UAB
Universitat Autònoma de Barcelona

Analysis and Design of Algorithms

Code: 102783
ECTS Credits: 6

2024/2025

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

Contact

Name: Francisco Javier Sánchez Pujadas

Email: javier.sanchez.pujadas@uab.cat

Teachers

Enrique Ruiz Amakrache

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 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 4. 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 practiceor 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 thereforeapproved 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 ingeneral 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 Campus Virtual and may be subject to possible changes in programming for reasons of adaptation to possible incidents. The Campus Virtual will always be informed about these changes as it is understood that this is the usual platform for exchanging information between teachers and students.

UNIQUE ASSESSMENT

It will be assessed the same as the continuous assessment. But with the following differences:

- Practices and problems will be given on the day of the subject's recovery exam. In this case, the practices do not have the possibility of recovery, since the student cannot correct and see the note as many times as he wants before delivery.
- 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 practical exams will be held on the day of the subject's recovery exam.
- If necessary, the students will be called to take the theory and practical recovery 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.
- http://en.cppreference.com/w/
- http://www.cplusplus.com/

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

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 laptop to do problems and practice in class.

Language list

Name	Group	Language	Semester	Turn

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