

Algorithm Design

Code: 104359
ECTS Credits: 6

2025/2026

Degree	Type	Year
Data Engineering	OB	2

Contact

Name: Enrique Ruiz Amakrache

Email: enrique.ruiz@uab.cat

Teachers

Pau Cano Ribe

Enrique Ruiz Amakrache

Teaching groups languages

You can view this information at the [end](#) of this document.

Prerequisites

Although it is not compulsory, it is highly advisable to have studied the following subjects:

- Fonaments de Programació
- Programació Avançada
- Estructures de Dades
- Anàlisi de Grafs i prop d'Informació
- Enginyeria l'Rendiment

The teaching team will assume that the student has previously acquired the knowledge taught in these subjects.

Objectives and Contextualisation

Based on the fact that the student already has some basic knowledge about programming, this course is focused on introducing different styles and paradigms of algorithm design. The main objective is that students develop skills in the design and analysis of algorithms in order to solve real-world problems effectively and efficiently according to the requirements established by a potential client.

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.

The algorithm aims to find the fastest way to solve problems, and this has two sides. The first, and most important, is to find algorithms with the least complexity, that is to say that they perform the least possible number of operations. The second corresponds to program the implementations of these algorithms in such a way that the execution is as fast as possible. Therefore, the objectives include learning about algorithm development techniques and implementing fast programs.

Competences

- Design efficient algorithmic solutions to computational problems, implement them in the form of robust software developments which are structured and easy to maintain, and verify their validity.
- Make a critical evaluation of work carried out.
- Students must be capable of collecting and interpreting relevant data (usually within their area of study) in order to make statements that reflect social, scientific or ethical relevant issues.

Learning Outcomes

1. Decide on the most suitable data-learning method depending on the characteristics of the data to be analysed.
2. Make a critical evaluation of work carried out.
3. Students must be capable of collecting and interpreting relevant data (usually within their area of study) in order to make statements that reflect social, scientific or ethical relevant issues.

Content

The subjects covered in the course will be:

- Topic 1. Revision of basic concepts (Pre-conditions, post-conditions, invariants, Computational complexity).
- Topic 2. Search algorithms: Taxonomy and strategies.
- Topic 3. Greedy.
- Topic 4. Divide and conquer.
- Topic 5. Backtracking.
- Topic 6. Branch & Bound.
- Topic 7. Dynamic programming.
- Topic 8. Stochastic Algorithms

Activities and Methodology

Title	Hours	ECTS	Learning Outcomes
Type: Directed			
Face-to-face classes	50	2	1
Type: Supervised			
Follow-up in the assimilation of theoretical concepts	4	0.16	2, 1
Follow-up in the assimilation of theoretical concepts	4	0.16	2, 1, 3
Type: Autonomous			
Autonomous work	12	0.48	2, 1
Development of a project	44	1.76	2, 1, 3
Exams preparation	12	0.48	2, 1
Preparation prior to the lectures	12	0.48	2, 1, 3

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 sessions will be highly practical and will focus on the students consolidating the knowledge that is the objective of learning this subject.

The methodology of the subject is 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.

Directed 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 to become fluent in the programming of the algorithms seen, the students will have to do part of the work on their own with

1. individual exercises that will have to be submitted for being evaluated
2. as part of a project that will be carried out throughout the course.

Programming project: As part of the autonomous work required of students, they will have to carry out a programming project that will be developed throughout the course, as they progress through the syllabus. Each part of the project will be related to one of the planned topics and will be presented during the class sessions. A few hours of the sessions will be devoted to the work to be done and also to monitoring the correct development of the project, as well as raising any doubts.

Annotation: Within the schedule set by the centre or degree programme, 15 minutes of one class will be reserved for students to evaluate their lecturers and their courses or modules through questionnaires.

Assessment

Continous Assessment Activities

Title	Weighting	Hours	ECTS	Learning Outcomes
Evaluation of activities developed in the classroom	5	1	0.04	2, 1, 3
Evaluation of activities developed out of the classroom	10	1	0.04	2, 1, 3
Final exam (recovery)	40	3	0.12	2, 1
First intermediate exam	20	2	0.08	2, 1
Group evaluation of the project	35	2	0.08	2, 1, 3
Individual project exam	10	1	0.04	2, 3
Second intermediate exam	20	2	0.08	2, 1

This subject does not accept unique evaluation.

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 activities throughout the course. These activities can be in the form of exercises, questionnaires, reports, etc. and can be done as part of the class sessions or can be proposed as homework. The final mark will be the weighted sum of the fixed assignments that are required.
3. The third part (P) will be assessed as a group (with the delivery of a project) and individually (with the assessment of a written test). The final mark will be obtained from the weighted sum of the group mark and the individual mark. The project consists of different deliveries throughout the course, the marks of which will make up the group mark. The minimum mark to pass the project is 5, while the individual exam must be passed with a minimum mark of 3.5. The final mark for this part of the course must be at least a 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.

P: In the case of failing the individual assessment of the project (it is necessary to have taken it the first time), it can be made up on the day assigned in the official exam week. The partial deliveries of the project can be recovered in the following deliveries, with a final mark of 80% (and therefore never higher than 8).

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 Caronte and may be subject to possible changes in programming for reasons of adaptation to possible incidents. 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.

Bibliography

- Fundamentos de Algorítmia, G Brassard P. Bratley. Prentice Hall.
- Análisis y diseño de algoritmos : un enfoque práctico, Eduardo Villegas Jaramillo, Luz Enith Guerrero Mendieta. (Primera edición.). Disponible en línea
- Técnicas de Diseño de Algoritmos, Rosa Guerequeta y Antonio Vallecillo.
- Técnicas de diseño de algoritmos, F. Perales, M. Mascaró. Universitat de les Illes Balears
- Karumanchi, Narasimha. "Algorithm Design Techniques: Recursion, Backtracking, Greedy, Divide and Conquer, and Dynamic Programming." (2018).
- Chun, Wesley. Core python programming. Vol. 1. Prentice Hall Professional, 2001.

Software

Python programming language (version 3.4 onwards) will be used during this course. We do not impose any restriction on the use of other languages.

Groups and Languages

Please note that this information is provisional until 30 November 2025. You can check it through this [link](#). To consult the language you will need to enter the CODE of the subject.

Name	Group	Language	Semester	Turn
(PAUL) Classroom practices	81	Catalan/Spanish	second semester	morning-mixed
(PAUL) Classroom practices	82	Catalan/Spanish	second semester	morning-mixed