

Subject card

Subject name and code	Advanced algorithms, PG_00155225						
Field of study	Informatics						
Date of commencement of studies	October 2024		Academic year of realisation of subject		2024/2025		
Education level	Master'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 polish		
Semester of study	2		ECTS credits		7.0		
Learning profile	academic		Assessment form				
Conducting unit							
Name and surname of lecturer (lecturers)	Subject supervisor		dr Janusz Dybizbański				
	Teachers		dr Janusz Dybizbański				
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		115.0	175
Subject objectives	The aim of the course is to familiarize students with the basic techniques of designing parallel algorithms.						
Learning outcomes	Course outcome		Subject outcome		Method of verification		
	[INFMU2_U03] designs, analyzes for correctness and computational complexity, and builds algorithms using advanced programming techniques and data structures		can design an algorithm using the balanced tree technique, pointer jumping, and Euler cycle can determine the computational complexity of the designed algorithms		[SU1] oral statement/conversation/discussion [SU4] test/exam - oral or written [SU8] observation of student's independent or team work		
	[INFMU2_W04] knows complex data structures and advanced methods of algorithmic solution of computationally difficult problems (exponential algorithms, approximation, heuristics)		the student knows selected data structures used by parallel algorithms the student knows the techniques of designing parallel algorithms		[SW4] test/exam - oral or written [SW1] oral statement/conversation/discussion		
	[INFMU2_U05] can apply known algorithms in specific situations, can effectively select the type of algorithm depending on the problem at hand		can design an algorithm using the balanced tree technique, pointer jumping, and Euler cycle		[SU1] oral statement/conversation/discussion [SU4] test/exam - oral or written [SU8] observation of student's independent or team work		
	[INFMU2_K03] is able and ready to formulate opinions on basic IT issues		the student understands the need to develop his or her knowledge		[SK1] oral statement/conversation/discussion [SK8] observation of student's independent or team work		
	[INFMU2_K01] knows the limits of his own knowledge and understands the need for further learning		the student can formulate opinions about parallel algorithms		[SK1] oral statement/conversation/discussion [SK8] observation of student's independent or team work		

Subject contents	- assumptions of the PRAM model, possible conflicts and their resolution in PRAM submodels - two methods of writing parallel algorithms - parameters of parallel algorithms - parallel algorithm design methods: - balanced tree method, - pointer jumping, - divide-and-conquer, - symmetry breaking, - pipelinning, - Euler cycle technique, - selected algorithms: arithmetic expression tree evaluation, minimum spanning tree, sorting, cycle coloring		
Prerequisites and co-requisites	no requirements		
Assessment methods and criteria	Subject passing criteria	Passing threshold	Percentage of the final grade
	activity in classes	0.0%	5.0%
	tests	50.0%	45.0%
	exam	50.0%	50.0%
Recommended reading	Basic literature	[1] Joseph JaJa, An Introduction to Parallel Algorithms, Addison-Wesley, [2] Selim Akl, The Design and Analysis of Parallel Algorithms, Prentice-Hall.	
	Supplementary literature	no applicable	
	eResources addresses	Adresy na platformie eNauczanie:	
	Example issues/ example questions/ tasks being completed	no applicable	
Work placement	Not applicable		

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