



Universitat de Lleida

**DEGREE CURRICULUM  
ALGORÍTMICA I COMPLEXITAT**

Academic year 2013-14

## Subject's general information

<b>Subject name</b>	ALGORÍTMICA I COMPLEXITAT
<b>Code</b>	102011
<b>Semester</b>	2n Q Avaluació Continuada
<b>Typology</b>	Obligatòria
<b>ECTS credits</b>	6
<b>Theoretical credits</b>	0
<b>Practical credits</b>	0
<b>Department</b>	Informàtica i Enginyeria Industrial
<b>Important information on data processing</b>	Consult <a href="#">this link</a> for more information.
<b>Distribution of credits</b>	Jordi Planes Cid 4.8 Maria Teresa Alsinet Bernadó 3.6

Jordi Planes Cid  
Maria Teresa Alsinet Bernadó

## Subject's extra information

### Suggestions

For questions or related issues, it is recommended to send an email to the teachers of the subject. We advise you to resolve problems and programming exercises proposed daily. It allows to reach the learning objectives set.

### The course as part of the academic plan

Course taught in the 2nd semester of 2nd year of Degree in Computer Engineering. It corresponds to the Subject "Information" within the module "Basic Training".

## Learning objectives

### See competences

## Competences

### Degree-specific competences

- Knowledge, design and efficient use of the most appropriate data type and structure to resolve a problem.

### Goals

- Formally characterize a problem. Analyze the efficiency of algorithms using asymptotic notation for the study of the temporary or runtime cost of algorithms. Analyze the efficiency of algorithms using asymptotic notation for the study of the algorithm spatial cost. Use formal verification techniques applied to algorithms for iterative and recursive algorithms. Use transformation techniques for recursive algorithms. Use algorithmic optimization techniques.
- Knowledge and application of basic computer technologies' algorithmic procedures to design solutions to problems, by analysing the suitability and complexity of the proposed algorithms.

### Goals

- Without Translate - Dissenyar i implementar estratègies algorísmies eficients per resoldre les diferents tipologies de problemes. Utilitzar les funcionalitats pròpies dels llenguatges de programació per a la implementació de les solucions. Utilitzar un entorn de desenvolupament de programes basat en un llenguatge de programació d'alt nivell. Desenvolupar implementacions eficients.
- Ability to understand and master the basic concepts of discrete mathematics, logic, algorithm and computational complexity, and their application to the resolution of engineering problems.

### Goals

- Without Translate - Dissenyar i implementar estructures de dades adecuades per representar la informació pròpia de cada problema. Dissenyar i implementar de forma eficient les operacions associades amb les estructures de dades identificades. Integrar de forma eficient les estructures de dades i les estratègies algorísmies necessàries per resoldre problemes complexes. Optimitzar l'eficiència de les solucions dissenyades.

- Ability to analyse, design, build and maintain applications in a general, secure and efficient way, by choosing the most appropriate programming paradigm and language.

## Goals

- Without Translate - Identificar la tipologia del problema i identificar l'estrategia algorísmia adequada. Dissenyar i implementar estratègies algorísmies adequades per resoldre les diferents tipologies de problemes. Dissenyar i implementar solucions algorísmies utilitzant la tècnica de divideix i venç. Dissenyar i implementar solucions algorísmies utilitzant la tècnica voraç. Dissenyar i implementar solucions algorísmies utilitzant la tècnica de retrocés. Optimitzar solucions algorísmies basades en la tècnica de retrocés mitjançant el disseny i implementació d'heurístiques de poda de l'espai de cerca. Dissenyar i implementar solucions algorísmies utilitzant la tècnica de programació dinàmica. Analitzar la complexitat espacial i temporal de les estratègies algorísmies adoptades.

## Subject contents

The course contents are structured in four learning units. The first is to characterize the formal study of algorithms. In this sense we will study the formal specification of algorithms based on preconditions and postconditions and analyze the efficiency of algorithms using asymptotic measures to study the run-time of algorithms. The second teaching unit aims to study formal verification techniques for iterative and recursive algorithms and processing techniques for the study of recursive algorithms. The third learning unit is to study algorithmic schemes, i.e., analyze, design and implement algorithms to solve not only a specific problem but a family of problems that share a common set of characteristics. We study four algorithmic schemes: divide and conquer, backtracking and greedy search. The systematic design and analysis of algorithms based on a specific scheme is focused on the study and development of solutions or strategies to solve a problem. A different approach consists in considering globally all algorithms or strategies that may solve a particular problem. This includes all possible algorithms or strategies that have not yet been defined. This approach is considered to be in the field of computational complexity which will be briefly introduced in the last teaching unit. The study of each technique and algorithmic scheme will be tackled, based on solving specific problems for each type. Furthermore, the algorithmic solutions developed throughout the course will be implemented in C++. From the standpoint of the implementation of the algorithms, empirical study of their run-time for different instances of the problems treated will be also considered. The empirical study of the run-time of the algorithm implementations will allow to finely compare the different algorithmic strategies developed during the course, as sometimes the asymptotic run-time behaviour may hinder important variations of the run time of different algorithms.

### 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. Formal verification of iterative and recursive algorithms.
5. Techniques for transformation of recursive algorithms.

6. Algorithmic schemes: divide and conquer.
7. Algorithmic schemes: greedy search.
8. Algorithmic schemes: backtracking.
9. Improving the backtracking scheme by using heuristics.
10. Algorithmic schemes: dynamic programming.
11. Introduction to computational complexity.

## Methodology

Els continguts del curs s'estructuren en quatre unitats didàctiques. La primera té com a objectiu estudiar la caracterització formal d'algorismes. En aquest sentit estudiarem la tècnica d'especificació formal d'algorismes basada en precondició i postcondició i analitzarem l'eficiència dels algorísmes mitjançant l'ús de la notació asimptòtica per a l'estudi del cost temporal o temps d'execució dels algorísmes. La segona unitat didàctica té com a objectiu l'estudi de tècniques de verificació formal d'algorismes aplicades sobre algorismes recursius i iteratius, i l'estudi de tècniques de transformació d'algorismes recursius. La tercera unitat didàctica té com a objectiu l'estudi d'esquemes algorísmics, és a dir, analitzar, dissenyar i aplicar algorismes capaços de resoldre no únicament un problema concret, sino una família de problemes tots amb la mateixa tipificació. Els esquemes algorísmics que estudiarem són quatre: divideix i venç, vorac, retrocès i programació dinàmica. L'anàlisi i disseny sistemàtic d'algorismes a partir d'un esquema concret es centra en l'estudi i desenvolupament de solucions o estratègies concretes per resoldre un problema. Una aproximació diferent consisteix en considerar globalment tots els algorismes o estratègies que poden resoldre un problema concret. Això inclou tots els possibles algorismes o estratègies que encara no s'han definit. Aquesta aproximació és la que es considera en el camp de la complexitat computacional el qual serà breument introduït en la darrera unitat didàctica. L'estudi de cada tècnica i esquema algorísmic l'abordarem a partir de la resolució de problemes concrets per a cada tipologia. A més, les solucions algorísmiques desenvolupades al llarg del curs seran implementades en el llenguatge C++. Des del punt de vista d'implementació dels algorismes, també es realitzarà un estudi empíric del temps d'execució per a diferents instàncies dels problemes tractats. L'estudi empíric del temps d'execució de les implementacions evidenciarà de forma pràctica l'eficiència de les diferents estratègies algorísmiques estudiades al llarg del curs.

## Evaluation

L'avaluació consisteix en dos exàmens i dos pràctiques.

## Bibliography

### Basic References:

# ALGORITHMS AND COMPLEXITY 2013-14

- \* G. Brassard y P. Bratley. Fundamentos de algoritmia. Prentice Hall. 1997.
- \* M. Goodrich y R. Tamassia. Algorithm Design. John Wiley & Sons. 2011.

## **Especification and verification:**

- \* R. Peña Marí. Diseño de programas. Formalismo y abstracción 3 ed. Prentice Hall 2003
- \* J.L. Balcázar. Programación Metódica. McGraw-Hill, 1993.

## **Data structures and algorithms:**

- \* J. Kleinberg and E. Tardos, Algorithm Design, Addison Wesley 2006.
- \* Cormen, T.H.; Leiserson, C.E. ; Rivest, R.L.; Stein, C. Introduction to Algorithms, (3<sup>a</sup> edición). MIT Press, 2002.
- \* M. Weiss. Estructuras de datos y algoritmos. Addison-Wesley Iberoamericana.1995.
- \* R. Kruse. Estructuras de Datos y Diseño de programas. .Prentice Hall. 1998.
- \* Skiena, S. The Algorithm Design Manual. Springer 2008.
- \* PROGRAMACION Y ESTRUCTURAS DE DATOS AVANZADAS, Lourdes Araujo Serna, Raquel Martínez Unanue, Miguel Rodríguez ArtachoEd Ramón Areces, 2011.
- \* ESQUEMAS ALGORÍTMICOS. ENFOQUE METODOLÓGICO Y PROBLEMAS RESUELTOS (1a). Rodríguez Artacho, Miguel ; Gonzalo Arroyo, Julio ; UNED
- \* Técnicas de Diseño de Algoritmos.GUEREQUETA, R. y VALLECILLO, A.:Ed. Universidad de Málaga (1998).

## **Computational complexity:**

- \* J.E. Hopcroft, R. Motwani, J. D. Ullman. Teoría de autómatas,lenguajes y computación. Pearson. Addison Wesley. Tercera Edición, 2007

## **Exercises:**

- \* R. Guerequeta y A. Vallecillo. Técnicas 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.
- \* Martí, N.; Ortega, Y.; Verdejo, J. A. Estructuras de datos y métodos algorítmicos. Prentice Práctica, 2003.
- \* ESTRUCTURAS DE DATOS Y MÉTODOS ALGORÍTMICOS:EJERCICIOS RESUELTOS (1a) Martí Oliet, Narciso ; Ortega Malleón, Yolanda ; Verdejo López, José Alberto ; PEARSON ALHAMBRA

## **Implementation:**

- \* Stroustrup, B. The C++ Programming Language. Addison-Wesley. 3rd edition. 1997.
- \* Meyers, S. Effective C++. 3rd edition. 2005.
- \* Meyers, S. More Effective C++. 1995.
- \* R. Sedgewick. Algoritmos en C++. Addison-Wesley / Diaz de Santos.1995.
- \* M. Weiss. Estructuras de datos en JAVA. Addison Wesley/ Pearson Education. Madrid 2000.