# COURSE DESCRIPTION CARD - SYLLABUS

Course name
Algorithms and Data Structures [S1DSwB1>AiSD]

## Course

Field of study
Data Science in Business

Year/Semester
1/1

Area of study (specialization)
–

Profile of study
general academic

Level of study
first-cycle

Course offered in
Polish

Form of study
full-time

Requirements
compulsory

## Number of hours

Lecture
0

Laboratory classes
0

Other
0

Tutorials
30

Projects/seminars
0

## Number of credit points

3,00

## Coordinators

Grzegorz Nowak
grzegorz.nowak@put.poznan.pl

dr inż. Marcin Nowak
marcin.nowak@put.poznan.pl

## Lecturers

## Prerequisites

Basic mathematical knowledge (including logic, operations on numbers, sequences), the ability to think logically, and a readiness to learn new tools and solve problems in an algorithmic manner.

## Course objective

The aim of the course is to introduce students to fundamental data structures and algorithm design methods, as well as to develop their analytical thinking skills and the ability to solve problems using an algorithmic approach.

## Course-related learning outcomes

Knowledge:
1. Defines fundamental concepts related to algorithms and data structures, including computational complexity and algorithm design strategies [DSB1_W01]
2. Characterizes classical search and sorting algorithms and describes their practical applications

[DSB1_W02]
3. Describes the structure and functioning of basic data structures, such as stacks, queues, lists, binary trees (including BSTs), and heaps [DSB1_W03]

Skills:
1. Analyzes computational complexity of algorithms, evaluates their efficiency in problem-solving contexts [DSB1_U03]
2. Implements sorting and search algorithms, and operations on fundamental data structures in a selected programming language [DSB1_U02]
3. Designs and applies appropriate data structures to effectively solve algorithmic problems [DSB1_U01]
4. Applies algorithm design strategies such as dynamic programming, greedy approach, and divide and conquer to develop efficient computational solutions [DSB1_U04]

Social competences:
1. Demonstrates readiness to independently broaden knowledge in the field of algorithms and data structures and to follow recent trends in this domain [DSB1_K01]
2. Collaborates effectively in teams during projects involving implementation and analysis of algorithms and data structures [DSB1_K02]

## Methods for verifying learning outcomes and assessment criteria

Learning outcomes presented above are verified as follows:

Tutorials:
There are two quizzes, each graded in the form of points - with a maximum of 50 points per quiz. The final grade is the sum of the points obtained from both quizzes. The first quiz takes place halfway through the course, and the second one at the end. The passing threshold is 50 points in total from both quizzes.

## Programme content

The course covers fundamental topics in algorithms and data structures. Students learn methods for analyzing computational complexity and algorithm design strategies such as dynamic programming, greedy algorithms, and divide and conquer. Classical searching and sorting algorithms are discussed, along with data structures including stacks, queues, heaps, lists, binary trees (including BSTs and balanced trees), as well as dictionaries and sets. The course also introduces the basics of graph algorithms, such as DFS, BFS, and Dijkstra’s algorithm. The classes are practical in nature, with an emphasis on selecting appropriate data structures and algorithms to solve analytical problems.

## Course topics

Introduction to algorithmics and algorithm analysis
Computational complexity in practice
Algorithm design strategies
Searching: linear and binary (on sorted lists)
Sorting algorithms
Data structures: stacks and queues - applications
Priority queue and heaps
Recursion and recursive algorithms
Introduction to dynamic programming
Singly and doubly linked lists
Binary trees and BSTs (Binary Search Trees)
Balanced trees - concept and applications
Dictionaries and sets - from a data structure perspective
Selected graph algorithms: graph representation, DFS and BFS traversals, Dijkstra’s algorithm and shortest path

## Teaching methods

Tutorials: problem-solving tasks, case studies, group work

## Bibliography

Basic:
Bhargava, A. (2022). Algorytmy. Ilustrowany przewodnik. Helion
Cormen T.H., Leiserson C, E., Rivest R.L., & Stein, C. (2024). Wprowadzenie do algorytmów, PWN

Additional:
Kubale, M. (2023). Łagodne wprowadzenie do analizy algorytmów. Wydawnictwo Politechniki Gdańskiej. Gdańsk.

## Breakdown of average student's workload

| | Hours | ECTS |
|---|---|---|
| Total workload | 75 | 3,00 |
| Classes requiring direct contact with the teacher | 30 | 1,00 |
| Student's own work (literature studies, preparation for laboratory classes/ tutorials, preparation for tests/exam, project preparation) | 45 | 2,00 |