Universitat de Lleida

GUÍA DOCENTE

ALGORÍTMICA Y COMPLEJIDAD

Coordinación: PLANES CID, JORDI

Año académico 2019-20

ALGORÍTMICA Y COMPLEJIDAD 2019-20

Información general de la asignatura

Denominación	ALGORÍTMICA Y COMPLEJIDAD
Código	102011
Semestre de impartición	2o Q(SEMESTRE) EVALUACIÓN CONTINUADA
Carácter	

Grado/Máster	Curso	Carácter	Modalidad
Grado en Ingeniería Informática	2	OBLIGATORIA	Presencial

Número de créditos de la asignatura (ECTS)	6
Tipo de actividad, créditos y grupos	

Tipo de actividad	PRALAB	TEORIA
Número de créditos	3	1.5
Número de grupos	3	1

Coordinación	PLANES CID, JORDI
Departamento/s	INFORMATICA E INGENIERIA INDUSTRIAL
Distribución carga docente entre la clase presencial y el trabajo autónomo del estudiante	60 horas de clase presencial, 90 horas de trabajo autónomo
Información importante sobre tratamiento de datos	Consulte este enlace para obtener más información.
Idioma/es de impartición	Catalán

ALGORÍTMICA Y COMPLEJIDAD 2019-20

Profesor/a (es/as)	Dirección electrónica\nprofesor/a (es/as)	Créditos impartidos por el profesorado	Horario de tutoría/lugar
PLANES CID, JORDI	jordi.planes@udl.cat	10,5	

Información complementaria de la asignatura

Recomendaciones

Para cualquier duda y/o cuestión se recomienda enviar un correo electrónico al profesorado de la asignatura.

Resolver los problemas y ejercicios de programación que se proponen diariamente permite alcanzar los objetivos de aprendizaje establecidos.

Recomendaciones: Conocimientos de Estructuras de datos y Matemática discreta.

Objetivos académicos de la asignatura

- Caracterizar formalmente los problemas. Analizar la eficiencia de los algoritmos mediante el uso de la notación asintótica .
- Identificar la tipología del problema e identificar la estrategia algoritmia adecuada.
- Diseñar e implementar estructuras de datos adecuadas para representar la información propia de cada problema.
- Diseñar e implementar estrategias algoritmos eficientes para resolver las diferentes tipologías de problemas.

Competencias

Competencias específicas de la titulación

GII-FB3. Capacidad para comprender y dominar los conceptos básicos de matemática discreta, lógica, algorítmica y complejidad computacional, y su aplicación para la resolución de problemas propios de la ingeniería.

GII-CRI6. Conocimiento y aplicación de los procedimientos algorítmicos básicos de las tecnologías informáticas para diseñar soluciones a problemas, analizando la idoneidad y complejidad de los algoritmos propuestos.

GII-CRI7. Conocimiento, diseño y utilización de forma eficiente los tipos y estructuras de datos más adecuados a la resolución de un problema.

GII-CRI8. Capacidad para analizar, diseñar, construir y mantener aplicaciones de forma robusta, segura y eficiente, eligiendo el paradigma y los lenguajes de programación más adecuados.

Competencias transversales de la titulación

EPS1. Capacidad de resolución de problemas y elaboración y defensa de argumentos dentro de su área de estudios.

EPS5. Capacidad para la abstracción y el razonamiento crítico, lógico y matemático.

Contenidos fundamentales de la asignatura

ALGORÍTMICA Y COMPLEJIDAD 2019-20

Organización del curso por temas:

1. Preliminares: algoritmo, notación, lógica de predicados, técnicas de demostración.

2. Especificación formal de algoritmos basada en pre-post condiciones.

3. Eficiencia de los algoritmos. Notación asintótica. Análisis de algoritmos.

4. Verificación formal de algoritmos recursivos e iterativos.

5. Técnicas de transformación de algoritmos recursivos.

6. Esquemas algorítmicos: divide y vencerás.

7. Esquemas algorítmicos: voraz.

9. Esquemas algorítmicos: programación dinámica.

9. Esquemas algorítmicos: búsqueda con retroceso.

11. Introducción a la complejidad computacional.

Ejes metodológicos de la asignatura

Los contenidos del curso se estructuran en cuatro unidades didácticas. La primera tiene como objetivo estudiar la caracterización formal de algoritmos. En este sentido estudiaremos la técnica de especificación formal de algoritmos basada en precondición y postcondición y analizaremos la eficiencia de los algoritmos mediante el uso de la notación asintótica para el estudio del coste temporal o tiempo de ejecución de los algoritmos. La segunda unidad didáctica tiene como objetivo el estudio de técnicas de verificación formal de algoritmos aplicadas sobre algoritmos recursivos e iterativos, y el estudio de técnicas de transformación de algoritmos recursivos. La tercera unidad didáctica tiene como objetivo el estudio de esquemas algorítmicos, es decir, analizar, diseñar y aplicar algoritmos capaces de resolver no únicamente un problema concreto, sino una familia de problemas todos con la misma tipificación. Los esquemas algorítmicos que estudiaremos son tres: divide y vencerás, voraz, y backtracking. El análisis y diseño sistemático de algoritmos a partir de un esquema concreto se centra en el estudio y desarrollo de soluciones o estrategias concretas para resolver un problema. Una aproximación diferente consiste en considerar globalmente todos los algoritmos o estrategias que pueden resolver un problema concreto. Esto incluye todos los posibles algoritmos o estrategias que aún no se han definido. Esta aproximación es la que se considera en el campo de la complejidad computacional, que será brevemente introducido en la última unidad didáctica. El estudio de cada técnica y esquema algorítmico lo abordaremos a partir de la resolución de problemas concretos para cada tipología. Además, las soluciones algorítmicas desarrolladas a lo largo del curso serán implementadas en el lenguaje Ocaml y C++. Desde el punto de vista de implementación de los algoritmos, también se realizará un estudio empírico del tiempo de ejecución para diferentes instancias de los problemas tratados. El estudio empírico del tiempo de ejecución de las implementaciones evidenciará de forma práctica la eficiencia de las diferentes estrategias algorítmicas estudiadas a lo largo del curso.

Plan de desarrollo de la asignatura

La asignatura se organiza en clases de grupo grande y clases de laboratorio. Cada semana el estudiante cursa 2 horas en grupo grande y 2 horas en grupo de laboratorio.

En las clases de grupo grande se presentan los esquemas algorítmicos y los fundamentos teóricos de la asignatura. Para cada esquema algorítmico y técnica formal se propone una colección de problemas los que debe resolver el estudiante. La solución de los problemas se revisa en las clases de grupo grande y de laboratorio.

En las clases de laboratorio se presentan las características más importantes del lenguaje python. Además, se aborda la implementación de las colecciones de problemas y se desarrolla la solución a las tres prácticas obligatorias de la asignatura.

La primera práctica obligatoria se iniciará durante la 3ª semana de curso y se entregará a la fecha fijada para la 1ª prueba escrita (1º parcial).

La segunda práctica obligatoria se iniciará durante la 10ª semana de curso y se entregará a la fecha fijada para la 2ª prueba escrita (2º parcial).

Semana	Descripción	Actividad Presencial Grupo Grande	Trabajo presencial/autónomo
1	Clase magistral i problemes	Tema 1	4h/6h
2	Clase magistral i problemes	Tema 2	4h/6h
3	Clase magistral i problemes	Tema 3	4h/6h
4	Clase magistral i problemes	Tema 4	4h/6h
5	Clase magistral i problemes	Tema 5	4h/6h
6	Clase magistral i problemes	Tema 6	4h/6h
7	Clase magistral i problemes	Tema 6	4h/6h
8	Clase magistral i problemes	Repaso	4h/6h
9	Prueba escrita	**Primer parcial**	2h/3h
10	Clase magistral i problemes	Tema 7	4h/6h
11	Clase magistral i problemes	Tema 7	4h/6h
12	Clase magistral i problemes	Tema 8	4h/6h
13	Clase magistral i problemes	Tema 9	4h/6h
14	Clase magistral i problemes	Tema 10	4h/6h
15	Clase magistral i problemes	Repaso	4h/6h
16	Prueba escrita	**Segundo parcial**	2h/3h
17	Prueba escrita	**Segundo parcial**	
18		Semana de estudio	
19	Prueba escrita	**Recuperación**	

Sistema de evaluación

Acrónimo	Actividades de evaluación	Ponderación	Nota mínima	En grupo	Obligatoria	Recuperable
T1	Examen 1º Parcial	10%	-	No	Sí	Sí
A	Test Activitades	20%	-	No	Sí	No
P	Práctica	30%	-	Sí	Sí	Sí
T2	Examen 2º Parcial	10%	-	No	Sí	Sí

E	Trabajos extra	30%	-	No	Sí	Sí

La evaluación consiste en dos exámenes y tres prácticas organizadas de la siguiente forma:

Prueba escrita 1: Formalización. Costes. Diseño recursivo e iterativo. Esquemas de transformación de algoritmos recursivos. Divide y vencerás.
Porcentaje (no se pide nota mínima)

Práctica obligatoria 1: Diseño recursivo e iterativo. Esquemas de transformación de algoritmos recursivos. Divide y vencerás.

La práctica tendrá una única fecha de entrega, no se podrá entregar fuera de este plazo, y no se podrá recuperar.
Porcentaje (no se pide nota mínima)

Entrega: Antes de la fecha fijada para la prueba escrita 1.

Validación de la práctica 1: Para definir la nota final de las prácticas, se realizará una validación escrita en la fecha fijada para la prueba escrita 1.

Prueba escrita 2: Esquemas voraz, retroceso, uso de heurísticas de optimización, programación dinámica.
Porcentaje (no se pide nota mínima)

Práctica obligatoria 2: Esquemas voraz, retroceso, uso de heurísticas de optimización.

La práctica tendrá una única fecha de entrega, no se podrá entregar fuera de este plazo, y no se podrá recuperar.
Porcentaje (no se pide nota mínima)

Entrega: Antes de la fecha fijada para la prueba escrita 2.

Validación de la práctica 2: Para definir la nota final de la práctica, se realizará una validación escrita en la fecha fijada para la prueba escrita 2.

A final de curso si la nota final <5 el estudiante se podrá presentar a mejorar la nota obtenida en las pruebas escritas.

Bibliografía y recursos de información

Bàsica:

- G. Brassard y P. Bratley. Fundamentos de algoritmia. Prentice Hall. 1997.
- Cormen, T.H.; Leiserson, C.E. ; Rivest, R.L.; Stein, C. Introduction to Algorithms, (3ª edición). MIT Press, 2002. * Skiena, S. The Algorithm Design Manual. Springer 2008.

Ejercicios:

- R. Guerequeta y A. Vallecillo. Tecnicas de diseño de algoritmos. Servicio de Publicaciones de la Universidad de Málaga. 2nd Ed. 2000. http://www.lcc.uma.es/~av/Libro/indice.html
- Gonzalo, J.; Rodríguez, M. Esquemas algorítmicos: enfoque metodológico y problemas resueltos, UNED, 1997.

Implementación:

- R. Sedgewick. Algoritmos en C++. Addison-Wesley / Diaz de Santos.1995.

Adaptaciones a los contenidos debidas al COVID-19

Los contenidos teóricos de la segunda parte de la asignatura se reducirán a:

Divide y vence

programación dinámica
backtracking

Adaptaciones a la metodologia debidas al COVID-19

Las clases de teoría y laboratorio se harán por videoconferencia, en el horario de la asignatura de miércoles y jueves.