

**Module Description, available in: EN**

## *Advanced Algorithms and Data Structures*

**General Information****Number of ECTS Credits**

3

**Module code**

FTP\_AdvAlgDS

**Valid for academic year**

2025-26

**Last modification**

2019-12-08

**Coordinator of the module**

Fabrizio Grandoni (SUPSI, fabrizio.grandoni@supsi.ch)

**Explanations regarding the language definitions for each location:**

- Instruction is given in the language defined below for each location/each time the module is held.
- Documentation is available in the languages defined below. Where documents are in several languages, the percentage distribution is shown (100% = all the documentation).
- The examination is available 100% in the languages shown for each location/each time it is held.

	Lausanne			Lugano	Zurich		
<b>Instruction</b>				X E 100%			
<b>Documentation</b>				X E 100%			
<b>Examination</b>				X E 100%			

**Module Category**

FTP Fundamental theoretical principles

**Lessons**

2 lecture periods and 1 tutorial period per week

**Entry level competences****Prerequisites, previous knowledge**

Familiarity with basic discrete math, logic and probability theory. Familiarity with one programming language such as C++, Java, or similar languages.

**Brief course description of module objectives and content**

Algorithms are at the heart of every computer program. Informally, an algorithm is a procedure to solve a (computational) problem within a finite number of elementary steps. The same problem can be addressed with different algorithms, hence it is important to compare the different options in order to choose the best one. Experimental analysis is one way to perform such comparison, but it has several limits. The main goal of this class is to learn how to analyze the performance of a given algorithm in a formal mathematical way. We will focus on some fundamental polynomial-time-solvable problems. Along the way, we will study some of the main techniques to design efficient algorithms, among which the use of efficient data structures.

## Aims, content, methods

### Learning objectives and acquired competencies

The main goal of this course is to learn basic techniques to design algorithms and data structures for fundamental polynomial-time-solvable problems, and mathematical tools to analyze their performance from a theoretical point of view.

### Contents of module with emphasis on teaching content

A. Analytical tools: worst-case analysis and asymptotic notation; analysis of recursive algorithms; analysis of randomized algorithms; amortized analysis.

B. Algorithmic techniques: greedy; dynamic programming; divide et impera; randomization; fast data structures.

C. Data structures: search trees; priority queues; union-find; hash tables.

D. Polynomial-time problems: sorting and selection; shortest paths; minimum spanning tree; maximum flow; maximum matching.

### Teaching and learning methods

Interactive lectures both for theory and exercises.

### Literature

- T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein. Introduction to Algorithms. MIT Press. Third edition.

- J. Kleinberg, E. Tardos. Algorithm Design. Addison-Wesley.

## Assessment

### Additional performance assessment during the semester

The module does not contain an additional performance assessment during the semester

### Basic principle for exams

**As a rule, all the standard final exams for modules and also all resit exams are to be in written form**

### Standard final exam for a module and written resit exam

#### Kind of exam

written

#### Duration of exam

120 minutes

#### Permissible aids

No aids permitted

**Exception: In case of an electronic Moodle exam, adjustments to the permissible aids may occur. Lecturers will announce the final permissible aids prior to the exam session.**

### Special case: Resit exam as oral exam

#### Kind of exam

oral

#### Duration of exam

30 minutes

#### Permissible aids

No aids permitted