

30232 - Algorithms for Difficult Problems

Syllabus Information

Academic year: 2023/24

Subject: 30232 - Algorithms for Difficult Problems

Faculty / School: 110 - Escuela de Ingeniería y Arquitectura

Degree: 439 - Bachelor's Degree in Informatics Engineering

ECTS: 6.0

Year: 4

Semester: First semester

Subject type:

Module:

1. General information

In this subject, students will improve their ability to design and develop algorithms by focusing on probabilistic, approximate and heuristic techniques. The student will learn to recognize the problems in whose solution these techniques are used, to design algorithms that use them and to justify their correctness and efficiency.

2. Learning results

The student, in order to pass this subject, must demonstrate the following results...

1. Know the non-exact computational models and is able to select the appropriate one and use it for modeling heterogeneous application domains.
2. Be able to identify NP-difficult problems.
3. Know probabilistic, approximate and heuristic algorithmic techniques for the same.
4. Know how to identify the most relevant components of a problem and select the most appropriate approximate technique for it and to argue in a reasoned way such choice.
5. Know how to compare problems and use this comparison to solve one problem from an efficient solution of another.
6. Know how to reason about the error rate and efficiency of approximate and probabilistic algorithms.
7. Know how to apply amortized efficiency analysis to advanced data structures.
8. Can work in a group, identify group objectives, outline a work plan to achieve them, recognize the different roles within a team, and commit to the tasks assigned.
9. Manage self-learning and development including management and organizational time.
10. Appreciate the need for continuous learning.

3. Syllabus

1. Introduction. NP-hard problems. Optimization problems.
2. Approximate algorithms. Concept. Algorithm design. Guarantees and limits.
3. Probabilistic algorithms: Las Vegas and Monte Carlo. Analysis. Pseudo-random generators.
4. Heuristics. Simulated annealing (simulated annealing).
5. Genetic algorithms.
6. Amortized analysis. Advanced data structures.
7. Specialized algorithms.

4. Academic activities

The program offered to the student to help their achieve the expected results comprises the following activities...

The subject syllabus will be developed in the classes taught in the classroom.

In the problem classes, problems will be solved by applying the concepts and techniques presented in the course syllabus. Problems and exercises will be proposed to be solved before the problem class at where different solutions to these problems will be presented and discussed. Exercises will also be proposed during the session of problems to be solved during the session, some individually and others to be worked in groups.

The practical work is carried out in a computer laboratory or on the personal computers of students. In the practice sessions students will work in teams and perform a series of programming tasks directly related to the topics studied in the subject. For this purpose, a series of works or programming exercises will be proposed for students to solve in groups and deliver them within the deadlines set in each case.

5. Assessment system

The student must demonstrate that they have achieved the intended learning outcomes by means of the following assessment activities

Recommended option (progressive release of final exams):

1. Practical part:

- Laboratory practices (in groups) during the four-month period: 20%.

2. Part of theory and problems:

- Individual work exercises during the four-month period: 20%.
- Intermediate written test: 30%.
- Final exam (take only one part): 30%.

Option based exclusively on final exams:

1. Practical part:

- Practical (individual) programming exam: 20%.

2. Part of theory and problems:

- Final exam (to be completed in its entirety): 80%.