UAB
Universitat Autònoma de Barcelona

Analysis and Design of Algorithms

Code: 102783
ECTS Credits: 6

2025/2026

Degree	Type	Year
Computer Engineering	OB	3
Computer Engineering	OT	4

Contact

Name: Francisco Javier Sánchez Pujadas

Email: javier.sanchez.pujadas@uab.cat

Teachers

Pau Cano Ribe

Teaching groups languages

You can view this information at the end of this document.

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. 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

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. Develope 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. Divide & Conquer
- Topic 4. Backtracking
- Topic 5. Branch & Bound
- Topic 6. Dynamic Programming
- Topic 7. Greedy
- Topic 8. Probabilistic Algorithms

- Topic 9. Analysis of algorithms

Activities and Methodology

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 project and exercises	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
Preparation of practical reports	8	0.32	2, 1, 7
Preparation prior to the lectures	12	0.48	2, 1, 4, 6

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 the 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.

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

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

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	10	1	0.04	3, 5, 2, 1, 4, 6, 7
Evaluation of activities developed out of the classroom	20	1	0.04	3, 5, 2, 1, 4, 6, 7
Final Exam (recuperation)	30	3	0.12	3, 5, 2, 1, 4, 6
First Theoretical-Practical Partial Exam	15	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 exam	10	1	0.04	2, 1
Second Theoretical-Practical Partial Exam	15	2	0.08	3, 5, 2, 1, 4, 6

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

- Synthesis tests (PS exams)
- Evaluable exercises (EA)
- Practical project (P)

1. The first part (PS) consists of carrying out two partial tests in which the students will be evaluated individually. The minimum grade to pass each test is 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 proposed within class sessions or can be proposed as homework to do at home. The final grade will come from the weighted sum of the set deliveries that are requested.

3. The third part (P) will be evaluated individually with the delivery of a project and the evaluation of a written test. The final grade will be obtained from the weighted sum of the group and individual grades. The project consists of different assignments throughout the course, the grades of which will make up the group grade. The minimum grade to pass the project is 5, while the individual exam must be passed with a minimum grade of 5. The final grade for this part must be at least 5.

The minimum grades of 5 will be 4 if students attend more than 50% of classes.

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

RECOVERY

PS: In the case of failing or not appearing for any of the individual tests, they can be made up on the day assigned to the official exam week.

EA: This part will not have the possibility of being made up.

P: In the event of failing the written project test (must have been submitted the first time), it can be recovered on the day assigned to the official exam week. Partial project submissions can be recovered in the following submissions, with afinal grade of 80% of the improvement in thegrade.

REVIEW OF QUALIFICATIONS

The review of exam grades will be done on the day indicated by the professor after each exam.

The review of submissions will be requested by e-mail and may be in person, non-face-to-face or by e-mail.

HONORARY ENROLLMENT

As many enrolments as possible within the university regulations will be given, provided that the final grade obtained is at least a 9.

NOT ASSESSABLE

Not having presented any assessable part of the subject.
The parts of the subject not presented have a grade of 0.

VALIDATION

Projects from previous years are not validated.

SUBJECT GRADE

Elements that will be evaluated:

Exams:

First theory part.

Second theory part.
Practice exam (verifies that work has been done in practice).
Theory retake exam (parts eliminate material).
Practice retake exam.
Problems:

Problems submitted by students.
Practice:

Practice submissions.
Theory Note: corresponds to two theory part exams on the subject. The part exams must be passed to eliminate material in the retake exam.
Problem Note: Corresponds to the note obtained for a series of problems that the student must submit. There is no retake for problem submissions.
Practice Note: In this section there is a note for the submissions and a note for the practice exam:

Submission Note: Corresponds to the note obtained in the practice submissions. Failed practical assignments can be made up in subsequent assignments. The value of the grade improvement will be 80%.

Practical exam grade: This is the score obtained in the practical exam. There will be a retake of the practical exam. The practical exam is given on the same day as the second part exam and the retake exam.
Class attendance is not mandatory but highly recommended to pass the subject.
There is automated class attendance control with the correction server.
Attendance and assessment of thesubject:
The minimum grade to be able to make an average will be 4 (MinAverage=4) if class attendance is greater than 50% of the classes. Otherwise, the minimum grade to make an average will be 5 (MinAverage=5).
This minimum grade applies to all exams, practical assignments and intermediate grades for theory and practicals.

The final grade for the subject is obtained by combining the evaluation of these three activities as follows:

The final grade for the subject is obtained by combining the evaluation of these three activities as follows:

If Theory Grade> = MinAverage and Practice Grade> = MinAverage THEN Final Grade = 0.4 * Theory Grade + 0.2 * Problem Grade + 0.4 * Practice Grade
ELSE Final Grade = MIN (Theory Grade, Practice Grade)
IF First Part Exam Grade>=MinAverage and Second Part Exam Grade>=MinAverage THEN Theory Grade = 0.5 * First Part Exam Grade + 0.5 * Second Part Exam Grade
ELSE Theory Grade = MIN (First Part Exam Grade, Second Part Exam Grade)
Problem Grade = Average of the problems submitted.
IF Practical exam grade >= MinAverage and Assignment grade>= MinAverage THEN Practical grade = 0.2 * Practical exam grade + 0.8 * Assignment grade
ELSE Practical grade = MIN(Practical exam grade, Assignment grade)
Submission grade = Weighted average of the practical assignments if they all exceed MinAverage, otherwise the minimum of the grades of the assignments.

The parts of the subject not presented have a grade of 0.

PLAGIA, 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 in the grade will be graded with a zero (0). Assessment activities graded in this way and by this procedure will not be recoverable. If it is necessary to pass any of these assessment activities to pass the subject, this subject will be directly failed, with no opportunity to recover it in the same course. These irregularities include, among others:
- total or partial copying of a practice, report, or any other assessment activity;
- allowing copying;
- presenting a group work not done entirely by the group members;
- presenting as one's own materials prepared by a third party, even if they are translations or adaptations, and in general works with non-original and exclusive elements of the student;
- having communication devices (such as mobile phones, smart watches, etc.) accessible during individual theoretical-practical assessment tests (exams).

In summary: copying, allowing copying or plagiarism in any of the assessment activities is equivalent to a FAIL with a grade lower than 3.5.

USE OF ARTIFICIAL INTELLIGENCE

A restricted use of AI is permitted. Usually to resolve doubts and facilitate learning. A use of AI that limits the student's learning will be considered as plagiarism or copying. So that the teacher can assess the illicit use of AI, he may ask students to explain how they have solved the problems or practices submitted. Not demonstrating that one has the necessary knowledge to do the work submitted will be considered as plagiarism.

COMMUNICATION

The dates of assessments and submissions will be published on the virtual campus and may be subject to possible changes in programming for reasons of adaptation to possible incidents. The Virtual Campus will always be informed about these changes since it is understood that this is the usual platform for exchanging

information between teachers and students. The synchronous communication platform will preferably be Teams.

SINGLE EVALUATION

It will be evaluated in the same way as the continuous assessment. But with the following differences:

The practices and problems will be submitted on the day of the subject's retake exam. In this case, the practices do not have the possibility of recovery, since the student cannot correct and see the gradeas many times as he wants before submission.
Due to the fact that there is no monitoring of the student's work, it will be mandatory to take a practice validation exam, which must be passed.
The theory and practice exams will be held on the day of the subject's retake exam.
If necessary, students will be called to take the theory and practical retake exams.

Bibliography

- Brassard, G., Bratley, P., & Garcia-Bermejo, R. (1997). *Fundamentos de algoritmia* (Vol. 86). Madrid: Prentice Hall.
- Guerequeta, R., & Vallecillo, A. (2019). *Técnicas de diseño de algoritmos*.
- Villegas Jaramillo, & Guerrero Mendieta, L. E. (2016). Análisis y diseño de algoritmos: un enfoque práctico / Eduardo Villegas Jaramillo, Luz Enith Guerrero Mendieta. (Primera edición.). Disponible en línia
- Eckel, B. (2021). Thinking in C++.
- Stroustrup, B. (1993). *C++ el lenguaje de Programación*. Eddison Wesley/Díaz de Santos.

Web:

- Algoritmo, https://es.wikipedia.org/wiki/Algoritmo
- http://en.cppreference.com/w/
- http://www.cplusplus.com/

- Algoritmo voraz, https://es.wikipedia.org/wiki/Algoritmo_voraz

- Recursión (ciencias de computación), https://es.wikipedia.org/wiki/Recursi%C3%B3n_(ciencias_de_computaci%C3%B3n)
- Algoritmo divide y vencerás, https://es.wikipedia.org/wiki/Algoritmo_divide_y_vencer%C3%A1s
- Vuelta atrás, https://es.wikipedia.org/wiki/Vuelta_atr%C3%A1s
- Ramificación y poda, https://es.wikipedia.org/wiki/Ramificaci%C3%B3n_y_poda
- Programación dinámica, https://es.wikipedia.org/wiki/Programaci%C3%B3n_din%C3%A1mica
- Algoritmo probabilista, https://es.wikipedia.org/wiki/Algoritmo_probabilista
- ¿Cuál es el algoritmo MÁS IMPORTANTE de la historia? https://www.youtube.com/watch?v=kflbmvCWdwk
- La destacable historia detrás del algoritmo más importante de todos los tiempos https://www.youtube.com/watch?v=2Xkv-W9tOXU

Software

Programming will be done in C++ and Microsoft Visual Studio 2022(UAB licence) will be used. The following options will be used for the installation of VS2022:

- Microsoft Visual Studio Community 2022
- Desktop development with C++
- C++ MFC latest v142

Students must bring a Windows 10/11 laptop to do problems and practice in class.

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	440	Catalan	first semester	morning-mixed
(PAUL) Classroom practices	441	Catalan	first semester	morning-mixed