

Analysis and Design of Algorithms

Code: 102783
ECTS Credits: 6

Degree	Type	Year	Semester
2502441 Computer Engineering	OB	3	1
2502441 Computer Engineering	OT	4	1

Contact

Name: Aura Hernández Sabaté
Email: Aura.Hernandez@uab.cat

Use of Languages

Principal working language: catalan (cat)
Some groups entirely in English: No
Some groups entirely in Catalan: Yes
Some groups entirely in Spanish: No

Teachers

Francisco Javier Sánchez Pujadas

Prerequisites

There are no formal prerequisites, but students should be familiar with the most basic concepts of the following subjects:

First year of the degree:

- Fundamentals of Computer Science
- Programming Methodology
- Discrete Mathematics

Second grade course:

- Programming Laboratory

Objectives and Contextualisation

This subject is the continuation of the programming subjects seen in the first and second courses, such as Fundamentals of Computer Science, Programming Methodology and Programming Laboratory. Starting from the premise that the student already has some basic knowledge about programming, advanced knowledge is given so that the student is able to make an analysis of the algorithms designed to find those that solve a problem posed in an optimal way, taking into account the requirements of efficiency and/or intrinsic precision of each environment.

Therefore it is expected that at the end of the course the students will know:

- Formally specify problems and programs, and verify them.
- Use formal tests to validate programs and invariants to design based on contracts.
- Calculate the algorithmic and computational complexity of an algorithm.

Besides, they will know how to choose different styles and paradigms of algorithm design such as:

- recursivity
- backtracking
- dynamic programming
- probabilistic algorithms
- Etc.

Competences

- Computer Engineering
- Acquire thinking habits.
- Have the capacity to conceive, develop and maintain computer systems, services and applications employing the methods of software engineering as an instrument to ensure quality.
- Have the capacity to define, evaluate and select hardware and software platforms for the development and execution of computer systems, services and applications.
- Have the capacity to evaluate the computational complexity of a problem, know algorithmic strategies that can lead to their resolution and recommend, develop and implement these to guarantee the best performance in accordance with the established requirements.
- Work in teams.

Learning Outcomes

1. Develop a capacity for analysis, synthesis and prospection.
2. Develop scientific thought .
3. Evaluate the complexity of algorithms and identify their weak points.
4. Identify and select suitable algorithmic strategies for problems.
5. Know the operation mechanisms of different programming paradigms.
6. Select the best programming technique for solving complex problems .
7. Work cooperatively.

Content

- Topic 1. Pre-conditions, post-conditions and invariants
- Topic 2. Recursivity and Computational Complexity
- Topic 3. Backtracking
- Topic 4. Branch & Bound
- Topic 5. Dynamic Programming
- Topic 6. Greedy
- Topic 7. Probabilistic Algorithms
- Topic 8. Analysis of algorithms

Methodology

Bearing in mind that the final objective of the subject is that the students are able to analyze and design algorithms efficiently according to a given problem, the work of the students is the central axis of their learning, accompanied and guided by the teaching staff. For this reason, the face-to-face classes will be highly practical and will focus on the students consolidating the knowledge that is the objective of learning this subject.

The general methodology of the subject can be divided into three phases:

Preparation of the class: the objective of this phase is that the students can learn the concepts that will be worked on in the following session through various activities offered by the teaching staff, such as watching videos, reading texts, etc.

Face-to-face class: the aim of this phase is to consolidate the concepts seen and put them into value within the context of the subject. The teaching staff will ensure that the students delve deeper into these concepts through exercises (more or less) guided during the session.

Autonomous work: in order for the student body to become fluent in the programming of the algorithms seen, they will have to do part of the work on their own, whether they are individual exercises or within a project.

TRANSVERSE COMPETENCES: In this course the following transversal competences will be worked on and evaluated:

- T01.02 - Developing the capacity for analysis, synthesis and foresight
- T01.03 - Developing scientific thinking
- T03.01 - Working cooperatively

Activities

Title	Hours	ECTS	Learning Outcomes
Type: Directed			
Face-to-face classes	50	2	3, 5, 1, 4, 6, 7
Type: Supervised			
Follow-up in the assimilation of theoretical concepts	4	0.16	3, 5, 1, 4, 6
Reinforcement and follow-up in the resolution of projects	4	0.16	3, 1, 4, 6
Type: Autonomous			
Autonomous work	12	0.48	3, 5, 2, 1, 4, 6, 7
Development of a project	36	1.44	3, 5, 2, 1, 4, 7
Exams preparation	12	0.48	3, 5, 1, 4, 6
Pre-class preparation	12	0.48	2, 1, 4, 6
Preparation of practical reports	8	0.32	2, 1, 7

Assessment

Three types of activities will be evaluated independently and the weighted sum of them will give the final mark. These three activities are:

- Synthesis tests (PS)
- Assessable exercises (EA)
- Practical project (P)

1. The first part (PS) consists of two partial tests in which the students will be evaluated individually. The minimum grade to pass each test is 4, but the average of the two must reach 5.

2. The second part (EA) will consist of the delivery of small exercises throughout the course. The final grade will come from the weighted sum of the fixed deliveries that are requested.

3. The third part (P) will be evaluated in a group (with the delivery of a project) and individual (with the evaluation of a written test). The final mark will be obtained from the weighted sum of the two previous marks.

The minimum grade to pass the project is 5, while the individual exam must be passed with a minimum grade of 3.5. The final grade of this part must be a minimum of 5.

To pass the course it is necessary that the evaluation of each of the parts exceeds the minimum required and that the total evaluation exceeds 5 points.

TRANSCRIPT, HONORARY AND NON-ASSESSABLE MATRICULATION

In case of not passing the subject due to the fact that some of the evaluation activities do not reach the minimum required grade, the numerical grade of the file will be the maximum grade between the suspended partial, or in its defect, the maximum suspended grade. With the exceptions of students who:

- 1) do not participate in any of the evaluation activities, which will be awarded the grade of "not evaluable" (any student who submits a practice or a scheduled assessment will have a grade),
2. they have committed irregularities in an act of evaluation, which will be awarded the lowest value between 3.0 and the maximum numerical score referred to above (and therefore approved by compensation will not be possible).

As many registrations as possible will be given within the regulations of the university, as long as the minimum grade is 9.

RECUPERATION

PS: In the case of suspending or not appearing at any of the individual tests, the day assigned to the official week of exams can be made up.

EA: This part will not have the possibility of recovering.

Q: In the case of suspension of the individual evaluation of the project (it is necessary to have presented the first time), the day assigned to the official week of examinations can be made up. The project is not recoverable.

VALIDATION

Projects from previous years are not validated.

PLAGIARISM, COPIES, ETC:

Without prejudice to other disciplinary measures deemed appropriate, and in accordance with current academic regulations, irregularities committed by a student that may lead to a variation of the grade will be graded with a zero (0). Evaluation activities that are graded in this manner and by this procedure will not be recoverable. If it is necessary to pass any of these evaluation activities in order to pass the course, this course will be suspended directly, without the opportunity to recover it in the same course. These irregularities include, among others:

- the total or partial copy of a practice, report, or any other evaluation activity;
- allow copying;
- present a group work not done in its entirety by the members of the group;
- present as their own materials produced by a third party, even if they are translations or adaptations, and in general work with elements that are not original and exclusive to the student;
- have communication devices (such as mobile phones, smart watches, etc.) accessible during the individual theoretical-practical evaluation tests (examinations).

In short: copy, let copy or plagiarize in any of the evaluation activities is equivalent to a SUSPENSE with a grade lower than 3.5.

COMMUNICATION

The dates of evaluations and deliveries will be published in the document manager Caronte (<https://caronte.uab.cat/>) and may be subject to possible changes in programming for reasons of adaptation to

possible incidents. The Caronte will always be informed about these changes as it is understood that this is the usual platform for exchanging information between teachers and students.

Assessment Activities

Title	Weighting	Hours	ECTS	Learning Outcomes
Delivering Exercises	20	1	0.04	3, 5, 2, 1, 4, 6, 7
Final Exam (recuperation)	40	4	0.16	3, 5, 2, 1, 4, 6
First Theoretical-Practical Partial Exam	20	2	0.08	3, 5, 2, 1, 4
Group evaluation of the project	30	2	0.08	3, 5, 2, 1, 4, 6, 7
Individual project evaluation	10	1	0.04	2, 1
Second Theoretical-Practical Partial Exam	20	2	0.08	3, 5, 2, 1, 4, 6

Bibliography

- Fundamentos de Algoritmia, G Brassard P. Bratley. Prentice Hall.
- Técnicas de Diseño de Algoritmos, Rosa Guerequeta y Antonio Vallecillo.
- Técnicas de diseño de algoritmos, F. Perales, M. Mascaró. Universitat de les Illes Balears
- Thinking in C++ 2nd Edition by Bruce Eckel Volume 1 y Volume 2.
(<http://www.mindview.net/Books/TICPP/ThinkingInCPP2e.html>)
- El lenguaje de programación C++, Bjarne Stroustrup
- <http://en.cppreference.com/w/>
- <http://www.cplusplus.com/>