Universidad de Alcalá

# Teaching Guide

# ALGORITHMS AND COMPLEXITY

## Bachelor's Degree in Computer Science

## Universidad de Alcalá

**Academic Year 2019/2020**

**2nd Year – 2nd Term**

# TEACHING GUIDE

| | |
|---|---|
| Course name: | **Algorithms and Complexity** |
| Course code: | **780021** |
| Degree program: | **Computer Science** |
| Departament: | **Computer Science** |
| Type: | **Optional of specific technology** [1] |
| ECTS credits: | **6** |
| Year and term: | **Second year/ Second term** |
| Teacher: | Francisco Manuel Sáez de Adana Herrero |
| Office hours: | Indicated on the first day of lessons |
| Classes offered on: | English/Spanish |

## 1. PRESENTATION

Algorithm theory is one of the pillars of programming and its relevance is shown in the development of any application, beyond the mere construction of programs. This subject deals with the analysis and design of algorithms, and aims to expose the student to the necessary techniques for its design and implementation, as well as to present the tools that allow it to measure its effectiveness and efficiency.

The objective of the subject is not to provide the student with known solutions to specific problems, but to provide the student with the basic algorithmic techniques that will allow him to approach the development of correct and efficient programs to solve non-trivial problems, studying in the development of the subject the techniques and patterns of design of the most important algorithms (families of algorithms), and deepening in the design and evaluation of the algorithms themselves.

### Prerequisites and Recommendations

Because the subject presents different techniques and advanced data structures for solving problems, the student must know and have practice in imperative programming, and especially in the use of recursion. Therefore, it is recommended that the student has successfully passed the subjects of Fundamentals of Programming and Data Structures.

Regarding the mathematical part of the subject, the student should be familiar with the concepts of limit, combinatorial and resolution systems of equations, so the subjects

[1] Optional subjects of specific technology result in the Degree's mention, so they are obligatory in this Degree.

of Mathematical Fundamentals and Discrete Mathematics should have been successfully completed

## 2. COMPETENCES

### General skills:

CG8 Knowledge of basic materials and technologies that enable learning and development of new methods and technologies, as well as to equip them with great versatility to adapt to new situations.

CG9 Ability to solve problems with initiative, decision making, autonomy and creativity. Ability to communicate and transmit knowledge and skills of the profession of Technical Engineer.

### Specific skills:

CC1 Ability to have a thorough understanding of the fundamental principles and models of computation and know how to apply to interpret, select, assess, model, and create new concepts, theories, uses and technological developments related to computer science.

CC3 Ability to evaluate the computational complexity of a problem, meet algorithmic strategies that can lead to resolution and recommend, develop and implement one that guarantees the best performance according to the requirements.

CC4 Ability to know the basics, paradigms and own techniques of intelligent systems and analyze, design and build systems, services and applications that use these techniques in any scope.

### Outcomes:

LO1-Reason the theoretical foundations of algorithmic schemes.
LO2-Learn to perform the analysis of an algorithm, taking into account its efficiency, cost, cases, asymptotic notation, etc., understanding it as a quality criterion.
LO3-Apply with criteria the basic techniques of efficiency analysis and algorithm design.
LO4-Analyze the correctness of an algorithm using simple verification techniques.
LO5-Recognize the basic algorithmic schemes: divide and conquer, voracious, dynamic programming, backtracking, ramification and dimensioning algorithms.
LO6-Know some classical algorithms for fundamental problems, and recognize situations where you can reuse the repertoire of data structures and classical algorithms.
LO7-Write recursive programs about types that are not necessarily basic and be able to reason about their correctness and efficiency.
LO8-Be able to use advanced algorithm design and analysis techniques, and know how to customize general algorithmic schemes to solve problems.

LO9-Have criteria that allow, during the stages of specification, design and implementation, to choose the most appropriate alternative, and have elements to argue in a reasoned manner the choices made.
LO10-Take contact with some fundamental techniques of design and analysis of algorithms, as well as with some advanced programming techniques.
LO11-Know how to identify the most relevant components of a problem and select the most appropriate algorithmic technique for resolution.
LO12-Be able to select the most suitable types of data to improve the efficiency of an algorithmic solution.
LO13-Justified design of efficient algorithmic solutions for problems typical of the academic level of the subject.

## 3. COURSE CONTENTS

BLOCK 1: Introduction to the algorithm theory
Unit 1. Introduction to the algorithm theory: Introduction, basic concepts of the analysis of algorithms, calculation of the size of the problems, asymptotic notation, efficiency orders, resolution of recurrences by means of the characteristic equation.

BLOCK 2: Basic algorithms
Unit 2. Voracious algorithms: Data structures usually employed, concept of voracious algorithms, frequent efficiencies and examples.
Unit 3. Divide and conquer algorithms: Common data structures, concept of divide and conquer algorithms, frequent efficiencies and examples.
Unit 4. Algorithms of dynamic programming: Extension of algorithms divide and conquer, reuse of cases through data structures, concept of dynamic programming algorithms, frequent efficiencies and examples.

BLOCK 3: Advanced algorithms
Unit 5. Use of intensive recursion: Useful implementations of the necessary TADs, exploration of graphs, advantages and disadvantages of the intensive use of recursion, backtracking algorithms, problems of branching and pruning, delimitation of solutions.

BLOCK 4: Non-determinism and complexity classes
Unit 6. Non-deterministic algorithms: Introduction to non-determinism, numerical probabilistic algorithms, Monte Carlo algorithms, Las Vegas algorithms, Sherwood algorithms, validity of non-deterministic results and advantages and disadvantages against determinism.
Unit 7. Complexity: Concept of complexity, classification of P and NP problems, basic complexity results, impossibility of solving all problems

| Content blocks | Hours |
|---|---|
| **Introduction to algorithm theory**<br>Introduction to algorithm theory | 12 hours |
| **Basic algorithms**<br>Voracious algorithms<br>Divide and conquer algorithms<br>Algorithms of dynamic programming | 24 hours |
| **Advanced algorithms**<br>Use of intensive recursion | 14 hours |
| **Non-determinism and complexity classes**<br>Non-determinist algorithms<br>Complexity | 6 hours |

(*) Include IET (Intermediate evaluation test)

## 4. TEACHING/LEARNING METHODOLOGY. FORMATIVE ACTIVITIES

### 4.1. Credit distribution (especify in hours)

| | |
|---|---|
| Lesson hours: | 56 hours + PEC 4 hours |
| Independent study hours: | 90 hours |
| Total hours | 150 hours |

### 4.2. Methodological strategies, materials and didactic resources

The subject Algorithms and complexity is organized as a semester subject of 6 ECTS (150 hours).

In the teaching-learning process of the aforementioned contents, the following training activities will be used:

• Theoretical lessons.

• Practical lessons: solving problems.

• Laboratory sessions

.

• Tutorials: individual and/or group.

In addition, depending on the nature of the different parts of the subject matter, the following training activities may be used, among others:

• Preparation of works with individual responsibility, but with information management as a team.

• Put in common the information, problems and doubts that appear in the realization of the works.

• Organization and realization of public exhibition days with oral presentations and discussion of results.

• Use of Virtual Classroom Platform.

Class activities:

1. In the classroom: exhibition and discussion of the basic knowledge of the subject. Approach and theoretical resolution of exercises and related assumptions. Oriented to the teaching of the specific competences of the subject, especially those related to the techniques of creation and improvement of algorithms.

2. In the laboratory: approach and development of practical exercises that contribute to the understanding of the subject and to the development of the practice in the analysis of problems, critical reasoning and understanding of the methods of resolution.

Individual study activities:

1. Analysis and assimilation of the contents of the subject, resolution of problems, bibliographical consultation, preparation of individual and group work, realization of self-evaluations. Specially oriented to the development of methods for self-organization and planning of individual and team work.

2. Tutorials: individual and group counseling during the teaching-learning process, either in person or remotely.

Materials and resources:

• Reference bibliography on the subject.

• Personal computers.

• Development environments and manuals for their use.

• Internet connection.

• Virtual Classroom platform and manuals for their use.

## 5. ASSESMENT

The students shall adhere to the evaluation procedures according to título 2 (arts. 9 y 10) de la normativa de evaluación de aprendizajes de la UAH.

In any case, students will preferably be offered a continuous assessment system that has characteristics of formative assessment, so that it serves as feedback in the teaching-learning process by the student. For this purpose, the following evaluation procedures will be established:

### Assessment criteria

The criteria must meet the degree of acquisition of skills by the student. For this, the following criteria are defined:

AC1- Understanding and use of basic concepts and techniques about algorithms, design, analysis, efficiency, and cost of algorithms

AC2- Understanding and using the elements, techniques and use of basic algorithms, their data structures, basic strategies, voracity, usual efficiencies, reuse of cases and dynamic programming.

AC3- Understanding and using techniques of intensive recursion, advantages and disadvantages, problems of branching and pruning.

AC4- Understanding and use of non-determinism and probabilistic methods, validity of results, particular advantages and disadvantages, as well as the fundamental analysis of the problems of complexity and the algorithmic irresolubility of certain problems.

AC5-Ability for practical application in programmed practical laboratory cases.

### Evaluation procedures

This section indicates the evaluation instruments that will be applied in each of the evaluation criteria.

1.-Intermediate evaluation test (IET1) Of a written nature, consisting of solving problems and issues on thematic blocks 1 and 2.
2.-Intermediate evaluation test (IET2) Of a written nature, consisting of the resolution of problems and questions on thematic blocks 3 and 4
3.-Intermediate laboratory evaluation test (ILET1), consisting of the delivery and defense of a series of exercises related to the knowledge taught in laboratory class, as a practical application of them.
4.-Final evaluation test (FET) Written, consisting of solving problems and questions about the four thematic blocks of the subject and their laboratory practices, as well as the eventual delivery and oral defense of a work of laboratory.

## Grading criteria

This section quantifies the evaluation criteria for the passing of the subject

1) Ordinary call. Continuous assessment: In the ordinary call-continuous assessment the relationship between the criteria, instruments and qualification is as follows:

| Competences | Learning Outcomes | Assessment Criteria | Evaluation Procedures | Weight on the grade |
|---|---|---|---|---|
| CG8, CG9, CC1, CC3 | LO1-7, LO11 | AC1-4 | IET1 | 30% |
| CG8, CG9, CC1, CC3, CC4 | LO8-13 | AC1-4 | IET2 | 35% |
| CG8, CG9, CC1, CC3, CC4 | LO1-13 | AC5 | IELT1 | 35% |

As a general criterion, those students in ordinary call who do not show up for the assessment of the laboratory will be understood as not presented

2) Ordinary call. Final assessment. In the final assessment, the student must perform a final assessment test (PEF) that will include the theoretical and practical parts of the subject.

| Competences | Learning Outcomes | Assessment Criteria | Evaluation Procedures | Weight on the grade |
|---|---|---|---|---|
| CG8, CG9, CC1, CC3, CC4 | LO1-13 | AC1-4 | FET | 100% |

3) Extraordinary call: In the extraordinary assessment, the student will have to make a final evaluation assessment (PEF) that will include the theoretical and practical parts of the subject.

| Competences | Learning Outcomes | Assessment Criteria | Evaluation Procedures | Weight on the grade |
|---|---|---|---|---|
| CG1, CG8, CC1, CC3, CC4 | LO1-13 | AC1-4 | FET | 100% |

# 6.BIBLIOGRAPHY

There are versions in English of the recommended bibliography. However, the Spanish translations are listed in this section because they are the ones available at the library.

## Core bibliography


- A.V. Aho, J.E. Hopcroft, J.D. Ullman, Estructuras de datos y algoritmos, Addison-Wesley Iberoamericana, 1988.
- G.Brassard, P. Bratlet, Fundamentos de algoritmia, Prentice Hall, 1999.
- G.H. Gonnet, R. Baeza-Yates, Handbook of Algorithms and Data Structures, AddisonWesley, 1991.
- M.A. Weiss, Estructuras de datos y algoritmos, Addison-Wesley Iberoamericana, 1995.


## Further bibliography


- T. Cormen, C. Leirserson, R. Rivest, *Introduction to Algorithms*, McGraw Hill, 1998.
- R. Peña, *Diseño de Programas. Formalismo y Abstracción*. Prentice Hall, 1997.
- N. Wirth, *Algoritmos + Estructuras de datos = Programas*. Editorial del Castillo, 1984.
- N. Ziviani, *Diseño de algoritmos con implementaciones en Pascal y C*, Thompson, 2007.