Universitat de Lleida

GUÍA DOCENTE

TÉCNICAS DE COMPUTACIÓN

Coordinación: MATEO FORNÉS, JORDI

Año académico 2019-20

Información general de la asignatura

Denominación	TÉCNICAS DE COMPUTACIÓN
Código	102376
Semestre de impartición	1R Q(SEMESTRE) EVALUACIÓN CONTINUADA
Carácter	

Grado/Máster	Curso	Carácter	Modalidad
Grado en Técnicas de Interacción Digital y de Computación	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	3
Número de grupos	1	1

Coordinación	MATEO FORNÉS, JORDI
Departamento/s	INFORMATICA E INGENIERIA INDUSTRIAL
Distribución carga docente entre la clase presencial y el trabajo autónomo del estudiante	Globalmente la assignatura tiene 150 horas de trabajo repartidas en 60 horas de clase presencial y 90 horas de trabajo individual del estudiante. 6 ECTS = 25 * 6 = 150 horas de trabajo 40% --> 60 horas presenciales 60% --> 90 horas de trabajo autónomo del estudiante
Información importante sobre tratamiento de datos	Consulte este enlace para obtener más información.
Idioma/es de impartición	Catalán (en Castellano si algún alumno muestra dificultades con el Catalán).

Profesor/a (es/as)	Dirección electrónica\nprofesor/a (es/as)	Créditos impartidos por el profesorado	Horario de tutoría/lugar
MATEO FORNÉS, JORDI	jordi.mateo@udl.cat	6	

Información complementaria de la asignatura

Para cualquier duda y/o cuestión se recomienda enviar un correo electrónico al profesor 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

Básicas:

- *CB3*: Que los estudiantes tengan la capacidad de reunir e interpretar datos relevantes (normalmente dentro de su área de estudio) para emitir juicios que incluyan una reflexión sobre temas relevantes de índole social, científica o ética

Transversales:

- *CT3*: Adquirir capacitación en el uso de las nuevas tecnologías y de las tecnologías de la información y la comunicación

Generales:

- *CG10*: Adquirir capacitación en el uso de las nuevas tecnologías y de las tecnologías de la información y la comunicación

Específicas:

- *CE2*: 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 computación.
- *CE8*: Conocimiento y aplicación de los procedimientos algorítmicos básicos de las tecnologías informáticas para diseñar soluciones a problemas computacionales, analizando la idoneidad y complejidad de los algoritmos propuestos.
- *CE9*: Conocimiento, diseño y utilización de forma eficiente los tipos y estructuras de datos más adecuados a la resolución de un problema.
- *CE10*: Capacidad para analizar, diseñar, construir y mantener aplicaciones digitales interactivas de forma

robusta, segura y eficiente, eligiendo el paradigma y los lenguajes de programación más adecuados.

Contenidos fundamentales de la asignatura

Organización del curso por temas:

- **Tema 1**: Preliminares: algoritmo, notación, lógica de predicados, técnicas de demostración.
- **Tema 2**: Especificación formal de algoritmos basada en pre-post condiciones.
- **Tema 3**: Eficiencia de los algoritmos. Notación asintótica. Análisis de algoritmos.
- **Tema 4:** Verificación formal algoritmos recursivos e iterativos.
- **Tema 5**: Técnicas de transformación de algoritmos recursivos.
- **Tema 6**: Esquemas algorítmicos: divide y vencerás.
- **Tema 7**: Esquemas algorítmicos: voraz.
- **Tema 8**: Esquemas algorítmicos: programación dinámica
- **Tema 9**: Esquemas algorítmicos: búsqueda con retroceso.
- **Tema 10**: 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 lenguajes imperativos y funcionales. 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

En las clases de teoría 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 teoría 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 dic prácticas obligatorias de la asignatura.

Setmana	Teoría	Laboratorio	Trabajo Autónomo
1	Tema 1	Problemas y Ejemplos Tema 1	Problemas y Ejemplos Tema 1 (4h)

2019-20

Setmana	Teoría	Laboratorio	Trabajo Autónomo
2	Tema 2	Problemas y Ejemplos Tema 2	Problemas y Ejemplos Tema 2 (4h)
3	Tema 3	Problemas y Ejemplos Tema 3 ***Práctica 1***: Presentación	Problemas y Ejemplos Tema 3 (3h) ***Práctica 1*** (2h)
4	Tema 4	Problemas y Ejemplos Tema 4	Problemas y Ejemplos Tema 3 (3h) ***Práctica 1*** (2h)
5	Tema 5	Problemas y Ejemplos Tema 5	Problemas y Ejemplos Tema 3 (3h) ***Práctica 1*** (2h)
6	Tema 6	Problemas y Ejemplos Tema 6	Problemas y Ejemplos Tema 6 (3h) Práctica 1 (2h)
7	Tema 6	Problemas y Ejemplos Tema 6	Problemas y Ejemplos Tema 6 (3h) ***Práctica 1*** (2h)
8	*Corrección P1. Eval. Continuada* Repaso. Dudas	***Práctica 1***: Entrega	Problemas y Ejemplos Tema 6 (3h)
9	**1er Parcial**		Estudiar
10	Tema 7	Problemas y Ejemplos Tema 7 ***Práctica 2***: Presentació	Problemas y Ejemplos Tema 7 (3h) ***Práctica 2*** (2h)
11	Tema 7	Problemas y Ejemplos Tema 7 *Corrección P2. Eval. Continuada*	Problemas y Ejemplos Tema 7 (3h) ***Práctica 2*** (2h)
12	Tema 8	Problemas y Ejemplos Tema 8	Problemas y Ejemplos Tema 8 (3h) ***Práctica 2*** (2h)
13	Tema 9	Problemas y Ejemplos Tema 9	Problemas y Ejemplos Tema 9 (3h) ***Práctica 2*** (2h)
14	Tema 10	Problemas y Ejemplos Tema 10	Problemas y Ejemplos Tema 10 (3h) ***Práctica 2*** (2h)
15	*Correcció P3. Eval. Continuada* Repaso.Dudas	***Práctica 2***: Entrega	Estudiar
16	**2° Parcial**		Estudiar
17			
18			
19	**Recuperación**		

Sistema de evaluación

Evaluación	Ponderació	Nota Mínima	En grup	Obligatòria
Examen 1er Parcial	25%	NO	NO	SI
Examen 2º Parcial	25%	NO	NO	SI
Prácticas	35%	5	SI (màx 2)	SI
Problemas	15%	NO	SI (màx 2)	SI

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:

- Baynat B., Chrétienne P. Hanen C., Kedad-Sidhoum S., Munier-Kordon A., Picouleau C. Exercices et problèmes d'algoritmique. Ed. Dunod. 3r. ed. 2010.
- 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.