Guía docente
270021 - A - Algorítmica

Última modificación: 30/01/2024

Unidad responsable: Facultad de Informática de Barcelona
Unidad que imparte: 723 - CS - Departamento de Ciencias de la Computación.

Titulación: GRADO EN INGENIERÍA INFORMÁTICA (Plan 2010). (Asignatura optativa).

Curso: 2023 **Créditos ECTS:** 6.0 **Idiomas:** Catalán, Castellano, Inglés

PROFESORADO

Profesorado responsable: MARIA JOSE SERNA IGLESIAS

Otros:

Primer quadrimestre:
MARIA JOSE BLESA AGUILERA - 11, 21
AMALIA DUCH BROWN - 12
SANTIAGO MARCO SOLA - 14, 22
CONRADO MARTÍNEZ PARRA - 21, 22
MARIA JOSE SERNA IGLESIAS - 11, 12, 13, 14

Segon quadrimestre:
MARIA JOSE BLESA AGUILERA - 12, 13
CONRADO MARTÍNEZ PARRA - 11, 12, 13
MARIA JOSE SERNA IGLESIAS - 11

CAPACIDADES PREVIAS

- Familiaridad con las técnicas básicas de programación y el lenguaje de programación C++: iteraciones, alternativas, funciones recursivas,
paso de parámetros, punteros, referencias, memoria dinámica, clases, objetos, métodos, ...

- Conocimiento de conceptos algorítmicos básicos: eficiencia de algoritmos, notación asintótica, grafos, recorridos de grafos, estructuras de datos (listas, árboles de búsqueda, hash, heaps, ...)

- Conocimientos básicos de matemática discreta, algebra lineal y cálculo

- Conocimientos básicos de teoría de probabilidad y estadística

- Conocimientos básicos de arquitectura de computadores y de la jerarquía de memoria

REQUISITOS

- Pre-requisito EDA
- Pre-Corequisito PE
- Corequisito PROP

COMPETENCIAS DE LA TITULACIÓN A LAS QUE CONTRIBUYE LA ASIGNATURA

Específicas:
CCO1.1. Evaluar la complejidad computacional de un problema, conocer estrategias algorítmicas que puedan conducir a su resolución, y recomendar, desarrollar e implementar la que garantice el mejor rendimiento de acuerdo con los requisitos establecidos.
CCO2.5. Implementar software de búsqueda de información (information retrieval).
CCO3.1. Implementar código crítico siguiendo criterios de tiempo de ejecución, eficiencia y seguridad.
CCO3.2. Programar considerando la arquitectura hardware, tanto en ensamblador como en alto nivel.
CT1.2C. Interpretar, seleccionar y valorar conceptos, teorías, usos y desarrollos tecnológicos relacionados con la informática y su aplicación a partir de los fundamentos matemáticos, estadísticos y físicos necesarios. CEFB1: Capacidad para la resolución de los problemas matemáticos que puedan plantarse en la ingeniería. Aptitud para aplicar los conocimientos sobre: algebra, cálculo diferencial e integral i métodos numéricos; estadística y optimización.
CT4.1. Identificar las soluciones algorítmicas más adecuadas para resolver problemas de dificultad mediana.
CT4.2. Razonar sobre la corrección y la eficiencia de una solución algorítmica.
CT5.2. Conocer, diseñar y utilizar de forma eficiente los tipos y las estructuras de datos más adecuados para la resolución de un problema.
CT5.3. Diseñar, escribir, probar, depurar, documentar y mantener código en un lenguaje de alto nivel para resolver problemas de programación aplicando esquemas algorítmicos y usando estructuras de datos.
CT5.4. Diseñar la arquitectura de los programas utilizando técnicas de orientación a objetos, de modularización y de especificación e implementación de tipos abstractos de datos.

Genéricas:
G7. APRENDIZAJE AUTÓNOMO: Detectar carencias en el propio conocimiento y superarlas mediante la reflexión crítica y la elección de la mejor actuación para ampliar este conocimiento. Capacidad para el aprendizaje de nuevos métodos y tecnologías y versatilidad para adaptarse a nueves situaciones.

METODOLOGÍAS DOCENTES

Las clases de teoria son de estilo magistral, con exposiciones teóricas por parte del profesor, intercaladas con numerosos ejemplos. Se espera que los estudiantes participen activamente con sus preguntas y comentarios a lo largo de estas clases. Cada semana se hacen dos horas y dos de problemas. En las clases de problemas se discutirán las soluciones propuestas por los estudiantes a los ejercicios planteados por el profesor con antelación (enunciados más complejos, con los que el estudiante ha podido trabajar durante una semana, autónomamente) o a los ejercicios cortos planteados en la propia clase, para trabajarlos en equipos de dos estudiantes o individualmente. Ocasionalmente, los estudiantes podrán ser requeridos para exponer su solución al resto de sus compañeros.

Complementado el estudio personal y la resolución de ejercicios prácticos en papel, se propondrá un proyecto de programación en el que el estudiante habrá de diseñar y codificar programas en C++ que resuelvan los problemas planteados, p.e. implementar dos (o más) algoritmos que resuelven un mismo problema y llevar a cabo experimentos que permitan comparar la eficiencia de los algoritmos, y a su vez, comparar dichas eficiencias con las predicciones del análisis teórico. El proyecto se desarrollará en equipo (equipos de dos o tres personas).
Él proyecto se usará para evaluar la competencia de aprendizaje autónomo, ya que requerirá estudiar un tema concreto, relacionado con los que se ven a lo largo del curso, pero no expuesto en clase de teoria.

OBJETIVOS DE APRENDIZAJE DE LA ASIGNATURA

1.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.
2.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
3.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
4.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.
5.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
6.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
7.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
8.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

HORAS TOTALES DE DEDICACIÓN DEL ESTUDIANTADO

Tipo	Horas	Porcentaje
Horas grupo mediano	30,0	20.00
Horas aprendizaje autónomo	84,0	56.00
Horas actividades dirigidas	6,0	4.00
Horas grupo grande	30,0	20.00

Dedicación total: 150 h

CONTENIDOS

Conceptos Algorítmicos Básicos

Descripción:
Análisis de caso peor. Notación asintótica. Divide y vencerás. Análisis de algoritmos recursivos. Ordenación lineal. Algortmos para grafos. Aleatorització.

Algoritmos Voraces

Descripción:
Esquema voraz. Planificación de tareas. Algoritmos de Bellman-Ford y Johnson para caminos mínimos. Algoritmos de Kruskal y Prim para árboles de expansión mínimos. Particiones (Union-find). Códigos de Huffman.

Programación Dinámica

Descripción:
Principio de optimalidad. Memoización. Árboles binarios de búsqueda óptimos. Algoritmo de Floyd-Warshall para caminos mínimos. Problema del viajante de comercio. Problema de la mochila. Otros ejemplos.

Flujos sobre Redes

Descripción:
Conceptos básicos. Teorema de maxflow-mincut. Algoritmo de Ford-Fulkerson. Aplicaciones: matchings y caminos sin aristas en común. Dualidad.

Estructuras de Datos y Algoritmos Avanzadoss

Descripción:
Una selección de algunos de los siguientes algoritmos y/o estructuras de datos (o otros). Programación lineal. Montículos de Fibonnacci. Hashing. Filtros Bloom. Blockchains. Map Reduce. Grafos aleatorios. PageRank.

ACTIVIDADES

Conceptos Algorítmicos Básicos

Descripción:
Recordar conceptos fundamentales aprendidos en las asignaturas previas, y familiarizarse con la terminología y notación que se utilizará a lo largo del curso. Aprender otras técnicas algorítmicas básicas.

Dedicación: 18h
Grupo grande/Teoría: 4h
Grupo mediano/Prácticas: 4h
Aprendizaje autónomo: 10h

Algoritmos voraces

Descripción:
Atender las clases de teoría y problemas dónde se desarrolla el tema y hacer los ejercicios propuestos por el profesor para hacer en casa o en clase

Objetivos específicos:
1

Competencias relacionadas:
G7. APRENDIZAJE AUTÓNOMO: Detectar carencias en el propio conocimiento y superarlas mediante la reflexión crítica y la elección de la mejor actuación para ampliar este conocimiento. Capacidad para el aprendizaje de nuevos métodos y tecnologías y versatilidad para adaptarse a nueves situaciones.

Dedicación: 19h
Grupo grande/Teoría: 5h
Grupo mediano/Prácticas: 4h
Aprendizaje autónomo: 10h

Programación Dinámica

Descripción:
Atender las clases de teoría y problemas dónde se desarrolla el tema y hacer los ejercicios propuestos por el profesor para hacer en casa o en clase

Objetivos específicos:
2

Competencias relacionadas:
G7. APRENDIZAJE AUTÓNOMO: Detectar carencias en el propio conocimiento y superarlas mediante la reflexión crítica y la elección de la mejor actuación para ampliar este conocimiento. Capacidad para el aprendizaje de nuevos métodos y tecnologías y versatilidad para adaptarse a nueves situaciones.

Dedicación: 32h
Grupo grande/Teoría: 7h
Grupo mediano/Prácticas: 10h
Aprendizaje autónomo: 15h

Flujos sobre Redes

Descripción:
Atender las clases de teoría y problemas dónde se desarrolla el tema y hacer los ejercicios propuestos por el profesor para hacer en casa o en clase

Objetivos específicos:
3

Competencias relacionadas:
G7. APRENDIZAJE AUTÓNOMO: Detectar carencias en el propio conocimiento y superarlas mediante la reflexión crítica y la elección de la mejor actuación para ampliar este conocimiento. Capacidad para el aprendizaje de nuevos métodos y tecnologías y versatilidad para adaptarse a nueves situaciones.

Dedicación: 32h
Grupo grande/Teoría: 7h
Grupo mediano/Prácticas: 10h
Aprendizaje autónomo: 15h

Estructuras de Datos Avanzadas

Descripción:
Atender las clases de teoría y problemas dónde se desarrolla el tema y hacer los ejercicios propuestos por el profesor para hacer en casa o en clase

Objetivos específicos:
5, 6

Competencias relacionadas:
G7. APRENDIZAJE AUTÓNOMO: Detectar carencias en el propio conocimiento y superarlas mediante la reflexión crítica y la elección de la mejor actuación para ampliar este conocimiento. Capacidad para el aprendizaje de nuevos métodos y tecnologías y versatilidad para adaptarse a nueves situaciones.

Dedicación: 8h
Grupo grande/Teoría: 2h
Grupo mediano/Prácticas: 2h
Aprendizaje autónomo: 4h

Proyecto de Aprendizaje Autónomo

Descripción:
El profesor asignará un proyecto de programación de mediana envergadura que involucra el estudio autónomo de un tema concreto incluyendo alguna componente que no se ha visto en clase. A final del curso podrá entrevistar al equipo para complementar la informacióde cara a la evaluación de esta actividad
Los estudiantes formarán equipos de dos o tres personas para desarrollar un programa junto con la documentación escrita y material adicional.

Objetivos específicos:
1, 2, 3, 4, 5, 6, 7, 8

Competencias relacionadas:
G7. APRENDIZAJE AUTÓNOMO: Detectar carencias en el propio conocimiento y superarlas mediante la reflexión crítica y la elección de la mejor actuación para ampliar este conocimiento. Capacidad para el aprendizaje de nuevos métodos y tecnologías y versatilidad para adaptarse a nueves situaciones.

Dedicación: 15h
Aprendizaje autónomo: 15h

Examen Final/Segundo parcial

Objetivos específicos:
1, 2, 3, 4, 5, 7

Competencias relacionadas:
G7. APRENDIZAJE AUTÓNOMO: Detectar carencias en el propio conocimiento y superarlas mediante la reflexión crítica y la elección de la mejor actuación para ampliar este conocimiento. Capacidad para el aprendizaje de nuevos métodos y tecnologías y versatilidad para adaptarse a nueves situaciones.

Dedicación: 13h
Actividades dirigidas: 3h
Aprendizaje autónomo: 10h

Examen parcial

Descripción:
Examen parcial liberatorio fuera de horas de clase.

Objetivos específicos:
1, 2, 5, 6, 7

Competencias relacionadas:
G7. APRENDIZAJE AUTÓNOMO: Detectar carencias en el propio conocimiento y superarlas mediante la reflexión crítica y la elección de la mejor actuación para ampliar este conocimiento. Capacidad para el aprendizaje de nuevos métodos y tecnologías y versatilidad para adaptarse a nueves situaciones.

Dedicación: 7h
Actividades dirigidas: 2h
Aprendizaje autónomo: 5h

Tutorización del proyecto

Descripción:
Resolver dudas, hacer el seguimiento de la actividad, etc

Objetivos específicos:
6, 7, 8

Competencias relacionadas:
G7. APRENDIZAJE AUTÓNOMO: Detectar carencias en el propio conocimiento y superarlas mediante la reflexión crítica y la elección de la mejor actuación para ampliar este conocimiento. Capacidad para el aprendizaje de nuevos métodos y tecnologías y versatilidad para adaptarse a nueves situaciones.

Dedicación: 6h
Actividades dirigidas: 6h

SISTEMA DE CALIFICACIÓN

La nota final (NF) se calcula a partir de la nota de la resolución de problemas algorítmicos (A), la de las pruebas escritas parcial (M) (materia correspondiente a las 6 -7 primeras semanas del curso) y examen final (F); y la nota del proyecto asociado al aprendizaje autónomo (P).

La nota final se calcula de acuerdo con la fórmula:

NF = 0.7 max(0.4 M + 0.6 F, F) + 0.2 P +0.1 A

El profesorado evaluará el grado de adquisición de la competencia de aprendizaje autónomo a partir de la nota obtenida en un ejercicio de programación que involucra un trabajo de aprendizaje autónomo por parte de los alumnos. La nota P se evaluarà en una escala numérica de 0 a 10.

La nota cualitativa de la competencia transversal se determina a partir de la nota numèrica
P según los siguientes tramos: [0,5) => D, [5,6.5) => C, [6.5,8.5) => B, [8.5,10] => A

BIBLIOGRAFÍA

Básica:
- Kleinberg, J.; Tardos, E. Algorithm design. Pearson, 2014. ISBN 9781292023946.
- Dasgupta, S.; Papadimitriou, C.; Vazirani, U. Algorithms. McGraw-Hill, 2008. ISBN 9780073523408.
- Moore, C.; Mertens, S. The Nature of computation. Oxford University press, 2011. ISBN 9780199233212.

Complementaria:
- Brassard, G.; Bratley, P. Fundamentals of algorithmics. Prentice-Hall International, 1996. ISBN 013073487X.
- Skiena, S. The algorithm design manual. 2nd ed. Springer, 2008. ISBN 9781848000698.
- Skiena, S.S. The algorithm design manual. Third edition. Cham: Springer, 2020. ISBN 9783030542559.
- Mehta, D.P.; Sahni, S. (eds.). Handbook of data structures and applications. Chapman & Hall/CRC, 2005. ISBN 1584884355.
- Lamport, L. LaTeX: a document preparation system: user's guide and reference manual. 2nd ed. Addison-Wesley, 1994. ISBN 0201529831.
- Motwani, R.; Raghavan, P. Randomized Algorithms. Cambridge University Press, 1995. ISBN 0521474655.
- Mitzenmacher, M.; Upfal, E. Probability and computing: randomized algorithms and probabilistic analysis. Cambridge University Press, 2005. ISBN 0521835402.
- Easley, D.; Kleinberg, J. Networks, crowds, and markets: reasoning about a highly connected world. Cambridge University Press, 2010. ISBN 9780521195331.
- Cormen, T.H [et al.]. Introduction to algorithms. 4th ed. Cambridge: MIT Press, 2022. ISBN 9780262046305.
- Sedgewick, R.; Wayne, K. Algorithms. 4th ed. Upper Saddle River, NJ: Addison-Wesley, 2011. ISBN 9780321573513.