STUDY MODULE DESCRIPTION FORM

Name of the module/subject **Algorithms and Complexity**	Code **1010334511010334958**

Field of study **Information Engineering**	Profile of study (general academic, practical) **(brak)**	Year /Semester **1 / 1**
Elective path/specialty **-**	Subject offered in: **Polish**	Course (compulsory, elective) **obligatory**

Cycle of study: **First-cycle studies**	Form of study (full-time,part-time) **part-time**

No. of hours Lecture: **16** Classes: **8** Laboratory: **8** Project/seminars: **-**	No. of credits **5**

Status of the course in the study program (Basic, major, other) **(brak)**	(university-wide, from another field) **(brak)**

Education areas and fields of science and art	ECTS distribution (number and %**)**
technical sciences	**5 100%**
Technical sciences	**5 100%**

Responsible for subject / lecturer:

dr inż. Krzysztof T. Zwierzyński
email: Krzysztof.Zwierzynski@put.poznan.pl
tel. +48 61 665 3755
Wydział Elektryczny
ul. Piotrowo 3A 60-965 Poznań

Prerequisites in terms of knowledge, skills and social competencies:

1	**Knowledge**	The student has a basic knowledge resulting from the high school.
2	**Skills**	The student can carry out tasks arising from the high school curriculum.
3	**Social competencies**	The student has the social skills acquired in high school.

Assumptions and objectives of the course:

The aim of the course is to familiarize students with the methods of constructing algorithms using basic techniques, including the analysis of computational complexity.

Study outcomes and reference to the educational results for a field of study

Knowledge:

1. The student has structured and theoretically founded knowledge of the basic algorithms and analysis techniques for designing algorithms, abstract data structures and their implementation, computationally difficult problems. - [K_W04]

Skills:

1. Student can construct algorithms using basic algorithmic techniques and analyze their complexity. - [K_U09]

2. He can assess the suitability of the student routine methods and tools for solving simple problems typical for computer engineering, and select and use appropriate technologies. - [K_U22]

Social competencies:

1. The student is aware of the importance of the accurate completion of the project, notational standards of behavior, respect for linguistic correctness and timely submissions - [K_K07]

Assessment methods of study outcomes

The lecture is classified on the basis of test results. Formal requirement is to get more than half of the maximum number of points added up for all the responses received.

Exercises are classified on the basis of test activities including first completion of the class. Formal criterion of assessment is to test more than half the maximum number of points added up for all the responses received.

Laboratories are classified on the basis of current activity in class (30 points) and two control tests (30 + 40 points). Is required to obtain at least 50 points. The first test verifies the student's skills in designing algorithms using static data structures. The second test verifies the skills in the use of recursion and dynamic data structures.

Course description

The problem, the algorithm, the computational complexity of time and space, the problem of decision-making, problem optimization. Designing efficient algorithms: data structures (lists, stacks, queues, priority queues, hash table), representations of the set (list, vector bit, array), graph representations (matrix neighborhood, the neighborhood list), binary tree and orders going through the tree (preorder, postorder, inorder), recursion, the strategy of 'divide and conquer', balancing, dynamic programming, greedy algorithm, the search of relapses, heuristics. Sorting algorithms. Search, selection. Data structures for the tasks on sets: primary operations on sets, dictionaries, hashing, binary search tree binary search. Algorithms on graphs: spanning tree with minimal cost, search in breadth and depth, strong consistency. Matrix multiplication and related operations. Integer arithmetic. The hierarchy of the complexity of the problem: models of computation, classes P and NP, NP-complete problems. Undecidability.

Update 2017: cubit, quantum register, quantum gates, Shor algorithm.

Laboratories include: tracing of both finished and designed by the students implement algorithms discussed in the lecture and skills development in the design of data structures and computational complexity analysis of algorithms and memory.

Exercises performed on the task of designing algorithms with respect to their computational complexity.

Basic bibliography:

1. Cormen, Thomas H.; Leiserson, Charles E.; Rivest, Ronald L.; Stein, Clifford, Introduction to Algorithms, MIT Press., 2009.

2. K.T. Balińska, Projektowanie algorytmów i struktur danych, wyd. 4, Wydawnictwo Politechniki Poznańskiej, 2011.

3. M. Sawerwain, J. Wiśniewska, Informatyka kwantowa, PWN, Waszawa 2015.

Additional bibliography:

1. K.T.Balińska, K.T. Zwierzyński, Projektowanie algorytmów grafowych, Wydanie II, Wydawnictwo Politechniki Poznańskiej, 2004.

2. W. Lipski, Kombinatoryka dla programistów, Wydawnictwa Naukowo-Techniczne, Warszawa 2007.

Result of average student's workload

Activity	Time (working hours)
1. Participation in lectures, exercises, preparation of programs for laboratory classes and individual work with manual	150

Student’s workload

Source of workload	hours	ECTS
Total workload	150	5
Contact hours	80	3
Practical activities	50	2