

Course guide

270213 - AP3 - Algorithmics and Programming III

Last modified: 19/07/2023

Unit in charge: Barcelona School of Informatics
Teaching unit: 723 - CS - Department of Computer Science.

Degree: BACHELOR'S DEGREE IN DATA SCIENCE AND ENGINEERING (Syllabus 2017). (Compulsory subject).

Academic year: 2023 **ECTS Credits:** 6.0 **Languages:** Catalan

LECTURER

Coordinating lecturer: ENRIC RODRIGUEZ CARBONELL

Others: Primer quadrimestre:
ALBERT OLIVERAS LLUNELL - 11, 12, 13
ENRIC RODRIGUEZ CARBONELL - 11, 12, 13

PRIOR SKILLS

- Familiarity with the basic techniques of programming and the C++ programming language: iterations, alternatives, recursive functions, parameter passing, pointers, references, dynamic memory, classes, objects, methods, ...
- Knowledge of basic algorithmic concepts: efficiency of algorithms, asymptotic notation, graphs, graph traversal, data structures (lists, search trees, hash, heaps, ...)
- Basic knowledge of discrete mathematics, linear algebra and calculus
- Basic knowledge of probability theory and statistics

DEGREE COMPETENCES TO WHICH THE SUBJECT CONTRIBUTES

Specific:

CE2. To be able to program solutions to engineering problems: Design efficient algorithmic solutions to a given computational problem, implement them in the form of a robust, structured and maintainable program, and check the validity of the solution.

CE7. Demonstrate knowledge and ability to apply the necessary tools for the storage, processing and access to data.

Generical:

CG1. To design computer systems that integrate data of provenances and very diverse forms, create with them mathematical models, reason on these models and act accordingly, learning from experience.

CG2. Choose and apply the most appropriate methods and techniques to a problem defined by data that represents a challenge for its volume, speed, variety or heterogeneity, including computer, mathematical, statistical and signal processing methods.

CG5. To be able to draw on fundamental knowledge and sound work methodologies acquired during the studies to adapt to the new technological scenarios of the future.

Transversal:

CT4. Teamwork. Be able to work as a member of an interdisciplinary team, either as a member or conducting management tasks, with the aim of contributing to develop projects with pragmatism and a sense of responsibility, taking commitments taking into account available resources.

CT5. Solvent use of information resources. Manage the acquisition, structuring, analysis and visualization of data and information in the field of specialty and critically evaluate the results of such management.

CT6. Autonomous Learning. Detect deficiencies in one's own knowledge and overcome them through critical reflection and the choice of the best action to extend this knowledge.

CT7. Third language. Know a third language, preferably English, with an adequate oral and written level and in line with the needs of graduates.

Basic:

CB5. That the students have developed those learning skills necessary to undertake later studies with a high degree of autonomy

TEACHING METHODOLOGY

The syllabus is explained in a practical way, through the presentation of many examples.

Theory lectures introduce all of the required concepts and techniques, which are put into practice in the problem and lab lectures by means of a collection of problems and exercises in an automatic judge.

The two hours of theory classes are taught weekly. The two hours of lab classes are taught every other week. The two hours of problem classes are taught every other week.

The project integrates the contents and competences of all the course.

The course uses the C++ programming language.

LEARNING OBJECTIVES OF THE SUBJECT

1. Be aware of the limits of computation: to understand the implications of the question " $P=NP?$ ", understand the statement of Cook-Levin's Theorem, recognize and identify several classic NP-complete problems.
2. To know, explain, design, analyze, compare and implement exhaustive search algorithms using the backtracking technique.
3. To learn the dynamic programming scheme, identify when it can be applied and how, and be familiar with some fundamental dynamic programming algorithms.
4. To learn the scheme of greedy algorithms, identify when it can be applied and how, learn the most usual techniques for proving their correctness and be familiar with some fundamental greedy algorithms.
5. To complete and modify implementations of several algorithms for solving problems of average difficulty in the C++ programming language.
6. Identify and propose solutions to possible problems of efficiency in programs written in the C++ programming language.
7. To develop projects of average size as a member of a team, learning how to divide a project into smaller parts, to distribute them amongst its members and act with responsibility in a coordinated way for the successful accomplishment of the assigned tasks.
9. To learn algorithms based on local search for solving untractable problems efficiently. To learn a variety of metaheuristics of different nature and to be able to identify when and how they can be applied on concrete computationally hard problems.
10. To learn the foundations of finite automata and regular expressions to be able to use them in practice (search of patterns in texts, etc.)

STUDY LOAD

Type	Hours	Percentage
Hours small group	30,0	20.00
Hours large group	30,0	20.00
Self study	90,0	60.00

Total learning time: 150 h

CONTENTS

Tractability: classes of problems P and NP

Description:

Classes P and NP, Cook-Levin's Theorem, reductions, NP-completeness.



Exhaustive search

Description:

Theoretical foundations: space of solutions, partial solutions, pruning. Examples: subsets, permutations, travelling salesman, subset sum.

Dynamic programming

Description:

Top-down scheme (memoization). Bottom-up scheme (tabulation). Examples: Fibonacci, binomial numbers, knapsack, matrix sequence multiplication.

Greedy algorithms

Description:

Theoretical foundations: general scheme of greedy algorithms. Examples: task scheduling, etc.

Metaheuristics

Description:

Local search. Simulated Annealing, Tabu Search, GRASP, genetic algorithms.

Finite automata and regular expressions

Description:

Alphabets, words, languages. Deterministic finite automata, non-deterministic finite automata, finite automata with lambda-transitions, equivalence between automata models, minimization of automata.

Regular expressions, equivalence with automata. Operations.

ACTIVITIES

Tractability

Specific objectives:

1

Related competencies :

CG5. To be able to draw on fundamental knowledge and sound work methodologies acquired during the studies to adapt to the new technological scenarios of the future.

Full-or-part-time: 18h

Theory classes: 6h

Practical classes: 4h

Self study: 8h

Exhaustive Search

Specific objectives:

2, 5, 6

Related competencies :

CE2. To be able to program solutions to engineering problems: Design efficient algorithmic solutions to a given computational problem, implement them in the form of a robust, structured and maintainable program, and check the validity of the solution.
CT6. Autonomous Learning. Detect deficiencies in one's own knowledge and overcome them through critical reflection and the choice of the best action to extend this knowledge.

CB5. That the students have developed those learning skills necessary to undertake later studies with a high degree of autonomy

Full-or-part-time: 16h

Theory classes: 4h

Laboratory classes: 4h

Self study: 8h

Dynamic Programming

Specific objectives:

3, 5, 6

Related competencies :

CE2. To be able to program solutions to engineering problems: Design efficient algorithmic solutions to a given computational problem, implement them in the form of a robust, structured and maintainable program, and check the validity of the solution.
CT6. Autonomous Learning. Detect deficiencies in one's own knowledge and overcome them through critical reflection and the choice of the best action to extend this knowledge.

Full-or-part-time: 16h

Theory classes: 4h

Laboratory classes: 4h

Self study: 8h

Greedy Algorithms

Specific objectives:

4, 5, 6

Related competencies :

CE2. To be able to program solutions to engineering problems: Design efficient algorithmic solutions to a given computational problem, implement them in the form of a robust, structured and maintainable program, and check the validity of the solution.
CT6. Autonomous Learning. Detect deficiencies in one's own knowledge and overcome them through critical reflection and the choice of the best action to extend this knowledge.

Full-or-part-time: 16h

Theory classes: 4h

Practical classes: 2h

Laboratory classes: 2h

Self study: 8h

Metaheuristics

Specific objectives:

5, 9

Related competencies :

CE2. To be able to program solutions to engineering problems: Design efficient algorithmic solutions to a given computational problem, implement them in the form of a robust, structured and maintainable program, and check the validity of the solution.
CT6. Autonomous Learning. Detect deficiencies in one's own knowledge and overcome them through critical reflection and the choice of the best action to extend this knowledge.

Full-or-part-time: 18h

Theory classes: 4h

Laboratory classes: 6h

Self study: 8h

Finite automata and regular expressions

Specific objectives:

10

Related competencies :

CE7. Demonstrate knowledge and ability to apply the necessary tools for the storage, processing and access to data.

Full-or-part-time: 16h

Theory classes: 4h

Practical classes: 4h

Self study: 8h

Consolidation

Specific objectives:

1, 2, 3, 4, 9, 10

Related competencies :

CG5. To be able to draw on fundamental knowledge and sound work methodologies acquired during the studies to adapt to the new technological scenarios of the future.

CE7. Demonstrate knowledge and ability to apply the necessary tools for the storage, processing and access to data.

CE2. To be able to program solutions to engineering problems: Design efficient algorithmic solutions to a given computational problem, implement them in the form of a robust, structured and maintainable program, and check the validity of the solution.

CB5. That the students have developed those learning skills necessary to undertake later studies with a high degree of autonomy

Full-or-part-time: 12h

Theory classes: 4h

Self study: 8h

Mid term exam

Specific objectives:

1, 2, 3

Related competencies :

CG5. To be able to draw on fundamental knowledge and sound work methodologies acquired during the studies to adapt to the new technological scenarios of the future.

CE2. To be able to program solutions to engineering problems: Design efficient algorithmic solutions to a given computational problem, implement them in the form of a robust, structured and maintainable program, and check the validity of the solution.

CB5. That the students have developed those learning skills necessary to undertake later studies with a high degree of autonomy

Full-or-part-time: 3h

Guided activities: 3h

Project - Exhaustive Search

Specific objectives:

2, 5, 6, 7

Related competencies :

CG1. To design computer systems that integrate data of provenances and very diverse forms, create with them mathematical models, reason on these models and act accordingly, learning from experience.

CG2. Choose and apply the most appropriate methods and techniques to a problem defined by data that represents a challenge for its volume, speed, variety or heterogeneity, including computer, mathematical, statistical and signal processing methods.

CE2. To be able to program solutions to engineering problems: Design efficient algorithmic solutions to a given computational problem, implement them in the form of a robust, structured and maintainable program, and check the validity of the solution.

CT5. Solvent use of information resources. Manage the acquisition, structuring, analysis and visualization of data and information in the field of specialty and critically evaluate the results of such management.

CT6. Autonomous Learning. Detect deficiencies in one's own knowledge and overcome them through critical reflection and the choice of the best action to extend this knowledge.

CT7. Third language. Know a third language, preferably English, with an adequate oral and written level and in line with the needs of graduates.

CT4. Teamwork. Be able to work as a member of an interdisciplinary team, either as a member or conducting management tasks, with the aim of contributing to develop projects with pragmatism and a sense of responsibility, taking commitments taking into account available resources.

CB5. That the students have developed those learning skills necessary to undertake later studies with a high degree of autonomy

Full-or-part-time: 9h

Self study: 9h

Project - Greedy Algorithms

Specific objectives:

4, 5, 6, 7

Related competencies :

CG1. To design computer systems that integrate data of provenances and very diverse forms, create with them mathematical models, reason on these models and act accordingly, learning from experience.

CG2. Choose and apply the most appropriate methods and techniques to a problem defined by data that represents a challenge for its volume, speed, variety or heterogeneity, including computer, mathematical, statistical and signal processing methods.

CE2. To be able to program solutions to engineering problems: Design efficient algorithmic solutions to a given computational problem, implement them in the form of a robust, structured and maintainable program, and check the validity of the solution.

CT5. Solvent use of information resources. Manage the acquisition, structuring, analysis and visualization of data and information in the field of specialty and critically evaluate the results of such management.

CT6. Autonomous Learning. Detect deficiencies in one's own knowledge and overcome them through critical reflection and the choice of the best action to extend this knowledge.

CT7. Third language. Know a third language, preferably English, with an adequate oral and written level and in line with the needs of graduates.

CT4. Teamwork. Be able to work as a member of an interdisciplinary team, either as a member or conducting management tasks, with the aim of contributing to develop projects with pragmatism and a sense of responsibility, taking commitments taking into account available resources.

Full-or-part-time: 9h

Self study: 9h

Lab exam

Specific objectives:

2, 3, 4, 5, 6, 9, 10

Related competencies :

CE7. Demonstrate knowledge and ability to apply the necessary tools for the storage, processing and access to data.

CE2. To be able to program solutions to engineering problems: Design efficient algorithmic solutions to a given computational problem, implement them in the form of a robust, structured and maintainable program, and check the validity of the solution.

CT6. Autonomous Learning. Detect deficiencies in one's own knowledge and overcome them through critical reflection and the choice of the best action to extend this knowledge.

CB5. That the students have developed those learning skills necessary to undertake later studies with a high degree of autonomy

Full-or-part-time: 3h

Guided activities: 3h

Project - Metaheuristics

Specific objectives:

5, 6, 7, 9

Related competencies :

CG1. To design computer systems that integrate data of provenances and very diverse forms, create with them mathematical models, reason on these models and act accordingly, learning from experience.

CG2. Choose and apply the most appropriate methods and techniques to a problem defined by data that represents a challenge for its volume, speed, variety or heterogeneity, including computer, mathematical, statistical and signal processing methods.

CE2. To be able to program solutions to engineering problems: Design efficient algorithmic solutions to a given computational problem, implement them in the form of a robust, structured and maintainable program, and check the validity of the solution.

CT5. Solvent use of information resources. Manage the acquisition, structuring, analysis and visualization of data and information in the field of specialty and critically evaluate the results of such management.

CT6. Autonomous Learning. Detect deficiencies in one's own knowledge and overcome them through critical reflection and the choice of the best action to extend this knowledge.

CT7. Third language. Know a third language, preferably English, with an adequate oral and written level and in line with the needs of graduates.

CT4. Teamwork. Be able to work as a member of an interdisciplinary team, either as a member or conducting management tasks, with the aim of contributing to develop projects with pragmatism and a sense of responsibility, taking commitments taking into account available resources.

Full-or-part-time: 10h

Self study: 10h

Final exam

Specific objectives:

1, 2, 3, 4, 9, 10

Related competencies :

CG5. To be able to draw on fundamental knowledge and sound work methodologies acquired during the studies to adapt to the new technological scenarios of the future.

CE7. Demonstrate knowledge and ability to apply the necessary tools for the storage, processing and access to data.

CE2. To be able to program solutions to engineering problems: Design efficient algorithmic solutions to a given computational problem, implement them in the form of a robust, structured and maintainable program, and check the validity of the solution.

CB5. That the students have developed those learning skills necessary to undertake later studies with a high degree of autonomy

Full-or-part-time: 3h

Guided activities: 3h

GRADING SYSTEM

NPar = grade mid term exam

NFT = grade final theory exam

NFL = grade final lab exam

NPro = grade project

FINAL GRADE = $\max(30\%NPar + 30\%NFT + 20\%NFL + 20\%NPro, 60\%NFT + 20\%NFL + 20\%NPro)$

The grade of the reevaluation exam, if there is any and is higher, replaces the grade of the theory final exam (NFT). The grades of mid term, project and lab (NPar, NPro, NFL) are preserved.



BIBLIOGRAPHY

Basic:

- Cormen, T.H [et al.]. Introduction to algorithms [on line]. 4th ed. Cambridge: MIT Press, 2022 [Consultation: 26/07/2023]. Available on : <https://search-ebscohost-com.recursos.biblioteca.upc.edu/login.aspx?direct=true&AuthType=ip,uid&db=nlebk&AN=2932690&site=ehost-live&ebv=EK&ppid=Page-1>. ISBN 9780262046305.
- Stroustrup, B. The C++ programming language. 4th ed. Upper Saddle River: Addison-Wesley, 2013. ISBN 9780321563842.
- Weiss, M.A.; Chandavarkar, B.R. Data structures and algorithm analysis in C++. 4th ed., int. ed. Pearson, 2014. ISBN 0273769383.
- Garey, M.R.; Johnson, D.S. Computers and intractability: a guide to the theory of NP-Completeness. W.H. Freeman, 1979. ISBN 0716710447.
- Neapolitan, R.E. Foundations of algorithms. 5th ed. Jones and Bartlett Learning, 2015. ISBN 9781284049190.
- Siarry, P. (ed.). Metaheuristics. 5th ed. Springer, 2017. ISBN 9783319832845.
- Cases, R.; Màrquez, L. Llenguatges, gramàtiques i autòmats: curs bàsic [on line]. 2a ed. Barcelona: Edicions UPC, 2003 [Consultation: 26/07/2023]. Available on: <https://upcommons.upc.edu/handle/2099.3/36247>. ISBN 8483017288.
- Oliveras, Albert; Rodríguez Carbonell, Enric. Slides for AP3.

Complementary:

- Gendreau, M.; Potvin, J.-Y. Handbook of metaheuristics [on line]. 3rd ed. New York: Springer, 2018 [Consultation: 26/07/2023]. Available on : <https://ebookcentral-proquest-com.recursos.biblioteca.upc.edu/lib/upcatalunya-ebooks/detail.action?pq-origsite=primo&docID=5521308>. ISBN 9783319910857.
- Dasgupta, S.; Papadimitriou, C.; Vazirani, U.. Algorithms. Boston: Mc Graw Hill Higher Education, 2008. ISBN 9780073523408.
- Sedgewick, R.; Wayne, K. Algorithms. 4th ed. Upper Saddle River, NJ: Addison-Wesley, 2011. ISBN 9780321573513.
- Kleinberg, J.; Tardos, É. Algorithm design. New int. ed. Pearson, 2014. ISBN 9781292023946.
- Hopcroft, J.E.; Motwani, R.; Ullman, J.D. Introduction to automata theory, languages, and computation. 3rd ed. Boston: Pearson/Addison Wesley, 2007. ISBN 0321462254.

RESOURCES

Hyperlink:

- <https://jutge.org/>