DEGREE CURRICULUM

ALGORITHMS AND COMPLEXITY

Coordination: PLANES CID, JORDI

Academic year 2023-24

ALGORITHMS AND COMPLEXITY 2023-24

Subject's general information

<table>
<tr><td>Subject name</td><td>ALGORITHMS AND COMPLEXITY</td></tr>
<tr><td>Code</td><td>102061</td></tr>
<tr><td>Semester</td><td>2nd Q(SEMESTER) CONTINUED EVALUATION</td></tr>
<tr><td>Typology</td><td>
<table>
<tr><th>Degree</th><th>Course</th><th>Character</th><th>Modality</th></tr>
<tr><td>Bachelor's Degree in Computer Engineering</td><td>2</td><td>COMPULSORY</td><td>Attendance-based</td></tr>
<tr><td>Double bachelor's degree: Degree in Computer Engineering and Degree in Business Administration and Management</td><td>2</td><td>COMPULSORY</td><td>Attendance-based</td></tr>
</table>
</td></tr>
<tr><td>Course number of credits (ECTS)</td><td>4.5</td></tr>
<tr><td>Type of activity, credits, and groups</td><td>
<table>
<tr><th>Activity type</th><td>PRALAB</td><td>TEORIA</td></tr>
<tr><th>Number of credits</th><td>3</td><td>1.5</td></tr>
<tr><th>Number of groups</th><td>3</td><td>2</td></tr>
</table>
</td></tr>
<tr><td>Coordination</td><td>PLANES CID, JORDI</td></tr>
<tr><td>Department</td><td>COMPUTER ENGINEERING AND DIGITAL DESIGN</td></tr>
<tr><td>Important information on data processing</td><td>Consult this link for more information.</td></tr>
<tr><td>Language</td><td>Catalan</td></tr>
</table>

ALGORITHMS AND COMPLEXITY 2023-24

Teaching staff	E-mail addresses	Credits taught by teacher	Office and hour of attention
PLANES CID, JORDI	jordi.planes@udl.cat	12	

Subject's extra information

Suggestions

For any questions it is recommended to send an email to teachers of the course.

We advise you to solve the proposed problems and programming exercises, since it allows to reach the learning objectives.

The following courses are highly recommended: Data Structures and Discrete Mathematics.

Learning objectives

- Characterize formally problems. Analyze the efficiency of algorithms using asymptotic notation.
- Identify the nature of the problem and identify the appropriate strategy algorithms.
- Design and implement data up appropriate structures to represent information of each problem.
- Design and implement strategies for efficient algorithms to solve different types of problems.

Competences

Degree Competences

GII-FB3. Capacity to understand and master the basic concepts of discreet mathematics, logical, algorithmic and computational complexity, and its application to solve engineering problems.

GII-CRI6. Knowledge and application of the basic algorithmic procedures of the computer technologies to design problem solving, analysing the suitability and complexity of the algorithms proposed.

GII-CRI7. Knowledge, design and efficient use of the types and data structure more suitable for solving a problem.

GII-CRI8. Capacity to analyse, design, build and keep safety and efficiency in applications, choosing the paradigm and the most suitable programming languages.

Cross-disciplinary Competences

EPS1. Capacity to solve problems and prepare and defence arguments inside the area of studies.

EPS5. Capacity of abstraction and of critical, logical and mathematical thinking.

Subject contents

Organization of the course topics:

1. Preliminaries: algorithm, notation, predicate logic, proof techniques.

2. Formal specification of algorithms based on pre-post conditions.

3. Efficiency of algorithms. Asymptotic notation . Analysis of algorithms.

4. Techniques for transformation of recursive algorithms.

5. Algorithmic schemes: brute-force and greedy search.

6. Algorithmic schemes: divide and conquer.

7. Algorithmic schemes: dynamic programming.

8. Algorithmic schemes: backtracking.

Methodology

The contents of the course are structured in two didactic units. The first aims to study the formal characterization of algorithms. In this sense we will study the technique of formal specification of algorithms based on precondition and postcondition and we will analyze the efficiency of the algorithms by using the asymptotic notation for the study of the temporal cost or time of execution of the algorithms, and the study of techniques for transforming recursive algorithms. The second didactic unit aims to study algorithmic schemes, that is, to analyze, design and apply algorithms capable of solving not only a specific problem, but a family of problems all with the same typification.

The algorithmic schemes we will study are four: divide and conquer, voracious, dynamic programming, and regression. The systematic analysis and design of algorithms based on a specific scheme focuses on the study and development of specific solutions or strategies to solve a problem. The study of each technique and algorithmic scheme will be approached from the resolution of specific problems for each typology. In addition, the algorithmic solutions developed throughout the course will be implemented in the python language (optionally Haskell and Rust). From the point of view of implementation of the algorithms, an empirical study of the execution time will also be carried out for different instances of the treated problems. The empirical study of the execution time of the implementations will demonstrate in a practical way the efficiency of the different algorithmic strategies studied throughout the course.

Development plan

The course is organized in large group classes and laboratory classes. Each week students large group race in 2 hours and 2 hours lab group.

A large group classes are the algorithmic schemes and theoretical basis of the subject. For each technique and formal algorithmic scheme proposes a collection of problems which students must solve. Solving the revised large group classes and laboratory.

In the laboratory classes are tought the most important features of python. In addition to discussing implementation problems and collections solution develops the three practical compulsory works.

The first mandatory practice begin during the 3rd week of the course and will be given to the date fixed for the 1st written test (1st part).

Mandatory practice will begin during the second week of the 10th year and will be delivered to the date set for the 2nd written test (2nd part).

ALGORITHMS AND COMPLEXITY 2023-24

Week	Description	Classroom Activity Big Group	Classroom/independent work
1	Lecture and problems	Lesson 1	3.5h/6h
2	Lecture and problems	Lesson 2	3.5h/6h
3	Lecture and problems	Lesson 3	3.5h/6h
4	Lecture and problems	Lesson 4	3.5h/6h
5	Lecture and problems	Lesson 5	3.5h/6h
6	Lecture and problems	Lesson 6	3.5h/6h
7	Lecture and problems	Lesson 6	3.5h/6h
8	Lecture and problems	Review	3.5h/6h
9	Written tests	**First mid-term exam**	2h/3h
10	Lecture and problems	Lesson 7	3.5h/6h
11	Lecture and problems	Lesson 7	3.5h/6h
12	Lecture and problems	Lesson 8	3.5h/6h
13	Lecture and problems	Lesson 9	3.5h/6h
14	Lecture and problems	Lesson 10	3.5h/6h
15	Lecture and problems	Review	3.5h/6h
16	Written tests	**Second mid-term exam**	2h/3h
17	Written tests	**Second mid-term exam**	
18		Study week	
19	Written tests	**Recovery exam**	

Evaluation

Block	Evaluation Activities	Weighting	Minimum mark	Group	Mandatory	Recuperable
T	Activity Test	15%	-	Yes	No	No
E1	Exam 1	35%	-	No	Yes	Yes
P	Assignment	15%	-	Yes	No	No
E2	Exam 2	35%	-	No	Yes	Yes

The weighting may change in case the exams are not face-to-face.

The evaluation consists of activity test, two examns and one assignment organized as follows:

Activity Test: The knowledge learned in the laboratory will be evaluated before the first partial, in the same place and date of the exam 1. It is not recuperable.

Exam 1: Formalization. Costs. Recursive and iterative design. Schemes transformation of recursive algorithms.

Divide and conquer.

Exam 2: Greedy schemes, backtracking using heuristic optimization.

Assignment : Costs.Recursive and iterative design. Schemes transformation of recursive algorithms. Divide and conquer scheme. Greedy schemes, backtracking using heuristic optimization.

The assignment will only delivery date, will not be handed out this term.

Delivery: Before the date fixed for the second written test.

Validation Assignment 2: In order to define the final mark, will be a written validation date fixed.

Alternative evaluation:

Exam (70% marking): Formalization. Costs. Recursive and iterative design. Schemes transformation of recursive algorithms. Divide and conquer. Greedy schemes, backtracking using heuristic optimization.

Assignment (30% marking): Costs.Recursive and iterative design. Schemes transformation of recursive algorithms. Divide and conquer scheme. Greedy schemes, backtracking using heuristic optimization.

Bibliography

Basic References:

- G. Brassard y P. Bratley. Fundamentos de algoritmia. Prentice Hall. 1997.
- Cormen, T.H.; Leiserson, C.E. ; Rivest, R.L.; Stein, C. Introduction to Algorithms, (3ª edición). MIT Press, 2002. * Skiena, S. The Algorithm Design Manual. Springer 2008.
- S. Dasgupta, C. H. Papadimitriou, U. V. Vazirani. Algorithms. 2006.

Exercices:

- Baynat B., Chrétienne P. Hanen C., Kedad-Sidhoum S., Munier-Kordon A., Picouleau C. Exercices et problèmes d'algoritmique. Ed. Dunod. 3r. ed. 2010.
- R. Guerequeta y A. Vallecillo. Tecnicas de diseño de algoritmos. Servicio de Publicaciones de la Universidad de Málaga. 2nd Ed. 2000. http://www.lcc.uma.es/~av/Libro/indice.html
- Gonzalo, J.; Rodríguez, M. Esquemas algorítmicos: enfoque metodológico y problemas resueltos, UNED, 1997.
- T.Alsinet, A.Corchero, J.Planes. Algorithms and complexity. UdL, 2013.