, search,insert, delete	[SW4] test/exam - oral or written
[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 is able to give definitions of selected classical, commonly used data structures and illustrate them with an example is able to give examples of algorithms with different time complexity computational complexity and is able to evaluate the time complexity of a simple algorithm	[SU4] test/exam - oral or written
[INFL3_U02] can precisely formulate questions to deepen one's understanding of a given topic or find missing elements of reasoning	is able to formulate questions with precision, serving to deepen his own understanding of the of a given topic or to find missing elements of reasoning	[SU8] observation of student's independent or team work

Subject contents

- Introductory concepts: semantic correctness, pessimistic and expected time complexity, asymptotic complexity notation.
- Sorting by comparisons. Algorithms of quadratic complexity, linear-logarithmic complexity (heapsort), average linear-logarithmic complexity (quicksort).
- Lower bound theorems of pessimistic and expected time complexity for sorting algorithms.
- Sorting in linear time.
- Basic data structures: lists, stacks, queues, priority queues. Implementations using arrays and linked structures. Data structures for dictionary operations (insert, delete, search): hash tables, binary search trees,
- Methods for constructing efficient algorithms: divide-and-conquer method, dynamic programming (longest common subsequence), greedy algorithms (Huffman codes).
- Searching for a pattern in a text

Prerequisites and co-requisites

ability to program knowledge of the mathematical apparatus at the level of a discrete mathematics lecture

Assessment methods and criteria	Subject passing criteria	Passing threshold	Percentage of the final grade
	Activity during lectures and exercises	0.0%	0.0%
	tests	51.0%	50.0%
	implementations	100.0%	0.0%
	exam	51.0%	50.0%
Recommended reading	Basic literature	T. H. Cormen, C. E. Leiserson , R. L. Rivest, C. Stein -- Introduction to algorithms, Wydawnictwo Naukowe PWN 2012. L. Banachowski, K. Diks, W. Rytter -- Algorytmy i struktury danych, Wydawnictwo Naukowe PWN 2018	
	Supplementary literature	none	
	eResources addresses		
Example issues/ example questions/ tasks being completed	none		
Work placement	Not applicable		

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