University of Gdańsk

Subject card

Subject name and code	Algorithms and Data Structures, PG_00204257		
Field of study	Mathematics		
Date of commencement of studies	October 2026	Academic year of realisation of subject	2027/2028
Education level	Bachelor's studies	Subject group	Obligatory subject group in the field of study Optional subject group
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	Institute of Mathematics -> Faculty of Mathematics, Physics and Informatics -> Rector		
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	2.0	38.0	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, constructing effective algorithms.

Learning outcomes

Course outcome	Subject outcome	Method of verification
[MATL3_U09] is able to recognize problems, including practical issues, that can be solved algorithmically; is able to make specification for such a problem	The student is able to identify computational problems, particularly those related to data processing and information retrieval, formulate a precise problem specification, and propose appropriate algorithms and data structures to solve it.	[SU5] implementation of a problem task
[MATL3_U10] is able to create and analyze an algorithm consistent with the specification and write it in a selected programming language, as well as compile, run and test a computer program written by himself/herself	The student is able to design, implement and analyse the time complexity of algorithms that make use of classical data structures, and is capable of writing a solution in a chosen programming language, compiling it, running it, and testing it for correctness and efficiency.	[SU5] implementation of a problem task
[MATL3_W08] knows and understands at an advanced level the basics of computational techniques and programming that support the work of a mathematician and understands their limitations	The student knows and understands selected algorithms and data structures, methods of their design and analysis, and the limitations of computational methods, including lower bounds on complexity and conditions under which certain techniques can or cannot be applied.	[SW4] test/exam - oral or written

<table>
<tr><td>Subject contents</td><td>1. Selected data structures: lists, stacks, queues, trees and their implementations.
2. Analysis of algorithms, time complexity.
3. Sorting by comparison. 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</td></tr>
<tr><td>Prerequisites and co-requisites</td><td>Knowledge of the concepts of limit of a sequence, sum of a series, expected value of a discrete random variable, knowledge of the basics of programming.</td></tr>
<tr><td>Assessment methods and criteria</td><td><table>
<tr><th>Subject passing criteria</th><th>Passing threshold</th><th>Percentage of the final grade</th></tr>
<tr><td>test</td><td>51.0%</td><td>50.0%</td></tr>
<tr><td>projects</td><td>51.0%</td><td>50.0%</td></tr>
</table></td></tr>
<tr><td>Recommended reading</td><td><table>
<tr><td>Basic literature</td><td>1. T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, Introduction to algorithms, 2022.</td></tr>
<tr><td>Supplementary literature</td><td>1. L. Banachowski, K. Diks, W. Rytter, Algorytmy i struktury danych, PWN 2018</td></tr>
<tr><td>eResources addresses</td><td></td></tr>
</table></td></tr>
<tr><td>Example issues/ example questions/ tasks being completed</td><td>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.</td></tr>
<tr><td>Work placement</td><td>Not applicable</td></tr>
</table>

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