GUÍA DOCENTE ABREVIADA DE LA ASIGNATURA

G682 - Diseño de Algoritmos

Grado en Ingeniería Informática

Curso Académico 2018-2019

1. DATOS IDENTIFICATIVOS					
Título/s	Grado en Ingeniería Informática			Tipología y Curso	Optativa. Curso 4
Centro	Facultad de Ciencias				
Módulo / materia	MATERIA DE COMPUTACIÓN MENCIÓN EN COMPUTACIÓN				
Código y denominación	G682 - Diseño de Algoritmos				
Créditos ECTS	6	Cuatrimestre		Cuatrimestral (1)	
Web					
Idioma de impartición	Español	English friendly	No	Forma de impartición	Presencial

Departamento	DPTO. MATEMATICAS, ESTADISTICA Y COMPUTACION
Profesor responsable	JOSE LUIS MONTAÑA ARNAIZ
E-mail	joseluis.montana@unican.es
Número despacho	Facultad de Ciencias. Planta: + 1. DESPACHO PROFESORES (1036)
Otros profesores	

3.1 RESULTADOS DE APRENDIZAJE
- Profundizar en el uso de técnicas de análisis de eficiencia de algoritmos. Saber razonar sobre la corrección y la eficiencia de algoritmos. Comprender algoritmos clásicos para problemas fundamentales. Saber identificar los componentes más relevantes de un problema y seleccionar la técnica algorítmica más adecuada. Ser capaz de escoger los tipos de datos más adecuados para mejorar la eficiencia de una solución algorítmica.
- Conocer comprender y saber utilizar estrategias de diseño de algoritmos. Para cada una de las siguientes técnicas: algoritmos voraces, divide y vencerás, vuelta atrás y ramificación y programación dinámica: el esquema general, el análisis de eficiencia en el caso peor, y la aplicación a algunos problemas clásicos.
- Conocer, comprender y saber utilizar estructuras de datos avanzadas: tries, mf-sets, dispersión abierta y cerrada.
- Conocer y comprender los conceptos de algoritmo probabilista y las distintas clases de complejidad asociadas a los mismos. Conocer y comprender el concepto de algoritmo aproximado y de algoritmo probablemente aproximadamente correcto.
- Conocer y comprender el concepto de algoritmo heurístico y de técnica metaherística. Saber cuando un problema debe ser atacado mediante técnicas heurísticas.

4. OBJETIVOS
Conocer el esquema de los algoritmos voraces, identificar cuándo y cómo aplicarlo, conocer las técnicas más habituales de demostración de la corrección de estos algoritmos, y familiarizarse con algunos algoritmos voraces fundamentales, tales como el algoritmo de Dijkstra, el de Kruskal y el de Prim.
Conocer el esquema de programación dinámica, identificar cuándo se puede aplicar y cómo y familiarizarse con algunos algoritmos de programación dinámica fundamentales, por ejemplo, el algoritmo de Floyd o el cálculo de la distancia de edición
Conocer el problema básico de cálculo de flujos óptimos sobre redes, familiarizarse con un algoritmo básico (Ford-Fulkerson), entender el teorema de maxflow-mincut, reconocer cuándo un problema se puede formular en términos de un problema de flujos
Conocer la formulación de la programación lineal (LP), entender cómo muchos problemas computacionales se pueden formular en términos de LP, en concreto los problemas de flujos en redes, entender los principios del algoritmo de simplex, entender la relación fundamental entre un problema de LP (primal) y su dual
Comprender la importancia del aleatorización en el diseño de algoritmos y estructuras de datos, familiarizarse con algunas técnicas elementales de análisis probabilístico necesarias para estudiar la eficiencia de los algoritmos aleatoritados y familiarizarse con algunos ejemplos clásicos como quicksort aleatorizado, las skip lists, el test de primalidad de Rabin o el algoritmo de búsqueda de patrones de Karp-Rabin
Conocer algunos problemas computacionales específicos que surgen en ámbitos tan dispares como la búsqueda en bases de datos documentales, bases de datos proteínicas y genómicas, sistemas de información geográfica, recuperación de información basada en el contenido, compresión de datos, etc. y conocer algunas estructuras de datos avanzadas para dar respuesta a estas necesidades
Familiarizarse con el uso de los principios de diseño algorítmico para el diseño de estructuras de datos y conocer algunas técnicas esenciales para obtener implementaciones de éstas que garanticen la máxima eficiencia y saquen partido de las caracterísitcas específicas del hardware que debe soportar estas estructuras de datos
Desarrollar los hábitos, actitudes y habilidades necesarias para poder estudiar, de manera independiente o en equipo, un tema específico, haciendo uso de las fuentes de información disponibles (bibliografía, Web, ...) y conseguir el nivel de conocimiento y compresión del tema suficiente como para poder explicarlo a terceros, escribiendo un resumen y preparando un material audiovisual complementario
Conocer y comprender algunos principios básicos para el diseño de experimentos computacionales y aprender técnicas básicas de recogida de datos, validación y análisis estadístico de los datos recogidos y cómo sacar conclusiones; reconocer la necesidad, la utilidad y las limitaciones de los estudios experimentales en el diseño e implementación de algoritmos y estructuras de datos

6. ORGANIZACIÓN DOCENTE

	CONTENIDOS
1	Conceptos Algorítmicos Fundamentales. Análisis de eficiencia. Caso peor y caso medio. Notación de Landau. Divide y vencerás. Análisis de algoritmos recursivos: Binary Search. Quicksort, MergeSort, QuickSetect. Análisis de Estructuras de Datos: Árboles Binarios de Búsqueda. Árboles Balanceados. Heaps. Tries. Análisis de algoritmos voraces y estructuras asociadas: Dijkstra, Prim, Kruskal. Particiones. Hashing. Montículos de Fibonacci.
2	Programación Dinámica. Principio de optimalidad. Memoización. Distancia de edición. Árboles binarios de búsqueda óptimos. Algoritmo de Floyd-Warshall para caminos mínimos. Problema del viajante de comercio. Problema de la mochila.
3	Algoritmos probabilistas. Clases de complejidad BPP y ZPP. Análisis probabilístico. Ejemplos: test de primalidad de Rabin, skip lists, hashing universal.
4	Programación Lineal. Flujo en Redes. Juegos de suma cero. Evaluación de Circuitos. Método del Simplex: algoritmo detallado y análisis de eficiencia.
5	Algoritmos aproximados y heurísticos. Tipos de aproximación, relativa y absoluta. Cubrimiento de vértices. cubrimiento de vértices métrico. Cubrimiento de conjuntos. Mochila. Búsquedas Locales, Heurísticas y Metaheurísticas.

7. MÉTODOS DE LA EVALUACIÓN

Descripción	Tipología	Eval. Final	Recuper.	%
Prueba de conocimiento y relación de contenidos	Examen escrito	Sí	Sí	40,00
Prácticas	Evaluación en laboratorio	No	Sí	20,00
Problemas	Examen escrito	No	Sí	20,00
Trabajo en grupo y presentación oral	Trabajo	No	No	20,00
TOTAL				100,00

Observaciones

Las prácticas y ejercicios se evalúan a lo largo del curso debiendo ser entregadas en fechas señaladas al efecto.

Criterios de evaluación para estudiantes a tiempo parcial

Los alumnos que cursen la asignatura a tiempo parcial y que no puedan acogerse al procedimiento ordinario de evaluación podrán superar la asignatura concurriendo a un examen escrito (60% de la calificación) y un examen en laboratorio de prácticas y ejercicios (40% de la calificación). La nota mínima para hacer media en cada parte es de 4 puntos sobre 10.

8. BIBLIOGRAFÍA Y MATERIALES DIDÁCTICOS

BÁSICA

KLEINBERG, J.; TARDOS, E. , Algorithm Design , Addison-Wesley , 2005 .

CORMEN, T.; LEISERSON, C.; RIVEST; R.; STEIN, C. , Introduction to Algorithms , The MIT Press , 2009 .

SEDGEWICK, R. , Algorithms in C++: Part 1-4 & Part 5 (3rd ed) , Addison-Wesley , 2002 .

DASGUPTA, S.; PAPADIMITRIOU, C.; VAZIRANI, U. , Algorithms , McGraw-Hill , 2008 .

Esta es la Guía Docente abreviada de la asignatura. Tienes también publicada en la Web la información más detallada de la asignatura en la Guía Docente Completa.