Course guide  
200231 - AIC - Algorithmics and Complexity

**Unit in charge:** School of Mathematics and Statistics  
**Teaching unit:** 723 - CS - Department of Computer Science.  
**Degree:** BACHELOR'S DEGREE IN MATHEMATICS (Syllabus 2009). (Optional subject).  
**Academic year:** 2023  
**ECTS Credits:** 6.0  
**Languages:** English

**LECTURER**

**Coordinating lecturer:** MARIA JOSE SERNA IGLESIAS

**Others:**  
Primer quadrimestre:  
MARIA JOSE SERNA IGLESIAS - M-A

**PRIOR SKILLS**

This is an advanced course in algorithmics and complexity. Students are expected to have prior knowledge, at the second grade level, of algorithmic techniques, programming and mathematical methods, particularly discrete mathematics and probability.

**REQUIREMENTS**

The students are expected to have some knowledge of the basic algorithmic techniques, divide and conquer, greedy, linear programming and dynamic programming. Also they are expected to have a mathematical maturity at the level of second year in the FME.

**DEGREE COMPETENCES TO WHICH THE SUBJECT CONTRIBUTES**

**Specific:**
3. CE-2. Solve problems in Mathematics, through basic calculation skills, taking in account tools availability and the constraints of time and resources.
5. Ability to solve problems from academic, technical, financial and social fields through mathematical methods.
GM-CE1. CE-1. Propose, analyze, validate and interpret simple models of real situations, using the mathematical tools most appropriate to the goals to be achieved.
GM-CE3. CE-3. Have the knowledge of specific programming languages and software.

**Generical:**
GM-CB1. CB-1. Demonstrate knowledge and understanding in Mathematics that is founded upon and extends that typically associated with Bachelor's level, and that provides a basis for originality in developing and applying ideas, often within a research context.
GM-CB2. CB-2. Know how to apply their mathematical knowledge and understanding, and problem solving abilities in new or unfamiliar environments within broader or multidisciplinary contexts related to Mathematics.
8. CG-3. Have the ability to define new mathematical objects in terms of others already know and ability to use these objects in different contexts.
GM-CB3. CB-3. Have the ability to integrate knowledge and handle complexity, and formulate judgements with incomplete or limited information, but that include reflecting on social and ethical responsibilities linked to the application of their knowledge and judgements.
9. CG-4. Translate into mathematical terms problems stated in non-mathematical language, and take advantage of this translation to solve them.
TEACHING METHODOLOGY

Two hours of theory class and two hours of presentation and discussion of problems by the students. Students are expected to dedicate a certain number of hours per week to solving the problems proposed in class.

LEARNING OBJECTIVES OF THE SUBJECT

Provide a solid algorithmic basis to address the resolution of computational problems, both in a future professional job in industry as well as for a doctoral thesis in the field of discrete mathematics or theoretical computer science.

Revise the basic techniques and data structures used to solve algorithmic problems: divide and conquer, voracious, dynamic programming, heaps, hashing, linear programming. Introduce new topics such as computational complexity, random techniques, approximate algorithms and parameterization.

STUDY LOAD

<table>
<thead>
<tr>
<th>Type</th>
<th>Hours</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hours small group</td>
<td>30,0</td>
<td>20.00</td>
</tr>
<tr>
<td>Hours large group</td>
<td>30,0</td>
<td>20.00</td>
</tr>
<tr>
<td>Self study</td>
<td>90,0</td>
<td>60.00</td>
</tr>
</tbody>
</table>

Total learning time: 150 h

CONTENTS

Introduction

Description:
Asymptotic notation, algorithms cost analysis. Review and consolidation of the algorithmic technique.

Specific objectives:
Fractional knapsack, interval scheduling, Huffman codes

Related activities:
Problem solving

Full-or-part-time: 5h
Theory classes: 1h
Practical classes: 1h
Self study: 3h

Computational complexity

Description:

Specific objectives:
Turing Machines, the Halting problem, The word problem. The classes P, NP and EXP. Clique, SAT and variants, Independence set, Vertex cover.

Full-or-part-time: 25h
Theory classes: 5h
Practical classes: 5h
Self study: 15h
### Divide and conquer algorithms

**Description:**
Review and consolidation of the algorithmic technique.

**Specific objectives:**
Quick-select, Matrix product, the FFT, Product of polynomials.

**Related activities:**
Problem solving

**Full-or-part-time:** 10h
- Theory classes: 2h
- Practical classes: 2h
- Self study: 6h

### Dynamic programming

**Description:**
Review and consolidation of the algorithmic technique.

**Specific objectives:**
Sequence alignment, product of n matrices, Knapsack, Edit distance, Text justification, Shortest paths in graphs.

**Full-or-part-time:** 20h
- Theory classes: 4h
- Practical classes: 4h
- Self study: 12h

### Randomized algorithms. Modular arithmetic and primality

**Description:**
Introduction to randomized algorithms. Primality testing and applications.

**Specific objectives:**
Modular arithmetic, MCD, Random generation of prime numbers, Random algorithm for primality testing, Cryptography and RSA.

**Related activities:**
Problem solving.

**Full-or-part-time:** 30h
- Theory classes: 6h
- Practical classes: 6h
- Self study: 18h
**Approximation algorithms**

**Description:** Introduction to basic algorithmic techniques in approximation algorithms. Complexity classes and approximability limits.

**Specific objectives:**
Load balancing, Max cut, Knapsack, Traveling salesman problem. Integer and linear Programming, relax and round technique.

**Related activities:**
Problem solving.

**Full-or-part-time:** 30h
Theory classes: 6h
Practical classes: 6h
Self study : 18h

---

**Parameterization**

**Description:**
Introduction to the basic techniques to design parameterized algorithms. Parameterized complexity.

**Specific objectives:**
Parameters and complexity. Bounded search algorithms and kernelization. Graph parameters, treewidth.

**Related activities:**
Problem solving.

**Full-or-part-time:** 30h
Theory classes: 6h
Practical classes: 6h
Self study : 18h

---

**GRADING SYSTEM**

Midterm exam (P1)
Final exam covering all the course (F), or only the second part (P2).
Problem solving and presentation, participation (C)
E = F o (P1+P2)/2
Course mark: E*0.80+C*0.2

**EXAMINATION RULES.**

During the exams you will not be able to access any support material.

**BIBLIOGRAPHY**

**Basic:**
Complementary:
- Cygan, Marek; Saurabh, Saket; Pilipczuk, Marcin; Pilipczuk, Michal; Marx, Dániel; Lokshtanov, Daniel; Kowalik, Lukasz; Fomin, Fedor V. Parameterized algorithms. New York: Springer, 2015. ISBN 9783319212746.