200231 - AIC - Algorithmics and Complexity

Coordinating unit: 200 - FME - School of Mathematics and Statistics
Teaching unit: 723 - CS - Department of Computer Science
Academic year: 2019
Degree: BACHELOR'S DEGREE IN MATHEMATICS (Syllabus 2009). (Teaching unit Optional)
ECTS credits: 6  Teaching languages: English

Teaching staff

Coordinator: JOSE MARIA DIAZ CORT
Others: Primer quadrimestre:
JOSE MARIA DIAZ CORT - M-A
MARIA JOSE SERNA IGLESIAS - M-A

Prior skills

This is an advanced course in algorithms and complexity for students with a background in programming, basic algorithms, mathematical techniques as calculus, algebra and probability

Requirements

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

They also should be able to program in any standard programming language

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:
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.
9. CG-4. Translate into mathematical terms problems stated in non-mathematical language, and take advantage of this translation to solve them.
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.
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.
200231 - AIC - Algorithmics and Complexity

**Teaching methodology**

2 hours of classroom teaching and 2 hours of homework solving band discussion.

**Learning objectives of the subject**

The course aims to provide a solid background in algorithm and complexity, in preparation for either an industrial job or for graduate level work. The course will go over some known algorithmic techniques but using them to solve some real-life problems. It also will introduce new data structures and algorithmic techniques: randomized techniques, heuristics and study the complexity of problems.

**Study load**

<table>
<thead>
<tr>
<th>Total learning time: 150h</th>
<th>Hours large group:</th>
<th>30h</th>
<th>20.00%</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Hours medium group:</td>
<td>0h</td>
<td>0.00%</td>
</tr>
<tr>
<td></td>
<td>Hours small group:</td>
<td>30h</td>
<td>20.00%</td>
</tr>
<tr>
<td></td>
<td>Guided activities:</td>
<td>0h</td>
<td>0.00%</td>
</tr>
<tr>
<td></td>
<td>Self study:</td>
<td>90h</td>
<td>60.00%</td>
</tr>
</tbody>
</table>
# 200231 - AIC - Algorithmics and Complexity

## Content

<table>
<thead>
<tr>
<th>Topic</th>
<th>Degree competences to which the content contributes:</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Introduction</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Divide and conquer algorithms</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Modular arithmetic and primality</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Dynamic programming</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Complexity</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Heuristic and approximation algorithms</strong></td>
<td></td>
</tr>
</tbody>
</table>

### The nature of computation

<table>
<thead>
<tr>
<th>Learning time: 5h</th>
</tr>
</thead>
<tbody>
<tr>
<td>Theory classes: 2h</td>
</tr>
<tr>
<td>Practical classes: 2h</td>
</tr>
<tr>
<td>Self study: 1h</td>
</tr>
</tbody>
</table>

**Description:**
- Review of basic algorithmic techniques
- Complexity 1 P, NP NPC
- Number theoretical algorithms and RSA
- Hashing, crypto hashing and blockchain
- Complexity 2

**Specific objectives:**
- Give an broad introduction to the field of algorithmic and complexity
Qualification system

- 4 quizzes (Q)
- Final exam (F)
- Homework (problem solving and participation ) (P)

Course score: \( P \times 0.15 + Q \times 0.40 + F \times 0.45 \)

Regulations for carrying out activities

Each one hour quiz in classroom period. The final will be four hour at the end of the academic semester

Bibliography

Basic:
