GDAŃSK UNIVERSITY OF TECHNOLOGY

Subject card

Subject name and code	Algorithms and Data Structures, PG_00042217		
Field of study	Mathematics		
Date of commencement of studies	October 2023	Academic year of realisation of subject	2024/2025
Education level	first-cycle studies	Subject group	Optional subject group 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	4	ECTS credits	4.0
Learning profile	general academic profile	Assessment form	assessment
Conducting unit	Divison of Nonlinear Analysis -> Institute of Applied Mathematics -> Faculty of Applied Physics and Mathematics		
Name and surname of lecturer (lecturers)	Subject supervisor	dr inż. Jakub Maksymiuk	
	Teachers	mgr inż. Tomasz Gzella dr inż. Jakub Maksymiuk	

Lesson types and methods of instruction

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	0.0	60

Subject objectives

Getting familiar with most important algorithms and data structures, and also with basics of analysis of algorithms, including computational complexity.

Learning outcomes

Course outcome	Subject outcome	Method of verification
K6_U10	The student is able to write a program that meets the project requirements and test it.	[SU4] Assessment of ability to use methods and tools [SU1] Assessment of task fulfilment
K6_W08	The student is able to describe the advantages, disadvantages and limitations of selected algorithms.	[SW3] Assessment of knowledge contained in written work and projects

Subject contents

1. Mathematical foundations of analysis of algorithms, including the big-O notation.

2. Design and analysis of algorithms.

3. The "divide and conquer" method and its sample applications.

4. Heapsort, quicksort, sorting in linear time, medians and position statistics.

5. Elementary data structures: lists, queues, heaps, binary search trees.

6. Dynamic programming, greedy algorithms, amortized cost analysis.

7. Selected graph algorithms: DFS and BFS, finding shortest paths.

<table>
<tr><td>Prerequisites and co-requisites</td><td colspan="3"></td></tr>
<tr><td>Assessment methods and criteria</td><td colspan="3"><table><tr><th>Subject passing criteria</th><th>Passing threshold</th><th>Percentage of the final grade</th></tr><tr><td>software programs (lab work)</td><td>50.0%</td><td>100.0%</td></tr></table></td></tr>
<tr><td rowspan="3">Recommended reading</td><td>Basic literature</td><td colspan="2">T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein. Introduction to Algorithms, Third Edition. The MIT Press, Cambridge, 2009. ISBN 978-0262033848.</td></tr>
<tr><td>Supplementary literature</td><td colspan="2">A.V. Aho, J.E. Hopcroft, J.D. Ullman. Data Structures and Algorithms, 1st Edition. Pearson, 1985. ISBN 978-0201000238.</td></tr>
<tr><td>eResources addresses</td><td colspan="2">Adresy na platformie eNauczanie:</td></tr>
<tr><td>Example issues/ example questions/ tasks being completed</td><td colspan="3">Determine the order of a given function using the big-O notation, e.g., n^3+3n+2^n.

Describe a selected sorting algorithm and state its computational complexity.

Implement the DFS algorithm in a programming language of your choice.

Determine the computational complexity of an algorithm written in pseudocode.</td></tr>
<tr><td>Work placement</td><td colspan="3">Not applicable</td></tr>
</table>

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