Last update: 19-07-2019

270213 - AP3 - Algorithmics and Programming III

Coordinating unit: 270 - FIB - Barcelona School of Informatics

Teaching unit: 723 - CS - Department of Computer Science

Academic year: 2019

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

ECTS credits: 6 Teaching languages: Catalan

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

Basic:

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

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.

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 exercices 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 undertand 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 as member of a team, learning how to divide a project into smaller parts, to distribute them amongst its members and act with responsability 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 patters in texts, etc.)

Study load

Total learning time: 150h	Theory classes:	30h	20.00%
	Laboratory classes:	30h	20.00%
	Guided activities:	0h	0.00%
	Self study:	90h	60.00%

Content

Tractability: classes of problems P and NP

Degree competences to which the content contributes:

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

Exhaustive search

Degree competences to which the content contributes:

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

Dynamic programming

Degree competences to which the content contributes:

Description:
Theretical foundations: optimality, memoization, subproblem. Examples: Fibonacci, binomial numbers, coin exchange, maximum common subsequence, Floyd-Warshall algorithm for shortest paths, travelling salesman, knapsack, edition distance, matrix sequence multiplication.

Greedy algorithms

Degree competences to which the content contributes:

Description:
Theoretical foundations: general scheme of greedy algorithms. Examples: task scheduling, subset sum, Huffman codes, algorithms of Bellman-Ford and Johnson for shortest paths, algorithms of Kruskal and Prim for minimum spanning trees.

Metaheuristics

Degree competences to which the content contributes:

Description:
Constructive procedures. Local search. Metaheuristics: GRASP, Simulated Annealing, Tabu Search, Genetic algorithms, Ant colony, Path Relinking, etc. Applications to computationally complex problems.

Finite automata and regular expresssions

Degree competences to which the content contributes:

270213 - AP3 - Algorithmics and Programming III

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.

270213 - AP3 - Algorithmics and Programming III

Planning of activities

Tractability	Hours: 18h Theory classes: 6h Practical classes: 4h Laboratory classes: 0h Guided activities: 0h Self study: 8h
Specific objectives: 1	

Exhaustive Search	Hours: 16h Theory classes: 4h Practical classes: 0h Laboratory classes: 4h Guided activities: 0h Self study: 8h
Specific objectives: 2, 5, 6	

Dynamic Programming	Hours: 16h Theory classes: 4h Practical classes: 0h Laboratory classes: 4h Guided activities: 0h Self study: 8h
Specific objectives: 3, 5, 6	

Greedy Algorithms	Hours: 16h Theory classes: 4h Practical classes: 2h Laboratory classes: 2h Guided activities: 0h Self study: 8h
Specific objectives: 4, 5, 6	

Metaheuristics	Hours: 18h Theory classes: 4h Practical classes: 0h Laboratory classes: 6h Guided activities: 0h Self study: 8h
Specific objectives: 5, 9	

Finite automata and regular expressions	Hours: 16h Theory classes: 4h Practical classes: 4h Laboratory classes: 0h Guided activities: 0h Self study: 8h
Specific objectives: 10	

Consolidation	Hours: 12h Theory classes: 4h Practical classes: 0h Laboratory classes: 0h Guided activities: 0h Self study: 8h
Specific objectives: 1, 2, 3, 4, 9, 10	

Mid term exam	Hours: 3h Guided activities: 3h Self study: 0h
Specific objectives: 1, 2, 3	

Project - Exhaustive Search	Hours: 9h Guided activities: 0h Self study: 9h
Specific objectives: 2, 5, 6, 7	

Project - Greedy Algorithms	Hours: 9h Guided activities: 0h Self study: 9h
Specific objectives: 4, 5, 6, 7	

Lab exam	Hours: 3h Guided activities: 3h Self study: 0h
Specific objectives: 2, 3, 4, 5, 6, 9, 10	

Project - Metaheuristics	Hours: 10h Guided activities: 0h Self study: 10h
Specific objectives: 5, 6, 7, 9	

Final exam	Hours: 3h Guided activities: 3h Self study: 0h
Specific objectives: 1, 2, 3, 4, 9, 10	

Qualification 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, replaces the grade of the theory final exam (NFT). The grades of mid term, project and lab (NPar, NFL, NPro) are preserved.

270213 - AP3 - Algorithmics and Programming III

Bibliography

Basic:

Cormen, T.H. Introduction to algorithms [on line]. 3rd ed. Cambridge: MIT Press, 2009 [Consultation: 22/07/2019]. Available on: <https://ebookcentral.proquest.com/lib/upcatalunya-ebooks/detail.action?docID=3339142>. ISBN 9780262033848.

Stroustrup, B. The C++ programming language. 4th ed. Upper Saddle River, NJ: Addison-Wesley, 2013. ISBN 9780321563842.

Weiss, M.A.; Chandavarkar, B.R. Data structures and algorithm analysis in C++. 4th ed., int. ed. Boston: Pearson, 2014. ISBN 0273769383.

Garey, M.R.; Johnson, D.S. Computers and intractability: a guide to the theory of NP-Completeness. New York, NY: W.H. Freeman, 1979. ISBN 0716710447.

Neapolitan, R.E. Foundations of algorithms. 5th ed. Burlington, MA: Jones and Bartlett Learning, 2015. ISBN 9781284049190.

Siarry, P. (ed.). Metaheuristics. 5th ed. Cham: Springer, 2017. ISBN 9783319832845.

Cases, R.; Màrquez, L. Llenguatges, gramàtiques i autòmats: curs bàsic [on line]. 2a ed. Barcelona: Edicions UPC, 2003 [Consultation: 22/07/2019]. Available on: <http://hdl.handle.net/2099.3/36247>. ISBN 8483017288.

Complementary:

Gendreau, M.; Potvin, J.-Y. Handbook of metaheuristics. 3rd ed. New York: Springer, 2018. ISBN 9783319910857.

Dasgupta, S.; Papadimitriou, C.; Vazirani, U.. Algorithms. Boston: Mc Graw Hill Higher Education, 2008. ISBN 9780073523408.

Sedgewick, R. Algorithms in C++. 3rd ed. Boston: Addison-Wesley, 1998-2002. ISBN 0201350882.

Kleinberg, J.; Tardos, É. Algorithm design. New int. ed. Boston: 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.

Others resources:

Hyperlink

https://jutge.org/