



Grado en Ingeniería Informática 30229 - Algoritmia básica

Guía docente para el curso 2014 - 2015

Curso: 3, Semestre: 2, Créditos: 6.0

Información básica

Profesores

- **Francisco Javier Campos Laclaustra** jcampos@unizar.es

Recomendaciones para cursar esta asignatura

Interés y esfuerzo, además de los conocimientos adquiridos en las asignaturas previas de matemáticas, programación, estructuras de datos y algoritmos y teoría de la computación.

Actividades y fechas clave de la asignatura

El calendario de exámenes y las fechas de entrega de trabajos se anunciarán con suficiente antelación en la página web de la asignatura.

Inicio

Resultados de aprendizaje que definen la asignatura

El estudiante, para superar esta asignatura, deberá demostrar los siguientes resultados...

- 1:** Conoce esquemas algorítmicos variados y problemas fundamentales que utilizan los mismos.
- 2:** Sabe particularizar esquemas algorítmicos generales para resolver problemas.
- 3:** Sabe identificar las componentes más relevantes de un problema y seleccionar la técnica algorítmica más adecuada para el mismo, además de argumentar de forma razonada dicha elección.
- 4:** Sabe comparar problemas y utilizar dicha comparación para resolver un problema a partir de una solución eficiente de otro.
- 5:** Sabe razonar sobre la corrección y eficiencia de los algoritmos avanzados que se utilizan.

- 6:** Habilidad para trabajar en grupo, identificar objetivos del grupo, trazar un plan de trabajo para alcanzarlo, reconocer los diferentes papeles dentro de un equipo y asume el compromiso de las tareas encomendadas.
- 7:** Gestión del autoaprendizaje y de desarrollo incluyendo el tiempo de gestión y de organización.
- 8:** Apreciar la necesidad del aprendizaje continuo.

Introducción

Breve presentación de la asignatura

El reto de diseñar algoritmos eficientes para resolver problemas puede resultar apasionante. En esta asignatura se presentan una serie de técnicas algorítmicas eficientes que pueden ser aplicadas para resolver con elegancia un amplio espectro de problemas de interés práctico.

La URL de la web de la asignatura es <http://webdiis.unizar.es/asignaturas/AB/>

Contexto y competencias

Sentido, contexto, relevancia y objetivos generales de la asignatura

La asignatura y sus resultados previstos responden a los siguientes planteamientos y objetivos:

En esta asignatura el alumno mejorará su capacidad para diseñar y desarrollar algoritmos haciendo énfasis en la identificación y aplicación de los esquemas que puedan ser utilizables para obtener soluciones eficientes a una amplia gama de problemas. Se presentarán algunos de los esquemas algorítmicos fundamentales, como: dividir para vencer, voracidad, programación dinámica, búsqueda con retroceso, ramificación y poda, etc. El alumno aprenderá a reconocer los problemas que requieren este tipo de esquemas para su resolución y cómo aplicarlos.

Contexto y sentido de la asignatura en la titulación

La *Especialidad en Computación* abarca una amplia gama de conceptos, desde los fundamentos teóricos y algorítmicos hasta la vanguardia de los desarrollos en bioinformática, robótica, visión por computador, videojuegos, y otras áreas interesantes. *Algoritmia básica* es la primera de las asignaturas de la Especialidad y se centra en el diseño de algoritmos correctos y eficientes mediante el conocimiento y aplicación de un conjunto de esquemas reutilizables.

Al superar la asignatura, el estudiante será más competente para...

- 1:** Combinar los conocimientos generalistas y los especializados de Ingeniería para generar propuestas innovadoras y competitivas en la actividad profesional.
- 2:** Resolver problemas y tomar decisiones con iniciativa, creatividad y razonamiento crítico.
- 3:** Comunicar y transmitir conocimientos, habilidades y destrezas en castellano y en inglés.
- 4:** Evaluar la complejidad computacional de un problema, conocer estrategias algorítmicas que puedan conducir a su resolución y recomendar, desarrollar e implementar aquella que garantice el mejor rendimiento de acuerdo con los requisitos establecidos.

Importancia de los resultados de aprendizaje que se obtienen en la asignatura:

La Algoritmia se define como el estudio de los algoritmos. Cuando nos disponemos a resolver un problema, es posible que haya toda una gama de algoritmos disponibles. En este caso, es importante decidir cuál de ellos hay que utilizar. Dependiendo de nuestras prioridades y de los límites del equipo que esté disponible para nosotros, quizá necesitemos seleccionar el algoritmo que requiera menos tiempo, o el que utilice menos espacio, o el que sea más fácil de programar y así sucesivamente. La respuesta puede depender de muchos factores, tales como los números implicados, la forma en que se presenta el problema, o la velocidad y capacidad de almacenamiento del equipo de computación disponible. Quizás suceda que ninguno de los algoritmos disponibles sea totalmente adecuado, así que tendremos que diseñar un algoritmo nuevo por nuestros propios medios. La Algoritmia es la ciencia que nos permite evaluar el efecto de estos diferentes factores externos sobre los algoritmos disponibles, de tal modo que sea posible seleccionar el que más se ajuste a nuestras circunstancias particulares; también es la ciencia que nos indica la forma de diseñar un nuevo algoritmo para una tarea concreta.

Evaluación

Actividades de evaluación

El estudiante deberá demostrar que ha alcanzado los resultados de aprendizaje previstos mediante las siguientes actividades de evaluación

1:

A lo largo del cuatrimestre se plantearán enunciados prácticos de programación que deberán ser resueltos en el laboratorio. Para ello se formarán equipos integrados por un número determinado de alumnos que se fijará al principio del curso. Los trabajos presentados por los alumnos se calificarán con una nota cuantitativa de 0 a 10. Para obtener dichas notas se valorará el funcionamiento de los programas según especificaciones, la calidad de su diseño y su presentación, la adecuada aplicación de los métodos de resolución, el tiempo empleado, así como la capacidad de cada uno de los integrantes del equipo para explicar y justificar el diseño realizado.

Los alumnos que hayan cumplido con los plazos de entrega fijados para los trabajos prácticos de programación serán exentos de la realización del examen práctico de programación en laboratorio. Para dichos alumnos, la calificación obtenida con sus prácticas se utilizará como nota de examen práctico de programación en laboratorio, ponderando un 30% de la nota final de la asignatura, salvo que el alumno decida presentarse a la prueba de examen práctico en laboratorio, en cuyo caso prevalecerá la nota obtenida en el examen práctico individual.

2:

Además, en la mitad aproximada del cuatrimestre tendrá lugar una prueba escrita intermedia que consistirá en la realización individual de ejercicios propuesto por el profesor. La prueba escrita intermedia se calificará con una nota cuantitativa de 0 a 10.

Los alumnos que hayan realizado la prueba escrita intermedia serán exentos de la realización de una parte del examen escrito. Para dichos alumnos, la calificación obtenida en la prueba escrita intermedia se utilizará como parte de la nota final, con una ponderación del 30%, salvo que el alumno decida presentarse al examen final completo, en cuyo caso prevalecerá la nota obtenida en éste.

3:

Evaluación global

La prueba global de evaluación de la asignatura consta de dos partes:

- **Examen práctico** de programación en laboratorio e individual. En cada convocatoria se realizará un examen práctico de programación en laboratorio, en el que se le plantearán al alumno ejercicios de naturaleza similar a los realizados en las prácticas o vistos en clase. La calificación obtenida pondera un 30% de la nota final de la asignatura.

Los alumnos que hayan cumplido con los plazos de entrega fijados para las prácticas de laboratorio serán exentos de la realización del examen práctico de programación en laboratorio convirtiéndose

automáticamente la nota numérica obtenida en la evaluación de sus prácticas de laboratorio en su nota final de examen práctico de programación. No obstante, para los alumnos exentos del examen práctico que se presenten al mismo, en cualquier convocatoria, prevalecerá la nota obtenida en el examen práctico individual de programación.

- **Examen escrito** en el que se deberán resolver problemas de naturaleza similar a los planteados en clase y en la prueba escrita intermedia y, en su caso, responder preguntas conceptuales o resolver algún ejercicio en el que se demuestre haber logrado los resultados de aprendizaje requeridos en la asignatura. La calificación obtenida pondera un 70% de la nota final de la asignatura.

Los alumnos que hayan realizado la prueba escrita intermedia estarán exentos de realizar una parte del examen escrito, debiendo realizar sólo una parte obligatoria del examen, que se determinará en ese mismo momento. En ese caso, la calificación obtenida en la prueba escrita intermedia pondera un 30%, y la de la parte obligatoria del examen escrito final un 40%.

4:

En resumen, opción de liberación progresiva de exámenes finales:

1. Parte práctica:

- Prácticas de laboratorio (en grupo) durante el cuatrimestre: 30%.

2. Parte de teoría y problemas:

- Prueba escrita intermedia: 30%.
- Examen final (sólo una parte): 40%.

5:

Por el contrario, opción basada exclusivamente en exámenes finales:

1. Parte práctica:

- Examen práctico (individual) de programación en laboratorio: 30%.

2. Parte de teoría y problemas:

- Examen final (completo): 70%.

Actividades y recursos

Presentación metodológica general

El proceso de aprendizaje que se ha diseñado para esta asignatura se basa en lo siguiente:

El estudio y trabajo continuado desde el primer día de clase.

El aprendizaje de conceptos y metodologías para el diseño e implementación de algoritmos correctos y eficientes a través de las clases magistrales, en las que se favorecerá la participación de los alumnos.

La aplicación de tales conocimientos al diseño y análisis de algoritmos y programas en las clases de problemas. En estas clases los alumnos desempeñarán un papel activo en la discusión y resolución de los problemas.

El trabajo en equipo desarrollado para resolver las prácticas de la asignatura, equipos integrados como máximo por dos alumnos, y cuyo resultado se plasma en la entrega de programas resultantes convenientemente diseñados y documentados, así como en la explicación y justificación del diseño realizado y decisiones adoptadas.

Un trabajo continuado en el que se conjugue la comprensión de conceptos, el análisis y la resolución de problemas de programación utilizando "lápiz y papel" y la puesta a punto en computador de algunos proyectos de programación.

Actividades de aprendizaje programadas (Se incluye programa)

El programa que se ofrece al estudiante para ayudarle a lograr los resultados previstos comprende las siguientes actividades...

- 1:** En las clases impartidas en el aula se desarrollará el temario de la asignatura.
- 2:** En las clases de problemas se resolverán problemas de aplicación de los conceptos y técnicas presentadas en el programa de la asignatura. Se propondrán problemas y ejercicios para ser resueltos antes de la clase de problemas en la que se presentarán y discutirán diferentes soluciones a dichos problemas. También se propondrán ejercicios durante la sesión de problemas para ser resueltos durante la misma, algunos de forma individual y otros para ser trabajados en grupo.
- 3:** El trabajo de prácticas se desarrolla en un laboratorio informático o bien en los computadores personales de los alumnos, en casa. En estas sesiones los alumnos deberán trabajar en equipo y realizar una serie de trabajos de programación directamente relacionados con los temas estudiados en la asignatura. Para ello se propondrán una serie de trabajos o ejercicios de programación para que los alumnos los resuelvan en grupo y los entreguen dentro de los plazos de tiempo que se fijen en cada caso.

Planificación y calendario

Calendario de sesiones presenciales y presentación de trabajos

La organización docente de la asignatura prevista es la siguiente.

- Clases teóricas (2 horas semanales)
- Clases de problemas (1 hora semanal)

Presentación de trabajos prácticos:

Los trabajos de programación a desarrollar en laboratorio o en casa deberán ser realizados y presentados de acuerdo a lo especificado para cada uno de ellos, y dentro de las fechas límite que se anunciarán en el enunciado de cada uno de los trabajos propuestos o con suficiente antelación.

Programa

Programa de la asignatura

1. Introducción.
2. Divide y vencerás.
3. Algoritmos voraces.
4. Programación dinámica.
5. Búsqueda con retroceso.
6. Ramificación y poda.
7. Programación lineal y reducciones.

Trabajo

Trabajo del estudiante

La dedicación del estudiante para alcanzar los resultados de aprendizaje en esta asignatura se estima en 156 horas distribuidas del siguiente modo:

- 45 horas, aproximadamente, de actividades presenciales (clases teóricas y de problemas);
- 45 horas de trabajo de programación en equipo para desarrollar los programas propuestos como trabajo de laboratorio;
- 60 horas de estudio personal efectivo (estudio de apuntes y textos, resolución de problemas, preparación de clases, desarrollo de programas);
- 6 horas de examen final de teoría escrito y de examen práctico en laboratorio.

Bibliografía

Bibliografía recomendada

Básica:

T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein. *Introduction to Algorithms (3rd edition)*, The MIT Press, 2009.

S. Dasgupta, C. Papadimitriou, U. Vazirani. *Algorithms*, McGraw-Hill, 2008.

G. Brassard, P. Bratley. *Fundamentos de Algoritmia*, Prentice Hall, 1997.

Complementaria:

J. Kleinberg, E. Tardos. *Algorithm Design*, Addison-Wesley, 2005.

I. Parberry and W. Gasarch. *Problems on Algorithms (2nd edition)*, free book, 2002.

Además: transparencias disponibles en la página web, fotocopias de problemas, ejercicios, etc.

Referencias bibliográficas de la bibliografía recomendada

Escuela Universitaria Politécnica

- Brassard, Gilles. Fundamentos de algoritmia / G. Brassard, T. Bratley ; traducción, Rafael García-Bermejo ; revisión técnica, Narciso Martí, Ricardo Peña, Luis Joyanes Aguilar . 1ª ed. en español, reimp. Madrid [etc.] : Prentice Hall, 2008
- Dasgupta, Sanjoy. Algorithms / Sanjoy Dasgupta, Christos Papadimitriou, Umesh Vazirani . Boston [etc.] : McGraw Hill Higher Education, cop. 2008
- Introduction to algorithms / Thomas H. Cormen ... [et al.] . 3rd ed. Cambridge, Massachusetts ; London : MIT Press, cop. 2009
- Kleinberg, Jon. Algorithm design / Jon Kleinberg, Éva Tardos . Boston : Pearson/Addison-Wesley, cop. 2006
- Parberry, I. Problems on algorithms (ebook) / Ian Parberry, W. Gasarch. - 2ª ed. New Jersey : Prentice Hall, 2005

Escuela Politécnica Superior

1. Introduction to algorithms / Thomas H. Cormen ... [et al.] . - 3rd ed. Cambridge, Massachusetts ; London : MIT Press, cop. 2009
2. Dasgupta, Sanjoy. Algorithms / Sanjoy Dasgupta, Christos Papadimitriou, Umesh Vazirani . Boston [etc.] : McGraw Hill Higher Education, cop. 2008
3. Brassard, Gilles : Fundamentos de algoritmia / G. Brassard, T. Bratley ; traducción, Rafael García-Bermejo ; revisión técnica, Narciso Martí, Ricardo Peña, Luis Joyanes Aguilar . - 1ª ed. en español, reimp. Madrid [etc.] : Prentice Hall, 2008
4. Tardos, E. Algorithm Design / J. Kleinberg, E. Tardos Addison-Wesley, 2005
5. Parberry, I. Problems on Algorithms / I. Parberry and W. Gasarch. - 2nd ed. free book, 2002