UNIVERSIDADE DA CORUÑA

Teaching Guide			
Identifying Data			2023/24
Subject (*)	Algorithms	Code	614G01011
Study programme	Grao en Enxeñaría Informática		

Descriptors				
Cycle	Period	Year	Type	Credits
Graduate	1st four-month period	Second	Obligatory	6

Language	Spanish		
Teaching method	Face-to-face		
Prerequisites			
Department	Ciencias da Computación e Tecnoloxías da InformaciónComputación		
Coordinador	Valderruten Vidal, Alberto	E-mail	alberto.valderruten@udc.es
Lecturers	, Aguado Martin, Maria Felicidad Cancela Barizo, Brais Casanova Crespo, Jose Maria Gómez Rodríguez, Carlos Hernandez Pereira, Elena Maria Jorge Castro, Jose Santiago Paris Fernandez, Javier Perez Vega, Gilberto Valderruten Vidal, Alberto Vidal Martin, Concepcion	E-mail	olga.zamaraeva@udc.es felicidad.aguado@udc.es brais.cancela@udc.es jose.casanova.crespo@udc.es carlos.gomez@udc.es elena.hernandez@udc.es santiago.jorge@udc.es javier.paris@udc.es gilberto.pvega@udc.es alberto.valderruten@udc.es concepcion.vidalm@udc.es
Web	moodle.udc.es/course/view.php?id=55374		
General description	This course on Algorithms allows the computer science engineering student to delve into algorithm design techniques, taking into account qualitative and quantitative factors in their evaluation. On the one hand, it completes the training on the writing of efficient and correctly structured programs. On the other hand, it approaches the most common problem-solving techniques that an engineer can find. It is worth noting that the conduction of experiments involving runtime measurements on different algorithms provides an empirical approach that is usually highly regarded by the student, who can thus establish the concrete interpretation of the complexities found. The difficulties that arise in some of the studied cases allow for a complementary reflection on aspects like computing resource management, process execution details, architectures and operating systems used, etc. The study and analysis of an important set of fundamental algorithms is also worth remarking, covering a large range of algorithmic techniques and their applications. The possibility of using different techniques for the resolution of some problems results naturally into thinking about the advantages and disadvantages of the different strategies, and the need to know how to choose the best alternative for each particular scenario. Lastly, it is important to develop the necessary rigor to develop solutions that not only adapt to a given specification, but also do so in an efficient way from the viewpoint of the needed computational resources. This will be illustrated by means of various practical cases where the existence of known efficient algorithms leads us to reject alternative designs, even when they look very natural at a first glance.		

Study programme competences	
Code	Study programme competences
A12	Coñecemento e aplicación dos procedementos algorítmicos básicos das tecnoloxías informáticas para deseñar solucións a problemas, analizando a idoneidade e a complexidade dos algoritmos propostos.
A13	Coñecemento, deseño e utilización de forma eficiente dos tipos e estruturas de datos máis adecuados á resolución dun problema.
B3	Capacidade de análise e síntese
C3	Utilizar as ferramentas básicas das tecnoloxías da información e as comunicacións (TIC) necesarias para o exercicio da súa profesión e para a aprendizaxe ao longo da súa vida.

UNIVERSIDADE DA CORUÑA

Learning outcomes			
Learning outcomes	**Study programme competences**		
To know how to apply techniques for algorithmic complexity analysis.	A12 A13	B3	
To recognize the importance of studying algorithm complexity and to know how to perform empirical studies to determine that complexity.	A12 A13	B3	C3
To know the most used techniques in algorithm design.	A12	B3	
To understand the elements of study about computational complexity.	A12 A13	B3	
To use different computational models and levels of abstraction needed for algorithm design.	A12	B3	
To identify data structures adapted to the studied algorithms to obtain more efficient and robust implementations.	A13	B3	C3

Contents	
Topic	**Sub-topic**
Lesson 1. Analysis of Algorithms. Code: T1. Outline: This first lesson addresses the analysis of algorithm complexity as one of the main goals of the course. The idea is to add algorithmic efficiency to the toolbox of already familiar criteria like program structure and correctness.	Lesson topics: 1. Analysis of the efficiency of algorithms: asymptotic notations, computation model, empirical verification of the analysis. 2. Calculation of runtimes: analysis of worst and average cases, calculation of O, resolution of recurrence relations.
Lesson 2. Data Structures Code: T2. Outline: In this lesson, a revision of basic data structures is proposed (stacks, lists, queues, trees, sets and graphs) to study their usage concerns regarding spatial and temporal complexities. Similarly, a deep study is done over interesting structures regarding execution times: hash tables and heaps. This last structure will be turned to when dealing with an improvement over graph algorithms and in certain dynamic programming cases. The complexity of the searching operation can be used as a leitmotif in this lesson. In the introduction of this lesson, it is important to insist on structure criteria of any application designed, motivating the use of abstract data structures and its implementation by modules. The objective is to establish general outlines of what is considered a programming discipline, which must be required from the student in the practicals.	Lesson topics: 1. Stacks, queues and lists 2. Trees and heaps 3. Hashing 4. Disjoint sets 5. Graphs (representation)

Lesson 3. Algorithms on sequences and sets of data Code: T3. Outline: The problem of sorting a sequence of elements becomes, in this part of the course, an ideal excuse both for studying the complexity of various kinds of algorithms and to present different algorithm design strategies that can be extrapolated to solve other problems. One of the algorithms that merit special attention is quicksort, as it can be used to introduce the fundamental characteristic of random algorithms, which can behave in different ways on the same input. A direct consequence is that the concepts of "best case" or "worst case" for a concrete input no longer makes sense, which is an important aspect to discuss in class.	Lesson topics: 1. Search algorithms 2. Sorting algorithms: insertion, Shell, heapsort, mergesort, quicksort 3. Random algorithms
Lesson 4. Greedy algorithms Code: T4. Outline: In this lesson, greedy algorithms are studied. Once the technique is explained using its general characteristics, presented using an example, the most representative algorithms of this category will be studied: graph algorithms, a solution for the knapsack problem and a planning task problem.	Lesson topics: 1. The knapsack problem 2. Graph algorithms: topological sorting, minimum spanning tree and shortest paths 3. Hashing
Lesson 5. Algorithm design by induction Code: T5. Outline: At this point, the student has already seen various algorithms that follow a divide-and-conquer strategy: mergesort and quicksort, binary search, maximum subsequence sum... the work proposed in the first part of this lesson consist in generalising the formulation of said strategy, identifying its distinct features in each of the proposed algorithms. The second unit of this lesson concerns the use of a bottom-up strategy to find a general solution from the solutions to elementary subproblems. From an efficiency viewpoint, the use of top-down techniques like "divide and conquer" will be questioned in some situations. The option of dynamic programming can yield a compromise allowing, when possible, an optimization of the amount of memory required by the algorithm.	Lesson topics: 1. Divide and conquer 2. Dynamic programming: optimality principle, knapsack problem
Lesson 6. Exploring graphs Code: T6 Outline: The objective of this lesson is to give a broader insight of graph applications to undertake problems of different nature, and to take into account algorithmic techniques linked to the development of relevant areas of computer science as artificial intelligence. The graph algorithms studied in greedy algorithms lesson (T4) agree on visiting all the graph nodes. The improvement of the execution times of those algorithms that avoid the exhaustive visit of the graph nodes will be emphasized.	Lesson topics: 1. Exploring graphs 2. Strategy games 3. Backtracking algorithms

Lesson 7. Computational complexity Code: T7 Outline: In this last lesson, we introduce a reasoning about the set of algorithms that can solve each kind of problem. We will deal with the complexity of problems, lower bounds for problem complexity and NP-completeness. In brief, we will address the main techniques and concepts used in the study of computational complexity.	Lesson topics: 1. NP-Completeness, NP-Complete problems

Planning				
Methodologies / tests	**Competencies**	**Ordinary class hours**	**Student?s personal work hours**	**Total hours**
Guest lecture / keynote speech	A12 A13 B3	28.75	28.75	57.5
Short answer questions	A12 A13 B3	1.25	6.25	7.5
Laboratory practice	A12 A13 B3 C3	19	19	38
Supervised projects	A12 A13 B3 C3	4	2	6
Problem solving	A12 A13 B3	5	10	15
Objective test	A12 A13 B3 C3	4	20	24
Personalized attention		2	0	2
(*)The information in the planning table is for guidance only and does not take into account the heterogeneity of the students.				

Methodologies	
Methodologies	**Description**
Guest lecture / keynote speech	Lectures where theoretical knowledge is taught using various resources: blackboard, slides, projections, demos and virtual resources. They may include guest lectures by invited speakers.
Short answer questions	Tests that consist in solving exercises involving the execution of cases using the algorithms studied in the course, or their adaptation to other situations. These tests are assessed.
Laboratory practice	Practicals designed by the professor, based in the knowledge acquired by the student in the keynote speeches, and which therefore complement them. The students will develop this work in groups of two or three throughout the course, and individually in a final practical that is included in the objective test. The practicals will consist in the implementation of programs that illustrate problems related with the course contents. A report of results will be required for assessment. During the hours assigned to each practical, the reports of the previous practical will be assessed.
Supervised projects	Supervised projects proposed by the professor and developed by the students, either in groups or individually.
Problem solving	Examples will be developed on the theoretical contents of each part of the course, and doubts will be solved. The resolution of some of the problems will be assessed individually.
Objective test	Knowledge of the theoretical and practical contents of the course will be assessed, as well as the final individual practical assignment.

Personalized attention	
Methodologies	**Description**

Supervised projects Laboratory practice Problem solving	Problem-solving lessons in small groups: Examples about theoretical contents related to the lesson will be developed and questions will be answered. Individual or in groups tests for monitoring purposes about the lesson studied. The teacher controls them by SGTs and assessment tests. Computer laboratory practicals: Programs will be implemented to learn problems related to the lesson. A report with results will be asked for assessment. Regarding individual tutoring, it will be maintained during each teacher's office hours through the following channels: - Email, for short answer questions. - Teams: virtual meetings, preferably upon request via email.

Assessment			
Methodologies	**Competencies**	**Description**	**Qualification**
Short answer questions	A12 A13 B3	Two objective tests of monitoring assessment, where the theoretical contents skills of the academic work will be evaluated. They will be made during lectures and will be pre-announced in the initial planning presented in the start of the course.	10
Objective test	A12 A13 B3 C3	Theoretical and operative knowledge of the subject will be evaluated. Individual theory exam: 50% Individual practice exam: 20% To take the first opportunity practice exam, it is mandatory to deliver the laboratory practices in time.	70
Laboratory practice	A12 A13 B3 C3	Four laboratory practicals made in groups of two or three, where it will be assessed: program structure, documentation quality, clarity, appropriateness, and result explanation. To deliver the laboratory practicals in time and form is a necessary condition to take the objective individual practical test for the first opportunity (January). Assessment is done by monitoring practical work, during the laboratory practicals sessions.	10
Problem solving	A12 A13 B3	Evaluation of two or three exercises where, after solving doubts, examples about content skills of the lesson will be developed. These exercises will be carried out in Small Group Tutorial (SGT) hours scheduled along the course. Sometimes, they may be finished in non-teaching hours.	10
Others			

Assessment comments

The individual practical exam (objective test) will take place the same day of the theory exam and different shifts may be established depending on the number of students enrolled; it is mandatory for the student to have in its laptop (or in its user account) all the practical work done in the course.
A student will have a status of ?Absent? if he does not attend the theory and practical exams in the official evaluation period.
Part-time enrollment students:In this subject, this fact involves that the final grade will be the best one between the one obtained following this teaching guide criteria and the one obtained in the objective test with the following division: 70% theory exam and 30% practical exam.
In the 2nd opportunity, the student may attend again the theory and practice exams (parts planned in the objective test). If they do not appear for any of these, the grade obtained in the 1st opportunity will be retained for them.
In the advanced opportunity of December the total grade (100%) corresponds to a specific exam with theoretical and practical issues.
The fraudulent execution of tests or evaluation activities, once verified, will directly result in a failing grade in the whole course year in which it is committed: the student will be graded as "failed" (numeric score 0) in the corresponding academic year, whether the commission of the offense occurs in the first opportunity or the second. For this, their grade will be modified in the record of the first opportunity, if necessary.

Sources of information	
Basic	- M. A. Weiss (1995). Estructuras de Datos y Algoritmos. Addison Wesley - U. Manber (1989). Introduction to Algorithms - A Creative Approach. Addison Wesley - G. Brassard y P. Bratley (1997). Fundamentos de Algoritmia. Prentice Hall
Complementary	- F. Aguado, F. Gago, M. Ladra, G. Pérez, C. Vidal y A. M. Vieites (2018). Problemas resueltos de Combinatoria. Laboratorio con SageMath. Paraninfo - B. W. Kernighan y D. M. Ritchie (1991). El lenguaje de programación C, 2ª edición. Prentice Hall - T. H. Cormen, C. E. Leiserson y R. L. Rivest (1990). Introduction to Algorithms. MIT Press - R. Peña Marí (2005). Diseño de Programas. Formalismo y Abstracción. Tercera edición.. Pearson Prentice Hall - R. Sedgewick (1988). Algorithms. Addison Wesley

Recommendations
Subjects that it is recommended to have taken before
Discrete Mathematics/614G01004 Programming II/614G01006
Subjects that are recommended to be taken simultaneously
Programming Paradigms/614G01014
Subjects that continue the syllabus
Concurrency and Parallelism/614G01018 Intelligent Systems/614G01020
Other comments

UNIVERSIDADE DA CORUÑA

As established in
the relevant regulations, this subject incorporates gender perspective (non-sexist language will be used, the participation of male and female students in class will be encouraged...). We will work to identify sexist prejudices and actitudes and will influence the surroundings to modify them and promote values of respect and equality. Any situations of gender discrimination should be detected, and actions and measures proposed to correct them.