Course guide
270012 - EDA - Data Structures and Algorithmics

Last modified: 30/01/2024

Unit in charge:	Barcelona School of Informatics
Teaching unit:	723 - CS - Department of Computer Science.
Degree:	BACHELOR'S DEGREE IN INFORMATICS ENGINEERING (Syllabus 2010). (Compulsory subject).

Academic year: 2023 **ECTS Credits:** 6.0 **Languages:** Catalan, Spanish, English

LECTURER

Coordinating lecturer: ALBERT OLIVERAS LLUNELL

Others:

Primer quadrimestre:
ILARIO BONACINA - 12, 32, 42
M. LUISA BONET CARBONELL - 13, 21, 22, 23, 51
FERNANDO GASTÓN CODONY - 11, 21, 31
ANTONI LOZANO BOIXADORS - 41, 42, 43
PABLO AGUSTIN MARTIN TORRES - 23, 41, 53
CONRADO MARTÍNEZ PARRA - 11, 12, 13, 33, 43
ALBERT OLIVERAS LLUNELL - 31, 32, 33, 51, 52, 53

Segon quadrimestre:
ILARIO BONACINA - 13, 43
FERNANDO GASTÓN CODONY - 42
CAROLINE LEONORE KÖNIG - 11
SANTIAGO MARCO SOLA - 14
ALBERT OLIVERAS LLUNELL - 11, 12, 13, 14, 41, 42, 43
SALVADOR ROURA FERRET - 12, 41
GABRIEL ALEJANDRO VALIENTE FERUGLIO - 11, 12, 13, 14, 41, 42, 43

PRIOR SKILLS

Students are expected to be familiar with imperative object-based programming techniques:
- parameter passing
- classes,
- objects,
- methods,
- pointers,
- dynamic memory,
- genericity,
- recurrence,
- standard classes usage,
- iterators

They are also expected to know at least one imperative object-oriented language, preferably C++.

Critical thinking capacity and mathematical maturity are required too.

REQUIREMENTS

- Prerequisite PRO1
- Prerequisite PRO2

DEGREE COMPETENCES TO WHICH THE SUBJECT CONTRIBUTES

Specific:

CT2.3. To design, develop, select and evaluate computer applications, systems and services and, at the same time, ensure its reliability, security and quality in function of ethical principles and the current legislation and normative.

CT2.4. To demonstrate knowledge and capacity to apply the needed tools for storage, processing and access to the information system, even if they are web-based systems.

CT4.1. To identify the most adequate algorithmic solutions to solve medium difficulty problems.

CT4.2. To reason about the correction and efficiency of an algorithmic solution.

CT4.3. To demonstrate knowledge and capacity to apply the fundamental principles and the basic techniques of the intelligent systems and its practical application.

CT5.1. To choose, combine and exploit different programming paradigms, at the moment of building software, taking into account criteria like ease of development, efficiency, portability and maintainability.

CT5.2. To know, design and use efficiently the most adequate data types and data structures to solve a problem.

CT5.3. To design, write, test, refine, document and maintain code in an high level programming language to solve programming problems applying algorithmic schemas and using data structures.

CT5.4. To design the programs¿ architecture using techniques of object orientation, modularization and specification and implementation of abstract data types.

CT5.5. To use the tools of a software development environment to create and develop applications.

CT8.6. To demonstrate the comprehension of the importance of the negotiation, effective working habits, leadership and communication skills in all the software development environments.

CT8.7. To control project versions and configurations.

Generical:

G6. SOLVENT USE OF THE INFORMATION RESOURCES: To manage the acquisition, structuring, analysis and visualization of data and information of the field of the informatics engineering, and value in a critical way the results of this management.

TEACHING METHODOLOGY

Topics are explained in a practical way through the use of numerous examples.

Theory classes cover the required concepts and techniques, which are put into practice in the problem-solving and laboratory classes by means of a collection of problems and exercises from an automatic judge.

The two-hour theory class will take place once a week. The two-hour laboratory class will take place once a fortnight. The two-hour problem-solving class will take place once a fortnight.

Programming for the game integrates knowledge and skills of the entire course.

The C++ programming language is used for this course.

LEARNING OBJECTIVES OF THE SUBJECT

1.Understand the definitions of the Big-O, Omega and Theta asymptotic notations and their usefulness in characterising the efficiency of algorithms in time and space.
2.Calculate the efficiency of iterative algorithms using appropriate calculation rules.
3.Describe the efficiency of recursive algorithms using recurrence relations and understand and apply master theorems to solve them.
4.Design algorithms for solving various problems of medium difficulty with time and space constraints.
5.Compare the efficiency of different algorithms for solving the same problem and select the most appropriate one.
6.Understand, explain, design, analyse, compare and implement algorithms (such as mergesort, quicksort, Karatsuba, Strassen, etc.) using divide and conquer.
7.Understand, explain, design, analyse, compare and implement the main data structures that can be used to implement dictionaries (tables, sorted tables, lists, sorted lists, hash tables, binary search trees, AVL trees).
8.Understand, explain, design, analyse, compare and implement the main data structures that can be used to implement priority queues (trees, heaps).
9.Understand, explain, design, analyse, compare and implement algorithms that solve classic graph problems such as traversals, topological sorts, shortest paths, etc.
10.Understand, explain, design, analyse, compare and implement exhaustive search algorithms using the backtracking technique.
11.Identify computational limits: understand the implications of the question "P = NP?", understand the statement of Cook-Levin's Theorem, and recognise and identify several classic NP-complete problems.
12.Complete and modify C++ programming language implementations of several algorithms to solve medium-difficulty problems
13.Identify and propose solutions to efficiency problems in algorithms and programs written in the C++ programming language.
14.Analyse a strategy game for designing and programming an effective, efficient, collaborative and competitive player that maximises the chances of winning the game and is capable of establishing partnerships and coordinating with other players.
15.Apply information search strategies (for bibliographic references, scientific articles, patents, credible web resources, etc.) and, making an ethical use of the compiled information and properly citing sources, produce a well-structured document describing a well-known algorithm that solves a given problem.
16.Compute the cost of an algorithm in the worst, best and average cases.

STUDY LOAD

Type	Hours	Percentage
Hours small group	15,0	10.00
Guided activities	6,0	4.00
Hours medium group	15,0	10.00
Hours large group	30,0	20.00
Self study	84,0	56.00

Total learning time: 150 h

CONTENTS

Analysis of Algorithms

Description:
Cost in time and space. Worst case, best case and average case. Asymptotic notation. Analysis of the cost of iterative and recursive algorithms.

Divide and conquer

Description:
Principles: partition into subproblems, recombination of solutions. Examples: mergesort, quicksort, Karatsuba's algorithm for multiplying large numbers, Strassen's algorithm for matrix multiplication.

Dictionaries

Description:
Operations of dictionaries and ordered dictionaries. Basic implementations: tables and lists. Advanced implementations: hash tables, binary search trees, AVL trees.

Priority Queues

Description:
Operations of priority queues. Implementations with heaps. Heapsort.

Graphs

Description:
Representations: adjacency matrices, adjacency lists and implicit representation. Depth-first search (DFS). Breadth-first search (BFS). Topological sort. Dijkstra's algorithm for shortest paths. Prim's algorithm for minimum spanning trees.

Exhaustive Search and Generation

Description:
Principles: solution space, partial solutions, pruning. Generation of subsets and permutations. Examples: knapsack, travelling salesman.

Notions of Intractability

Description:
Basic introduction to P and NP classes, Cook-Levin's Theorem, reductions and NP-completeness.

ACTIVITIES

Analysis of Algorithms

Description:
Topic 1 development.

Specific objectives:
1, 2, 3, 4, 5, 16

Full-or-part-time: 16h
Theory classes: 4h
Practical classes: 2h
Laboratory classes: 2h
Self study: 8h

Divide and conquer

Description:
Topic 2 development.

Specific objectives:
3, 4, 5, 6, 12, 13

Full-or-part-time: 16h
Theory classes: 4h
Practical classes: 2h
Laboratory classes: 2h
Self study: 8h

Dictionaries

Description:
Topic 3 development.

Specific objectives:
4, 5, 7, 12, 13

Full-or-part-time: 15h 30m
Theory classes: 3h 30m
Practical classes: 2h
Laboratory classes: 2h
Self study: 8h

Priority Queues

Description:
Topic 4 development.

Specific objectives:
4, 5, 8, 12, 13

Full-or-part-time: 8h 30m
Theory classes: 1h 30m
Practical classes: 2h
Laboratory classes: 1h
Self study: 4h

Mid-semester written exam

Description:
Learning objectives corresponding to topics 1 and 2 will be assessed.

Specific objectives:
1, 2, 3, 4, 5, 6, 12

Full-or-part-time: 6h
Guided activities: 2h
Self study: 4h

Graphs

Description:
Topic 5 development.

Specific objectives:
4, 5, 9, 12, 13

Full-or-part-time: 16h
Theory classes: 4h
Practical classes: 2h
Laboratory classes: 2h
Self study: 8h

Game Presentation

Description:
Becoming familiar with the Game: first matches, visualizations and debugging.

Specific objectives:
14

Related competencies :
G6. SOLVENT USE OF THE INFORMATION RESOURCES: To manage the acquisition, structuring, analysis and visualization of data and information of the field of the informatics engineering, and value in a critical way the results of this management.

Full-or-part-time: 2h 30m
Guided activities: 2h
Self study: 0h 30m

Game

Description:
Goals corresponding to learning objective 15 will be assessed.

A statement describing a strategy game will be published. Students will have to program a player for this game (i.e. implement a strategy aimed at winning).

A competition will be carried out in which students will play against each other, from which a ranking will be obtained. To participate in the competition, the players of the students will have to pass a qualification test.

The grade corresponding to this part will be computed from the position in the ranking in a proportional way, ensuring that the winner gets a 10 and that all students with a qualified player get a minimum of 5. Those students who have not been able to qualify a player will get a 0.

Specific objectives:
14

Related competencies :
G6. SOLVENT USE OF THE INFORMATION RESOURCES: To manage the acquisition, structuring, analysis and visualization of data and information of the field of the informatics engineering, and value in a critical way the results of this management.

Full-or-part-time: 8h
Self study: 8h

Game development

Description:
Learning the most appropriate strategies for the Game. Learning the solution to the most common questions about the Game.

Specific objectives:
14

Related competencies :
G6. SOLVENT USE OF THE INFORMATION RESOURCES: To manage the acquisition, structuring, analysis and visualization of data and information of the field of the informatics engineering, and value in a critical way the results of this management.

Full-or-part-time: 4h 30m
Guided activities: 2h
Self study: 2h 30m

Exhaustive Search and Generation

Description:
Topic 6 development.

Specific objectives:
4, 5, 10, 12, 13

Full-or-part-time: 18h
Theory classes: 4h
Practical classes: 2h
Laboratory classes: 4h
Self study: 8h

Notions of Intractability and Undecidibility

Description:
Topic 7 development.

Specific objectives:
11

Full-or-part-time: 11h
Theory classes: 4h
Practical classes: 3h
Self study: 4h

Grand Final of the Game

Description:
Attending the Game Grand Final. Learning the strategies used by the winners.

Specific objectives:
14

Related competencies :
G6. SOLVENT USE OF THE INFORMATION RESOURCES: To manage the acquisition, structuring, analysis and visualization of data and information of the field of the informatics engineering, and value in a critical way the results of this management.

Full-or-part-time: 3h 30m
Guided activities: 2h
Self study: 1h 30m

RGF library science tutorials.

Description:
Self-learning through the BRGF library science tutorials on: intellectual property, the ethical use of information and the use of reference management software.

Specific objectives:
15

Related competencies :
G6. SOLVENT USE OF THE INFORMATION RESOURCES: To manage the acquisition, structuring, analysis and visualization of data and information of the field of the informatics engineering, and value in a critical way the results of this management.

Full-or-part-time: 2h 30m
Self study: 2h 30m

Practical on sound use of computer resources.

Description:
Goals corresponding to learning objective 16 will be assessed.

A statement will be published consisting in the description of a computational problem and the name of an algorithm to solve it. Students will conduct research (in the library, on the web, etc.) into the problem and the algorithm and will write a brief, well-structured document that properly lists sources.

The document should be handed in on the day of the final exam.

Students' generic competences will be assessed on the basis of this document.

Specific objectives:
15

Related competencies :
G6. SOLVENT USE OF THE INFORMATION RESOURCES: To manage the acquisition, structuring, analysis and visualization of data and information of the field of the informatics engineering, and value in a critical way the results of this management.

Full-or-part-time: 3h
Self study: 3h

Computer-based exam

Description:
Laboratory aspects, i.e. implementation aspects, of the topics covered up to the date will be assessed.

Students will be issued two or three problems in front of their computer. These problems will have a statement, one or more public test suites and, possibly, an already implemented code. When students are ready to submit programs for particular problems, they upload them to an automatic judge which returns a verdict on program behaviour. Students can submit up to 10 solutions for the same problem. The lecturer will correct the last solution submitted for each problem.

Specific objectives:
7, 8, 9, 10

Full-or-part-time: 6h
Guided activities: 2h
Self study: 4h

Final Exam

Description:
Learning objectives for content corresponding to topics 1 to 7 will be assessed.

Specific objectives:
1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13

Full-or-part-time: 13h
Guided activities: 3h
Self study: 10h

GRADING SYSTEM

NPP = written mid-semester exam grade (0 to 10)
NO = computer-based exam grade (0 to 10)
NF = final exam grade (0 to 10)
NJ = game grade (0 to 10)

GRADE = mín(10, màx (22.5%NPP + 22.5%NF + 45%NO + 20%NJ , 45%NF + 45%NO + 20%NJ))

BIBLIOGRAPHY

Basic:
- Cormen, T.H. [et al.]. Introduction to algorithms. 4th ed. Cambridge: MIT Press, 2022. ISBN 9780262046305.
- Brassard, G.; Bratley, P. Fundamentos de algoritmia. Prentice Hall, 1997. ISBN 848966000X.
- Weiss, M.A. Data structures and algorithm analysis in C++. 4th ed int. Pearson, 2014. ISBN 0273769383.
- Kreher, D.L.; Stinson, D.R. Combinatorial algorithms: generation, enumeration and search. CRC Press, 1999. ISBN 084933988X.
- Garey, M.R.; Johnson, D.S. Computers and intractability: a guide to the theory of NP-Completeness. W.H. Freeman, 1979. ISBN 0716710447.
- Stroustrup, B. The C++ programming language. 4th ed. Addison-Wesley, 2013. ISBN 9780321563842.
- Neapolitan, R.E. Foundations of algorithms. 5th ed. Jones and Bartlett Learning, 2015. ISBN 9781284049190.

Complementary:
- Manber, U. Introduction to algorithms: a creative approach. Repr. with corr. Addison-Wesley, 1989. ISBN 0201120372.
- Harel, D.; Feldman, Y. Algorithmics: the spirit of computing. 3rd ed. Springer, 2012. ISBN 9783642441356.
- Dasgupta, S.; Papadimitriou, C.; Vazirani, U. Algorithms. Mc Graw Hill Higher Education, 2008. ISBN 9780073523408.

- Sedgewick, R; Wayne, K. Algorithms. 4th ed. Upper Saddle River, NJ: Addison-Wesley, 2011. ISBN 9780321573513.

RESOURCES

Hyperlink:

- http://uva.onlinejudge.org/- http://www.topcoder.com- http://www.cs.pitt.edu/~kirk/algorithmcourses/index.html- http://ocw.mit.edu/courses/#electrical-engineering-and-computer-science- https://www.cs.princeton.edu/courses/archive/fall12/cos226/lectures.php- http://www.cs.sunysb.edu/~algorith/- https://www.jutge.org/