



Teaching Guide				
Identifying Data				2015/16
Subject (*)	Algoritmos	Code	614G01011	
Study programme	Grao en Enxeñaría Informática			
Descriptors				
Cycle	Period	Year	Type	Credits
Graduate	1st four-month period	Second	Obligatoria	6
Language	Spanish			
Teaching method	Face-to-face			
Prerequisites				
Department	Computación			
Coordinador	Valderruten Vidal, Alberto	E-mail	alberto.valderruten@udc.es	
Lecturers	Blanco Ferro, Antonio angel Fontenla Romero, Oscar Gómez Rodríguez, Carlos Hernandez Pereira, Elena Maria Jorge Castro, Jose Santiago Valderruten Vidal, Alberto	E-mail	antonio.blanco.ferro@udc.es oscar.fontenla@udc.es carlos.gomez@udc.es elena.hernandez@udc.es santiago.jorge@udc.es alberto.valderruten@udc.es	
Web	www.madsgroup.org/docencia/alg			
General description	<p>A materia de Algoritmos permite ao estudante de Enxeñaría informática profundizar nas técnicas de deseño dos algoritmos tendo en conta factores cualitativos e cuantitativos na avaliación dos mesmos. Dunha parte completa a formación na elaboración de programas eficientes e correctamente estruturados, e doutra parte permite abordar as técnicas de deseño máis utilizadas na resolución dos problemas que pode encontrar o enxeñeiro.</p> <p>É de destacar que a realización de experimentos de medición de tempos de execución sobre os distintos algoritmos analizados achega un enfoque empírico que adoita ser moi valorado polo estudante, que pode así constatar a interpretación concreta das complexidades atopadas. As dificultades plantexadas por algúns casos estudados permiten unha reflexión complementaria sobre aspectos como a xestión de recursos informáticos, detalles de execución de procesos, arquitecturas e sistemas operativos utilizados, etc.</p> <p>Tamén é destacable o estudo e análise dun conxunto importante de algoritmos fundamentais, cubrindo un amplo espectro de técnicas algorítmicas e de as súas aplicacións. A posibilidade de usar distintas técnicas na resolución dalgúns problemas leva naturalmente a pensar nas vantaxes e inconvintes das distintas estratexias, e na necesidade de saber elixir a que mellor se adapta a cada situación.</p> <p>Por último é importante profundizar no rigor necesario para desenvolver non só solucións que se adapten a unhas especificacións dadas, sino ademais que o fagan de modo eficiente dende o punto de vista dos recursos informáticos necesarios. Resulta fundamental a ilustración mediante varios casos prácticos nos que a existencia de algoritmos eficientes coñecidos leva a desechar os deseños alternativos por moi naturais que podan resultar a primeira vista.</p>			

Study programme competences / results	
Code	Study programme competences / results
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.

Learning outcomes	
Learning outcomes	Study programme competences / results



Recoñecer a importancia do estudo da complexidade dos algoritmos e saber realizar estudos empíricos para determinala.	A12 A13	B3	C3
Saber aplicar as técnicas de análise da complexidade dos algoritmos.	A12 A13	B3	
Identificar estruturas de datos adaptadas aos algoritmos estudados para obter implementacións máis eficientes e robustas.	A13	B3	C3
Coñecer as técnicas máis utilizadas no deseño dos algoritmos.	A12	B3	
Utilizar diferentes modelos de computación e niveis de abstracción necesarios para o deseño de algoritmos.	A12	B3	
Comprender elementos de estudo sobre a complexidade computacional.	A12 A13	B3	

Contents	
Topic	Sub-topic
<p><b>Tema 1</b></p> <p>Título do tema: Análise de Algoritmos.</p> <p>Código: T1</p> <p>Presentación: Neste primeiro tema plantexase a análise da complexidade dos algoritmos como uno dos principais obxectivos do curso.</p> <p>En síntese, tratase de engadir aos criterios que xa deben resultar familiares, os de estruturación e de corrección dos programas, o da eficiencia dos algoritmos.</p>	<p>Unidades de contido:</p> <ol style="list-style-type: none"> <li>1. Análise da eficiencia dos algoritmos: Notacións asintóticas, Modelo de computación, Verificación empírica da análise.</li> <li>2. Cálculo dos tempos de execución: Análise dos casos peor e medio, Cálculo da O, Resolución de recurrencias.</li> </ol>
<p><b>Tema 2</b></p> <p>Título do tema: Estruturas de datos.</p> <p>Código: T2</p> <p>Presentación: Neste segundo tema propónse unha revisión das estruturas de datos básicas (pilas, listas, colas, árbores, conxuntos e grafos) co obxectivo de estudar todas as implicacións do seu uso en canto ás complexidades espacial e temporal.</p> <p>Igualmente profundarase no estudo de estruturas interesantes desde o punto de vista do tempo de execución: as taboas de dispersión e os montículos, estrutura ésta última á que recurriremos máis adiante cando se trate de implementar melloras en algoritmos de grafos e nalgún caso de programación dinámica. A complexidade da operación de procura pode servir como fio condutor en boa parte deste tema.</p> <p>Convén nuna introdución desta parte do curso insistir nos criterios de estruturación que debemos manter no deseño de calquer aplicación, motivando o uso de tipos de datos abstractos e a súa implementación modular. O obxectivo é dar así as liñas xerais da que se considera a disciplina de programación que debe esixirse ao estudante para a realización das prácticas.</p>	<p>Unidades de contido:</p> <ol style="list-style-type: none"> <li>1. Pilas, colas, listas.</li> <li>2. Árbore, montículos.</li> <li>3. Dispersión (hashing).</li> <li>4. Conxuntos disxuntos.</li> <li>5. Grafos (representación).</li> </ol>



<p>Tema 3</p> <p>Título do tema: Algoritmos sobre secuencias e conxuntos de datos</p> <p>Código: T3</p> <p>Presentación: O problema da ordenación dunha secuencia de elementos convértese nesta parte do curso nunha escusa ideal tanto para estudar a complexidade de varios tipos de algoritmos como para presentar diferentes estratexias de deseño de algoritmos que se poden extrapolar para a resolución doutros problemas.</p> <p>Un dos algoritmos que merecen especial atención é á ordenación rápida, dado que permite introducir a característica fundamental dos algoritmos aleatorios, que compórtanse de forma distinta na execución dunha mesma entrada. Unha consecuencia directa é que o califcativo de peor caso ou mellor caso para una entrada deixa de ter sentido, aspecto que é importante debatir en clase.</p>	<p>Unidades de contido:</p> <ol style="list-style-type: none"><li>1. Algoritmos de procura.</li><li>2. Algoritmos de ordenación: Inserción, Shell, Montículos (heapsort), Fusión (mergesort), Ordenación Rápida (quicksort).</li><li>3. Algoritmos aleatorios.</li></ol>
<p>Tema 4</p> <p>Título do tema: Algoritmos voraces</p> <p>Código: T4</p> <p>Presentación: Neste tema estúdanse algoritmos ávidos ou voraces. Unha vez explicada a técnica a través das súas características xerais que presentaremos coa axuda dalgún exemplo, estudaranse os algoritmos máis representativos desta categoría: os algoritmos de grafos, unha solución ao problema da mochila e algún problema de planificación de tarefas.</p>	<p>Unidades de contido:</p> <ol style="list-style-type: none"><li>1. Problema da mochila.</li><li>2. Algoritmos de grafos: Ordenación Topolóxica, Árbore de recubrimiento mínimo, Caminos mínimos.</li><li>3. Problemas de planificación de sistemas informáticos.</li></ol>
<p>Tema 5</p> <p>Título do tema: Deseño de algoritmos por indución</p> <p>Código: T5</p> <p>Presentación: Neste punto, xa se viron ao longo do curso varios algoritmos que seguen a estratexia divide e vencerás: ordenación por fusión e ordenación rápida, procura dicotómica, suma da subsecuencia máxima. . . O traballo proposto na primeira unidade deste tema consiste basicamente en xeneralizar as formulacións da dita estratexia identificando as súas distintas características en cada un dos algoritmos propostos.</p> <p>Na segunda unidade do tema expónse o uso dunha estratexia ascendente mediante a procura dunha solución xeral a partir das solucións de subproblemas elementais. Desde o punto de vista da eficiencia cuestionarase o uso de técnicas descendentes como divide e vencerás en determinadas situacións. Mentres que coa opción da programación dinámica procurarase un compromiso que permita, cando sexa posible, unha optimización da cantidade de memoria requirida polo algoritmo.</p>	<p>Unidades de contido:</p> <ol style="list-style-type: none"><li>1. Divide e Vencerás.</li><li>2. Programación dinámica: Principio de optimalidade, Problema da mochila.</li></ol>



<p><b>Tema 6</b>  <b>Título do tema:</b> Exploración de grafos.  <b>Código:</b> T6  <b>Presentación:</b> O obxectivo deste tema é o de dar unha visión máis ampla das aplicacións dos grafos no tratamento de problemas de diversa índole, así como a de non deixar de lado as técnicas algorítmicas ligadas ao desenvolvemento de importantes áreas da computación como a intelixencia artificial.  Os algoritmos de grafos vistos no tema de algoritmos voraces (T4) coinciden en realizar un percorrido de todos os nodos do grafo. Insistirase entón en cómo mellorar os tempos de execución dos algoritmos que se presenten evitando un análise exhaustiva de todos os nodos.</p>	<p><b>Unidades de contido:</b></p> <ol style="list-style-type: none"> <li>1. Xogos de estratexia.</li> <li>2. Percorridos.</li> <li>3. Algoritmos con retroceso.</li> </ol>
<p><b>Tema 7</b>  <b>Título do tema:</b> Complexidade Computacional.  <b>Código:</b> T7  <b>Presentación:</b> Neste último tema plantexamos un razoamento sobre o conxunto dos algoritmos capaces de resolver cada tipo de problema. Falaremos das complexidades dos problemas, de cotas inferiores para a complexidade dos problemas e de NP-compleción, en definitiva, das principais técnicas e conceptos que se utilizan no estudo da complexidade computacional.</p>	<p><b>Unidades de contido:</b></p> <ol style="list-style-type: none"> <li>1. NP-Compleitud, Problemas NP-completos</li> </ol>

Planning				
Methodologies / tests	Competencies / Results	Teaching hours (in-person & virtual)	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	Clases maxistras na exposición dos coñecementos teóricos utilizando diferentes recursos: o encerado, transparencias, proxeccións, demostracións e a facultade virtual. Pode incluír conferencia invitada.
Short answer questions	En xeral consiste na realización de exercicios sobre a execución de casos sobre os algoritmos estudados ou sobre a súa adaptación a outras situacións. Estas probas son avaliadas.
Laboratory practice	Prácticas deseñadas polo profesor basadas nos coñecementos que o estudante vai adquirindo nas clases maxistras e que polo tanto as complementan. Os estudantes desenvolverán estes traballos en parellas ao longo do curso, e individualmente nunha práctica final que se inclúe na proba obxectiva. Implementaranse programas que ilustren os problemas relacionados co el tema. Esixirase o informe de resultados para a súa avaliación. Ao longo das horas asignadas a cada práctica evaluaranse os informes da práctica anterior.
Supervised projects	Traballos tutelados propostos polo profesor e desenvolto polos estudantes ou ben en grupo ou ben individualmente.



Problem solving	Desenvolveranse exemplos sobre os coñecementos teóricos relacionados co tema, e resolveranse dúbidas. Avaliarase individualmente a resolución dalgúns problemas.
Objective test	Avaliarase o dominio dos coñecementos teóricos e operativos da materia. Avaliarase igualmente a práctica individual final.

### Personalized attention

Methodologies	Description
Problem solving Laboratory practice Supervised projects	Clases de problemas en grupos reducidos: desenvolveranse exemplos sobre os coñecementos teóricos relacionados co tema, e resolveranse dúbidas. Traballos tutelados ben individuais ben en grupo sobre algún aspecto do tema a estudar. Son controlados por parte do profesor mediante tutorías en grupo e controles de avaliación. Prácticas de aula de informática: implementaranse programas que ilustren os problemas relacionados co tema. Pedirase o informe de resultados para a súa avaliación.

### Assessment

Methodologies	Competencies / Results	Description	Qualification
Problem solving	A12 A13 B3	Avaliación de 3 traballos nos que desenvolveranse exemplos sobre os coñecementos teóricos relacionados co tema trala resolución de dúbidas.  Realizaranse durante as horas de Traballo en Grupo Reducido (TGR) planificadas ao longo do curso, podendo nalgún caso completarse en horas no presenciais.	10
Objective test	A12 A13 B3 C3	Avaliarase o dominio dos coñecementos teóricos e operativos da materia.  Exame individual de teoría (2h): 50% Exame individual de prácticas (2h): 20%  Para ser convocada/o ao exame de prácticas da 1ª oportunidade, é necesaria a entrega en prazo de todas as prácticas de laboratorio.	70
Laboratory practice	A12 A13 B3 C3	4 prácticas realizadas en parellas, nas que se avaliará: estruturación dos programas, calidade da documentación, claridade, adecuación e explicación dos resultados.  A entrega en tempo e forma das prácticas é condición necesaria para acceder ao exame individual final de prácticas da 1ª oportunidade.  A avaliación realízase mediante tutorías de seguimento das prácticas, durante as horas de práctica.	10
Short answer questions	A12 A13 B3	3 probas escritas de avaliación continua, nas que se avaliará o dominio dos contidos dos traballos académicos a revisar.  Realizaranse durante as horas de teoría e anunciaranse con polo menos 5 días de antelación.	10
Others			

### Assessment comments



Na 2ª oportunidade o estudante poderá presentarse novamente tanto ao exame de prácticas como ao exame teórico (partes previstas na proba obxectiva).

O exame individual de prácticas (proba obxectiva) terá lugar no mesmo día fixado para o exame da materia e poderán establecerse distintos turnos dependendo do número de estudantes matriculados; é necesario dispor, previamente, do material de todas as prácticas realizadas no longo do curso na

conta de usuario de cada estudante.

Presentarse ao exame de prácticas ou ao exame de teoría suporá unha calificación distinta ao "non presentado" na acta correspondente.

Según o previsto na Norma que regula o réxime de dedicación ao estudo dos estudantes de Grao da UDC, os estudantes poderán optar por unha dedicación a tempo parcial. A súa implantación no ámbito desta materia suporá que, de forma xeral, a calificación que figurará na acta será a mellor entre a obtida según o criterio especificado nesta sección da guía docente, e a obtida unicamente coa proba obxectiva ponderando nun 70% o exame de teoría e nun 30% o exame de prácticas.

Na oportunidade adiantada de decembro o 100% da avaliación corresponderá a un exame específico de teoría que incluírá cuestións relacionadas coas prácticas.

### Sources of information

<b>Basic</b>	<ul style="list-style-type: none"> <li>- G. Brassard y P. Bratley (1997). Fundamentos de Algoritmia. Prentice Hall</li> <li>- U. Manber (1989). Introduction to Algorithms - A Creative Approach. Addison Wesley</li> <li>- M. A. Weiss (1995). Estructuras de Datos y Algoritmos. Addison Wesley</li> </ul>
<b>Complementary</b>	<ul style="list-style-type: none"> <li>- R. Sedgewick (1988). Algorithms. Addison Wesley</li> <li>- R. Peña Marí (2005). Diseño de Programas. Formalismo y Abstracción. Tercera edición.. Pearson Prentice Hall</li> <li>- T. H. Cormen, C. E. Leiserson y R. L. Rivest (1990). Introduction to Algorithms. MIT Press</li> <li>- B. W. Kernighan y D. M. Ritchie (1991). El lenguaje de programación C, 2ª edición. Prentice Hall</li> </ul>

### Recommendations

#### Subjects that it is recommended to have taken before

Discrete Mathematics/614G01004

Programming II/614G01006

#### Subjects that are recommended to be taken simultaneously

#### Subjects that continue the syllabus

Concorrencia e Paralelismo/614G01018

Sistemas Intelixentes/614G01020

#### Other comments

(\*)The teaching guide is the document in which the URV publishes the information about all its courses. It is a public document and cannot be modified. Only in exceptional cases can it be revised by the competent agent or duly revised so that it is in line with current legislation.