

Técnicas de Diseño de Algoritmos

Código: 104393
Créditos ECTS: 6

| Titulación | Tipo | Curso | Semestre |
|---|------|-------|----------|
| 2503740 Matemática Computacional y Analítica de Datos | OB | 2 | 1 |

La metodología docente y la evaluación propuestas en la guía pueden experimentar alguna modificación en función de las restricciones a la presencialidad que impongan las autoridades sanitarias.

Contacto

Nombre: Francisco Javier Sánchez Pujadas

Correo electrónico: Javier.Sanchez.Pujadas@uab.cat

Uso de idiomas

Lengua vehicular mayoritaria: catalán (cat)

Algún grupo íntegramente en inglés: No

Algún grupo íntegramente en catalán: Sí

Algún grupo íntegramente en español: No

Prerequisitos

Aunque no es obligatorio, es muy aconsejable haber cursado las siguientes asignaturas:

- Iniciación a la Programación
- Algorismia y Combinatoria en Grafos. Métodos Heurísticos
- Programación Orientada a Objetos

Se considerará que el alumno ya ha adquirido los conocimientos impartidos en estas asignaturas.

Objetivos y contextualización

Partiendo de la base de que los alumnos tienen unos conocimientos básicos sobre programación y estructuras de datos, se pretende que el alumno sea capaz de analiza, diseñar e implementar algoritmos basándose en las técnicas de diseño de algoritmos existentes. Para cumplir con este objetivo, el alumno adquirirá los conocimientos sobre:

- Especificación formal de problemas. Cómo pasar de una descripción de un problema a una especificación válida para el desarrollo de un algoritmo que lo resuelva.
- Pruebas formales para validar programas. Diseño basado en contratos. Precondiciones, postcondiciones e invariantes.
- Análisis de complejidad.
- Paradigmas para el diseño de algoritmos. Greedy. recursividad, backtracking, branch & bound, programación dinámica, algoritmos probabilísticos, etc.

El desarrollo de un algoritmo empieza por formalizar el enunciado de un problema. A partir de este enunciado se diseña un algoritmo que solucione el problema, pero esto no es suficiente, hay que considerar cuánto tardará el algoritmo en darnos la solución. Así que nos interesa crear algoritmos lo más rápidos posibles. De esta forma podremos crear programas que solucionen problemas lo más grandes posibles en tiempos aceptables. Esta rapidez se consigue diseñando algoritmos que minimicen el número de operaciones a realizar para resolver un problema y desarrollando una implementación eficiente de las operaciones del algoritmo. Esto supondrá que en esta asignatura el alumno adquirirá conocimientos sobre algorítmica e implementación eficiente de algoritmos.

Competencias

- Diseñar, desarrollar y evaluar soluciones algorítmicas eficientes para problemas computacionales de acuerdo con los requisitos establecidos.
- Evaluar de manera crítica y con criterios de calidad el trabajo realizado.
- Que los estudiantes hayan demostrado poseer y comprender conocimientos en un área de estudio que parte de la base de la educación secundaria general, y se suele encontrar a un nivel que, si bien se apoya en libros de texto avanzados, incluye también algunos aspectos que implican conocimientos procedentes de la vanguardia de su campo de estudio.
- Que los estudiantes hayan desarrollado aquellas habilidades de aprendizaje necesarias para emprender estudios posteriores con un alto grado de autonomía.
- Que los estudiantes sepan aplicar sus conocimientos a su trabajo o vocación de una forma profesional y posean las competencias que suelen demostrarse por medio de la elaboración y defensa de argumentos y la resolución de problemas dentro de su área de estudio.
- Utilizar eficazmente bibliografía y recursos electrónicos para obtener información.

Resultados de aprendizaje

1. Evaluar de manera crítica y con criterios de calidad el trabajo realizado.
2. Evaluar y analizar la complejidad computacional de las soluciones algorítmicas para poder desarrollar e implementar aquella que garantice el mejor rendimiento.
3. Implementar soluciones recursivas a problemas de programación.
4. Que los estudiantes hayan demostrado poseer y comprender conocimientos en un área de estudio que parte de la base de la educación secundaria general, y se suele encontrar a un nivel que, si bien se apoya en libros de texto avanzados, incluye también algunos aspectos que implican conocimientos procedentes de la vanguardia de su campo de estudio.
5. Que los estudiantes hayan desarrollado aquellas habilidades de aprendizaje necesarias para emprender estudios posteriores con un alto grado de autonomía.
6. Que los estudiantes sepan aplicar sus conocimientos a su trabajo o vocación de una forma profesional y posean las competencias que suelen demostrarse por medio de la elaboración y defensa de argumentos y la resolución de problemas dentro de su área de estudio.
7. Seleccionar y utilizar las estrategias de programación apropiadas para la resolución de un problema dado.
8. Utilizar eficazmente bibliografía y recursos electrónicos para obtener información.

Contenido

El temario de la asignatura será:

- Especificación formal de algoritmos
- Complejidad Algorítmica
- Algoritmos Greedy
- Recursividad
- Backtracking
- Branch&bound
- Programación dinámica.
- Algoritmos Probabilísticos.

Metodología

Teniendo en cuenta que el objetivo final de la asignatura es que el alumnado sea capaz de analizar y diseñar algoritmos de forma eficiente según un problema dado, el trabajo del alumnado es el eje central de su aprendizaje, acompañado y guiado por el profesorado. Por este motivo, las clases presenciales serán altamente prácticas y se centrarán en que el alumnado consolide los conocimientos que son objetivo de aprendizaje de esta asignatura.

La metodología general de la asignatura se puede dividir en tres fases:

Preparación de la clase: el objetivo de esta fase es que el alumnado pueda aprender los conceptos que se trabajarán en la sesión siguiente mediante diversas actividades ofrecidas por el profesorado como puede ser el visionado de vídeos, la lectura de textos, etc.

Clase presencial: el objetivo de esta fase es la de consolidar los conceptos vistos y ponerlos en valor dentro del contexto de la asignatura. El profesorado velará para que el alumnado profundice en estos conceptos mediante ejercicios (más o menos) guiados durante la sesión. Así la clase empezará con una explicación resumida de los conceptos que el alumno habrá estudiado en la preparación de la clase. Este resumen dará una visión general sobre los conceptos en estudio y dará pie a que los alumnos puedan resolver las dudas que tengan. Luego se explicará el problema o trabajo práctica a realizar durante la segunda parte de la clase. Estos problemas o prácticas se realizarán con ordenador, ya que en muchos casos supondrán implementar algoritmos.

Trabajo autónomo: para que el alumnado tome soltura en la programación de los algoritmos vistos, este tendrá que hacer una parte del trabajo por su cuenta, ya sean ejercicios sueltos o dentro de un proyecto.

Para hacer más dinámico el aprendizaje basado en problemas y prácticas, se realizará un uso intensivo de correctores automáticos, siempre que sea posible. Así, para cada problema o práctica que tenga que entregar el alumno, se le proporcionará un corrector automático que le permitirá probar inmediatamente el programa que esté desarrollando. Así, el alumno sabrá si ha resuelto el problema o aún hay algún error que ha de solucionar. Además, para poner énfasis en la implementación óptima de algoritmos, el alumno podrá comparar los tiempos de ejecución de su solución con la proporcionada por el profesor. Así, se quiere gamificar los trabajos prácticos, marcando como objetivo y premiando superar la solución del profesor.

Para la realización de los problemas y prácticas se utilizará el lenguaje de programación C++ a nivel básico. Se selecciona este lenguaje por ser el lenguaje de alto nivel que obtiene el máximo rendimiento de los procesadores actuales, por ello, es especialmente apropiado para la implementación óptima de algoritmos.

Adaptación de la metodología a docencia no presencial

La docencia no presencial supondrá dar más importancia al trabajo autónomo del alumno e implica un cambio metodológico donde se aplicará la clase invertida. Considerando aquet concepto se planificará la docencia semanal como unos vídeos / pdfs con la materia teórica de la semana. De las 4h de docencia presencial se harán 2h de clase online. En estas clases se resolverán problemas y se avanzará en el desarrollo de la práctica. Al final de la semana se tendrá que hacer una entrega de problemas o prácticas para que el profesor pueda comprobar el seguimiento de la asignatura por parte del alumnado.

Preparación de la clase: Cada semana comenzará con el estudio o revisión del material que publicará el profesor en el campus virtual. Normalmente aquet material será una presentación hecha con powerpoint y un vídeo donde se comentarán las páginas de la presentación como si fuera una típica clase magistral de teoría. En aquet material se explicarán conceptos teóricos, se harán ejemplos de resolución de problemas y se propondrán problemas para las clases online y para entregar. El alumno deberá estudiar aquet material antes de las clases online.

Clase online: El profesor hará una pequeña introducción al problema / práctica que se tratará llevarán la clase. A partir de aquet momento se establecerá un diálogo donde los alumnos expresarán las dudas que tengan y propondrán como solucionar el problema / práctica. El profesor guiará a los alumnos hasta llegar a solucionar el problema a partir de las propuestas de los alumnos. En caso de que sea posible se implementarán en clase las propuestas de solución de los alumnos / profesor. La participación en clase se puntuará y servirá para mejorar la nota de la asignatura.

Trabajo autónomo: El trabajo autónomo será igual que en la metodología presencial.

Material necesario para la docencia no presencial

Para las clases online como por las consultas se utilizará la aplicación Microsoft Teams que debe estar instalada en el ordenador del alumno. Aquet ordenador debe tener cámara y micrófono para facilitar la

interacción en las clases y consultas. Para programar utilizará Visual Studio C ++ y el sistema operativo Microsoft Windows.

Actividades

| Título | Horas | ECTS | Resultados de aprendizaje |
|--|-------|------|---------------------------|
| Tipo: Dirigidas | | | |
| Clases presenciales (teoría, problemas y practicas) | 50 | 2 | 1, 2, 3, 4, 5, 6, 7, 8 |
| Tipo: Supervisadas | | | |
| Refuerzo y seguimiento en la resolución de los proyectos | 4 | 0,16 | 1, 2, 3, 4, 5, 6, 7, 8 |
| Seguimiento en la asimilación de los conceptos teóricos | 4 | 0,16 | 1, 2, 3, 4, 5, 6, 7, 8 |
| Tipo: Autónomas | | | |
| Elaboración informes prácticos | 8 | 0,32 | 1, 2, 3, 4, 5, 6, 7, 8 |
| Preparación previa a las clases | 48 | 1,92 | 1, 2, 3, 4, 5, 6, 7, 8 |
| preparación exámenes parciales | 12 | 0,48 | 1, 2, 3, 4, 5, 6, 7, 8 |
| trabajo autónomo | 12 | 0,48 | 1, 2, 3, 4, 5, 6, 7, 8 |

Evaluación

Criterios e indicadores de evaluación:

- Comprensión de los conceptos teóricos de la asignatura.
- Implementación correcta y eficiente de algoritmos.

Actividades e instrumentos de evaluación:

- Nota Teoría: corresponde a dos exámenes parciales de teoría sobre la asignatura. Se han aprobar los exámenes parciales para eliminar materia en el examen de recuperación.
- Nota de problemas: Corresponde a la nota obtenida por una serie de problemas que tendrá que entregar el alumno. No hay recuperación para las entregas de problemas.
- Nota Prácticas: En este apartado hay una nota de grupo y una individual:
 - Nota de grupo: Corresponde a la nota obtenida a las entregas por grupos de la práctica. Las entregas de prácticas suspendidas se puedes reucuperar en entregas posteriores.
 - Nota individual: Es la puntuación obtenida en el examen de prácticas. Habrá recuperación del examen de prácticas.

La nota final de la asignatura se obtiene combinando la evaluación de estas tres actividades de la siguiente manera:

Si Nota Teoría ≥ 5 y Nota Prácticas ≥ 5 ENTONCES

Nota Final = $0,4 * \text{Nota Teoría} + 0,2 * \text{Nota problemas} + 0,4 * \text{Nota Prácticas}$

SINO Nota Final = $\text{MIN}(\text{Nota Teoría}, \text{Nota Prácticas})$

SI Nota examen primer parcial ≥ 5 y Nota examen segundo parcial ≥ 5 ENTONCES

Nota Teoría = $0,5 * \text{Nota examen primer parcial} + 0,5 * \text{Nota examen segundo parcial}$

SINO Nota Teoría=MIN(Nota examen primer parcial, Nota examen segundo parcial)

Nota problemas = Media ponderada de los problemas entregados.

SI Nota Individual \geq 5 y Nota Grupo \geq 5 ENTONCES

Nota Prácticas = 0,2 * Nota Individual + 0,8 * Nota Grupo

SINO Nota Prácticas =MIN(Nota Individual, Nota Grupo)

Nota Individual = Examen de prácticas.

Nota Grupo = Media ponderada de las entregas de prácticas si todas están aprobadas, sino el mínimo de las notas de las entregas.

Convalidación de prácticas: No se convalidan prácticas de años anteriores.

Recuperación de prácticas: En el caso de haber suspendido una entrega de grupo, se podrá recuperar en las siguientes entregas de la práctica. La nota será $0.75 * (\max(\text{nota recuperación, nota entrega suspendida}) - \text{Nota entrega suspendida}) + \text{nota entrega suspendida}$. En el caso de suspender el examen de prácticas, el alumno deberá presentar a un examen de recuperación de prácticas el mismo día del examen de recuperación final.

Condiciones para aprobar la asignatura:

Para aprobar la asignatura se ha de haber aprobado (nota mayor o igual a 5):

- Nota final
- Nota teoría.
- Nota prácticas.
- Notas entregas de prácticas.
- Exámenes.

Condiciones por no evaluable:

No tener ninguna parte de la asignatura suspendida.

Condiciones para el suspenso:

No alcanzar una nota media superior o igual a 5.

Suspender alguna de las actividades de evaluación de la asignatura, aunque la media supere el 5. En este caso, la nota será la nota mínima obtenida de alguna de las partes (exámenes o prácticas).

Condiciones para la matrícula de honor:

La matrícula de honor se puede conseguir con una nota media superior o igual a 9,0.

Debido a que hay un número limitado de matrículas de honor que se pueden dar por grupo, se otorgarán por orden de nota de mayor a menor.

Prácticas, trabajos o exámenes copiados:

Sin perjuicio de otras medidas disciplinarias que se estimen oportunas, y de acuerdo con la normativa académica vigente, las irregularidades cometidas por un estudiante que puedan conducir a una variación de la calificación se calificarán con un cero (0). Las actividades de evaluación calificadas de esta forma y por este procedimiento no serán recuperables. Si es necesario superar cualquiera de estas actividades de evaluación para aprobar la asignatura, esta asignatura quedará suspendida directamente, sin oportunidad de recuperarla en el mismo curso. Estas irregularidades incluyen, entre otros:

- la copia total o parcial de una práctica, informe, o cualquier otra actividad de evaluación;

- dejar copiar;
- presentar un trabajo de grupo no hecho íntegramente por los miembros del grupo;
- presentar como propios materiales elaborados por un tercero, aunque sean traducciones o adaptaciones, y en general trabajos con elementos no originales y exclusivos del estudiante;
- tener dispositivos de comunicación (como teléfonos móviles, smart watches, etc.) accesibles durante las pruebas de evaluación teórico-prácticas individuales (exámenes).

En caso de que el estudiante haya cometido irregularidades en un acto de evaluación, la nota numérica del expediente será el valor menor entre 3.0 y la nota que le correspondería según el método de evaluación de la asignatura (y por tanto no será posible el aprobado por compensación).

En resumen: copiar, dejar copiar o plagiar en cualquiera de las actividades de evaluación equivale a un SUSPENSO

con nota inferior a 3.0.

Publicación de notas, fechas de exámenes, etc:

Las fechas de evaluación continua y entrega de trabajos se publicarán en el campus virtual y pueden estar sujetos a cambios de programación por motivos de adaptación a posibles incidencias. Siempre se informará en el campus virtual sobre estos cambios ya que se entiende que es el mecanismo habitual de intercambio de información entre profesor y estudiantes.

Procedimiento de revisión de las calificaciones

Para cada actividad de evaluación, se indicará un lugar, fecha y horade revisión en la que el estudiante podrá revisar la actividad con el profesor. En este contexto, se podrán hacer reclamaciones sobre la nota de la actividad, que serán evaluadas por el profesorado responsable de la asignatura. Si el estudiante no se presenta en esta revisión, no se revisará posteriormente esta actividad.

Actividades de evaluación

| Título | Peso | Horas | ECTS | Resultados de aprendizaje |
|---|----------------------------|-------|------|---------------------------|
| Entrega de ejercicios | Ver descripción evaluación | 1 | 0,04 | 1, 2, 3, 4, 5, 6, 7, 8 |
| Evaluación grupal del proyecto | Ver descripción evaluación | 2 | 0,08 | 1, 2, 3, 4, 5, 6, 7, 8 |
| Evaluación individual del proyecto | Ver descripción evaluación | 1 | 0,04 | 1, 2, 3, 4, 5, 6, 7, 8 |
| Examen Final (recuperación) | Ver descripción evaluación | 4 | 0,16 | 1, 2, 3, 4, 5, 6, 7, 8 |
| Primer Examen Parcial Teórico-Práctico | Ver descripción evaluación | 2 | 0,08 | 1, 2, 3, 4, 5, 6, 7, 8 |
| Segundo Examen Parcial Teórico-Práctico | Ver descripción evaluación | 2 | 0,08 | 1, 2, 3, 4, 5, 6, 7, 8 |

Bibliografía

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

Técnicas de Diseño de Algoritmos, Rosa Guerequeta y Antonio Vallecillo.

Técnicas de diseño de algoritmos, F. Perales, M. Mascaró. Universitat de les Illes Balears

Thinking in C++ 2nd Edition by Bruce Eckel Volume 1 y Volume 2.

(<http://www.mindview.net/Books/TICPP/ThinkingInCPP2e.html>)

El lenguaje de programación C++, Bjarne Stroustrup

<http://en.cppreference.com/w/>

<http://www.cplusplus.com/>