



Curso
2018-19

Algorítmica

Grado en Ingeniería Informática
2º curso (Grupos B y D)

José Luis Verdegay

Depto. de Ciencias de la Computación
e Inteligencia Artificial



¿Qué van a aprender en esta asignatura?

- Técnicas de resolución de problemas
 - ¿Cómo un GPS es capaz de darnos cualquier ruta?
 - ¿Cómo se hacen las pruebas de ADN?
 - ¿Cómo una empresa de transportes planifica los miles de envíos de paquetes para que lleguen al destino en el menor tiempo?
 - ¿Cómo apilar paquetes para que ocupen el mínimo espacio y se gaste el mínimo material de embalaje (cartón, papel, etc.)?
 - ¿Cómo una empresa es capaz de asignar a todos sus trabajadores para realizar tareas diferentes de forma óptima?
 - ¿Cómo se diseñan las rutas de los autobuses urbanos en la ciudad?
 - ¿Cómo coordinar los semáforos de los cruces?
 - ¿Qué carreteras hay que construir en una nueva ciudad para optimizar el transporte y conexiones de la misma?
 - ¿Cómo se diseña una red local minimizando el cableado?



Abstraer a partir de ejemplos

- La araña hambrienta y la mosca
- En una habitación cerrada de 12 pies de ancho, 12 pies de alto y 30 pies de largo, en el medio de una de las paredes cuadradas y a 1 pie del techo, hay una araña hambrienta. A su vez, en el medio de la pared cuadrada opuesta y a 1 pie del piso, se encuentra dormida una mosca.
- La araña (mutación defectuosa de la que le otorgó los poderes a Peter Parker convirtiéndolo en Spiderman) no cuenta entre sus poderes con la capacidad de hacer hilos, por lo tanto no se puede colgar y balancear para alcanzar más pronto su objetivo: la mosca. No tiene mas remedio que caminar y recorrer las paredes para llegar a su almuerzo.
- Si la araña camina a una velocidad de 5 pies por hora ¿Cuánto tiempo puede dormir la mosca sin preocuparse?



¿Qué van a aprender en esta asignatura?

- Técnicas de resolución de problemas
 - ¿Cómo un GPS es capaz de darnos cualquier ruta?
 - ¿Cómo se hacen las pruebas de ADN?
 - ¿Cómo una empresa de transportes planifica los miles de envíos de paquetes para que lleguen al destino en el menor tiempo?
 - ¿Cómo apilar paquetes para que ocupen el mínimo espacio y se gaste el mínimo material de embalaje (cartón, papel, etc.)?
 - ¿Cómo una empresa es capaz de asignar a todos sus trabajadores para realizar tareas diferentes de forma óptima?
 - ¿Cómo se diseñan las rutas de los autobuses urbanos en la ciudad?
 - ¿Cómo coordinar los semáforos de los cruces?
 - ¿Qué carreteras hay que construir en una nueva ciudad para optimizar el transporte y conexiones de la misma?
 - ¿Cómo se diseña una red local minimizando el cableado?

Hay un programa que lo hace. Pero, ¿cómo y por qué funciona ese programa?

ALGORITMIA

The image is a digital-themed composition. In the center, a laptop is open, and its screen displays a glowing, wireframe globe with a bright light source behind it, creating a lens flare effect. The globe is set against a dark background on the screen. Surrounding the laptop and filling the entire frame are various digital motifs: streams of binary code (0s and 1s) flowing like rivers, abstract geometric shapes, and a blue-toned background with a grid-like pattern. The overall aesthetic is futuristic and technological.



Los algoritmos están de moda

https://www.quora.com/profile/Andrew-McGregor-12

Quora

Search for questions, people, and topics

Buscar

Ask New Question

Sign In

Buscar usando Google

Un algoritmo completa la misteriosa 'Sinfonía inacabada' de Schubert... https://elpais.com/cultura/2019/02/04/actualidad/1549284459_0...

CULTURA

Un algoritmo completa la misteriosa 'Sinfonía inacabada' de Schubert

Varios músicos cuestionan la iniciativa de Huawei por artificial y carente de alma

JESÚS RUÍZ MANTILLA

Madrid - 4 FEB 2019 - 20:27 CET



Retrato sin datar de Schubert. DEA / A. DAGLI ORTI (DE AGOSTINI/GETTY IMAGES)

Franz Schubert es un perpetuo enigma. Murió joven, a los 31 años, pero más que desgastarse en una vida intensa, como tantos otros músicos, se fue consumiendo en cierta melancolía solitaria de cafés como desahogo a su total entrega a la creación. "El Estado debería ocuparse de mí", le escribió a un amigo, para dar cuenta de su devoción al trabajo. Y si corrieran estos tiempos, quizás lo hiciera la compañía tecnológica [Huawei](#), que ha diseñado mediante algoritmo e inteligencia artificial el remate de su *Sinfonía número ocho*, conocida como la *Inacabada*. Hacía la penúltima en su catálogo de obras orquestales. En la Viena romántica de principios del XIX, Schubert llegó a crear

I want to work at Google. What course(s) should I take and what should I do after college?



Andrew McGregor, Site Reliability Engineer at Google (2013-present)

Answered Oct 8 · Upvoted by Ray Farias, AdMob Full-Stack Software Engineer at Google (2016-present)

Don't do just CS. Sure, you do need to be a really good programmer, which means doing more than just your coursework in it, you need to really understand the material. You need good to great grades... [\(more\)](#)

What is the reason that giant companies (e.g., Google or Microsoft) ask typical question like binary search tree or traditional algorithm or question like algorithm complexity? What is the purpose? Most of them are not used in real world life.



Andrew McGregor, Site Reliability Engineer at Google (2013-present)

Answered Oct 8 · Upvoted by Bruce R. Miller, works at Google and David Castro, Technical Writer at Google (2016-present)

One of the things the interviewers are doing is trying to figure out if each candidate has the aptitude for what Google calls SRE (Site Reliability Engineering). Now, the average software engineer doesn't do much algorithmic work... but SREs do, and need to understand what is going on there. Poor worst case behaviour is the cause of so many outages and issues in production systems. Often a "Query of Death" is a worst-case problem, and changing to an algorithm with better worst case behaviour is the right answer.

The point is, knowing that can happen, and how to figure out that's what got you, in the midst of debugging a production emergency, is something that happens in the real world, and Google especially needs to find the people who can do it.

Even better is having a software engineering organisation full of people who won't make that mistake in the first place.

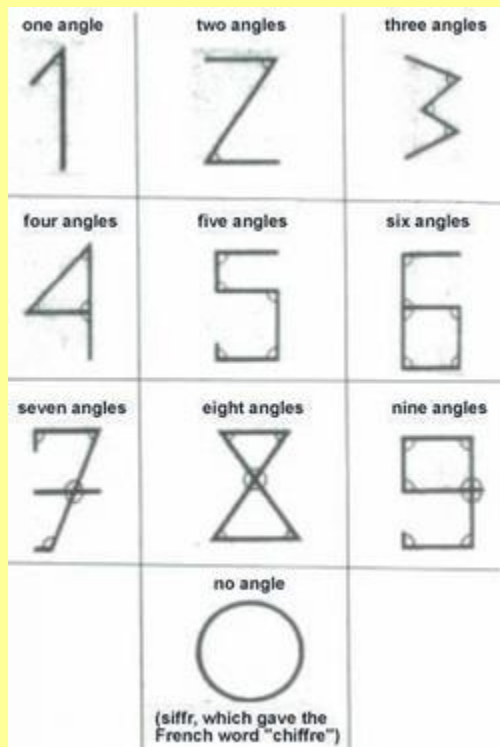
Don't get me wrong, it's not every day you need that knowledge... but about once a quarter, you do. I spent a couple of months this March-April doing that sort of work, to get a system sped up to the point it could cleanly handle production workload again.





Pero ¿son algo reciente?

La **Algorítmica** es la Ciencia que estudia los Algoritmos, y es uno de los pilares de la **Informática** (y de la Inteligencia Artificial)



فأما الأموال والجذور التي تعدل العدد فمثل قولك
مال وعشرة أجزاره يعدل تسعة وثلاثين درهما ومعناه أي مال اذا زدت عليه مثل
عشرة أجزاره بلغ ذلك كله تسعة وثلاثين . فبابه ^(١) أن تنصف الأجزاء وهي في
هذه المسئلة خمسة فتضربها في مثلها فتكون خمسة وعشرين فتزيدها على التسعة
والثلاثين فتكون أربعة وستين فتأخذ جذرها وهو ثمانية فتقص منه نصف
الأجزاء هو خمسة فيبقى ثلاثة وهو جذر المال الذي تريد والمال تسعة .

Texto de una edición árabe de álgebra (El Cairo, 1968)
donde se explica como resolver la ecuación $x^2 + 10x = 39$

Forma de escribir los números: Basada en los ángulos que tiene cada gráfica.



Índice

- Horario
- Competencias específicas de la asignatura
- Objetivos perseguidos
- Temario
- Programa detallado
- Metodología
- Sistema de evaluación
- Bibliografía
 - Básica,
 - De apoyo y consulta
- Forma de aprobar
- Material de ayuda





Horarios de clases y tutorías

2º B Grado en Ingeniería Informática 2º cuatrimestre															
	Lunes		Tutorías CCIA #7	Martes		Miércoles		Jueves		Viernes					
8:30-9:30															
9:30-10:30	AC 0.1					IA 0.1		ALG 0.1		FIS 0.1					
10:30-11:30	ALG 0.1					IA 0.1		ALG 0.1		FIS 0.1					
11:30-12:30	IA (B1) 2.7	ALG (B2) 2.6		AC (B3) 3.7			AC (B2) 3.7	FIS (B3) 2.7	FBD (B1) 2.6	AC (B1) 3.7	IA (B2) 2.7	FBD (B3) 2.6	FBD (B2) 2.6	FIS (B1) 3.7	IA (B3) 2.7
12:30-13:30	IA (B1) 2.7	FBD (B2) 2.6		AC (B3) 3.7			AC (B2) 3.7	FIS (B3) 2.7	FBD (B1) 2.6	AC (B1) 3.7	IA (B2) 2.7	FBD (B3) 2.6	FBD (B2) 2.6	FIS (B1) 3.7	IA (B3) 2.7
13:30-14:30															
2º D Grado en Ingeniería Informática 2º cuatrimestre															
15:30-16:30	AC 0.1		ALG 0.1		IA 0.1		ALG 0.1		FIS 0.1						
16:30-17:30	FBD 0.1		AC 0.1		IA 0.1		ALG 0.1		FIS 0.1						
17:30-18:30	ALG (D1) 2.6	AC (D2) 2.8	AC (D1) 2.3	FIS (D2) 2.6	FIS (D1) 2.7	ALG (D2) 2.6	IA (D1) 2.7	FBD (D2) 2.6	FBD (D1) 2.6	IA (D2) 2.7					
18:30-19:30	FBD (D1) 2.6	AC (D2) 2.8	AC (D1) 2.3	FIS (D2) 2.6	FIS (D1) 2.7	FBD (D2) 2.6	IA (D1) 2.7	FBD (D2) 2.6	FBD (D1) 2.6	IA (D2) 2.7					
19:30-20:30															
20:30-21:30															

El 7 de marzo (jueves) las dos horas de ALG serán de I.A. El 13 de marzo (miércoles), las dos horas de I.A. serán de ALG



Competencias específicas, básicas y transversales

LIBRO BLANCO

TÍTULO DE GRADO
EN INGENIERÍA
INFORMÁTICA

Agencia Nacional de Evaluación
de la Calidad y Acreditación

- R6:** Conocimiento y aplicación de los procedimientos algorítmicos básicos de las tecnologías informáticas para diseñar soluciones a problemas, analizando la idoneidad y complejidad de los algoritmos propuestos
- CB4:** Que los estudiantes puedan transmitir información, ideas, problemas y soluciones a un público tanto especializado como no especializado.
- T2:** Capacidad para tomar decisiones basadas en criterios objetivos (datos experimentales, científicos o de simulación disponibles) así como capacidad de argumentar y justificar lógicamente dichas decisiones, sabiendo aceptar otros puntos de vista.



Objetivos (resultados del aprendizaje)

- Plantearse la búsqueda de varias soluciones distintas para un mismo problema y evaluar la bondad de cada una de ellas.
- Tomar conciencia de la importancia del análisis de la eficiencia de un algoritmo como paso previo a su implementación en un lenguaje de programación.
- Conocer la notación asintótica para describir la eficiencia de un algoritmo, distinguiendo entre los distintos tipos de análisis que se pueden realizar: caso más favorable, más desfavorable y promedio.
- Saber realizar el análisis de eficiencia de un algoritmo, tanto a nivel teórico como empírico, y saber contrastar resultados experimentales con los teóricos.
- Conocer las técnicas básicas de resolución de ecuaciones de recurrencia.
- Comprender la técnica de resolución de un problema por división en problemas más pequeños.
- Conocer y saber aplicar los esquemas básicos de los algoritmos divide y vencerás.
- Comprender la técnica “greedy” de resolución de problemas y los distintos casos que se pueden presentar en la resolución de problemas por esta técnica: obtención de la solución óptima, de una solución no óptima, o no obtención de la solución.



Objetivos (resultados del aprendizaje)

- Comprender la técnica de resolución de problemas por programación dinámica, e identificar las diferencias con divide y vencerás y con avance rápido.
- Saber identificar problemas que cumplen el Principio del Óptimo y qué es necesario para poder aplicar esta técnica.
- Saber ver al árbol de estados como una representación lógica del conjunto de todas las posibles soluciones de un problema.
- Conocer las técnicas de exploración de grafos (vuelta atrás y ramificación y poda) y su aplicación en la resolución de problemas, entendiendo sus características principales y las diferencias entre ellas.
- Comprender y saber aplicar el uso de cotas para reducir el espacio de búsqueda en las técnicas de exploración en grafos.
- Conocer los criterios de aplicación de cada una de las distintas técnicas de diseño de algoritmos.
- Saber seleccionar e implementar el mejor algoritmo que resuelve un problema dado.



Temario

1

La eficiencia de los algoritmos

2

Algoritmos Divide y Vencerás

3

Algoritmos de Avance Rápido ("Greedy")

4

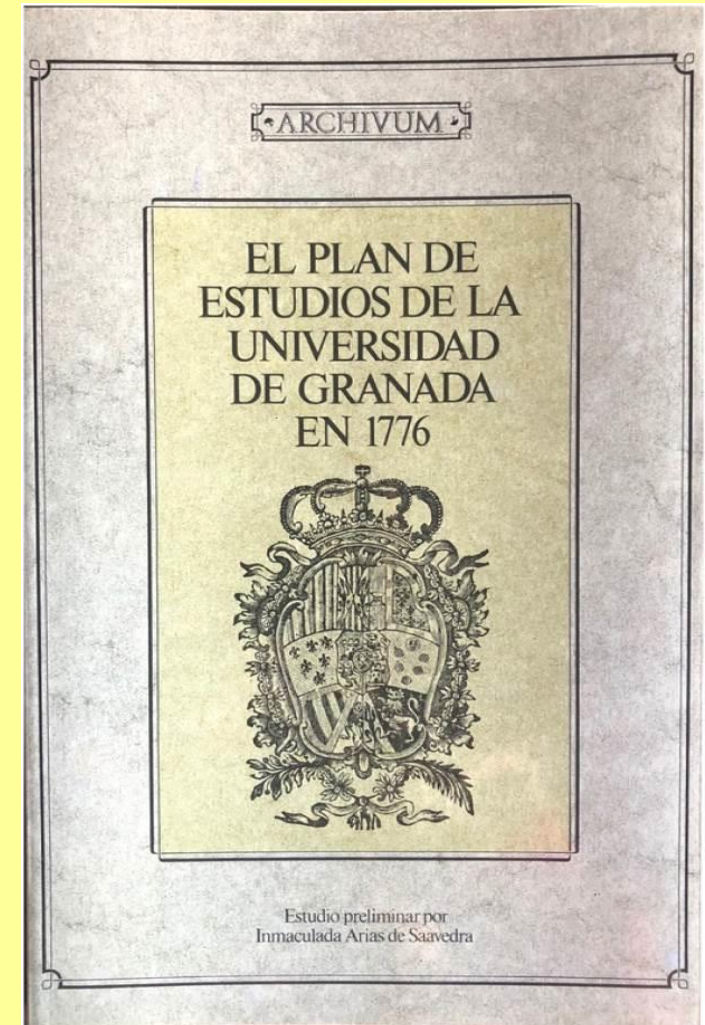
Programación Dinámica

5

Algoritmos para la exploración de grafos

6

Otras metodologías algorítmicas





Programa de la asignatura

Capítulo 1. La Eficiencia de los Algoritmos.

Tema 1. Planteamiento general.

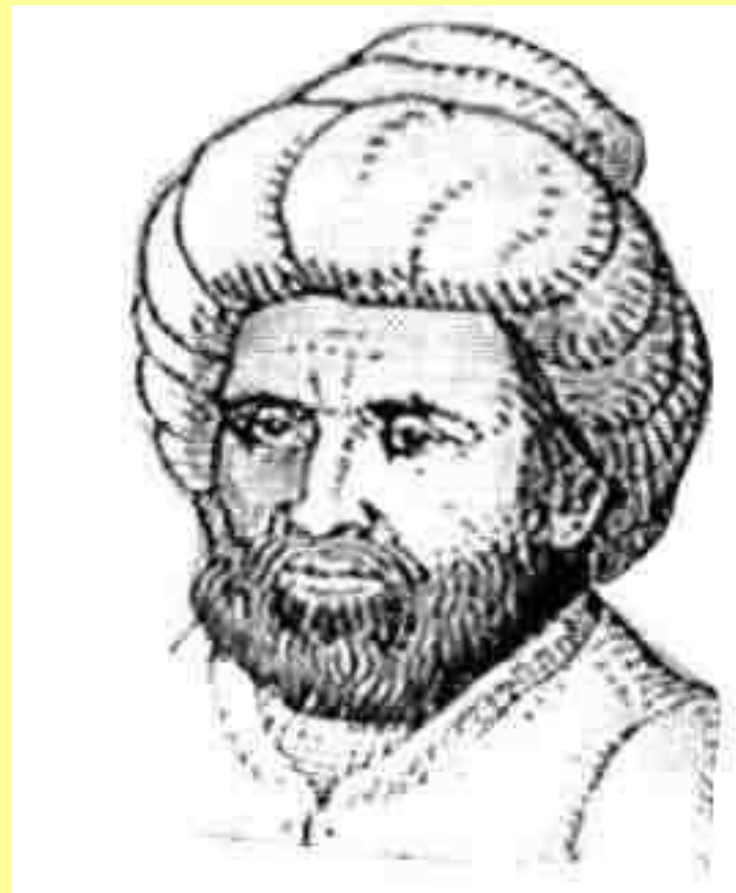
Ciencia de la Computación. El concepto de algoritmo. Problemas y casos. Distintos tipos de casos.

Tema 2: Tiempo de ejecución. Notaciones para la Eficiencia de los Algoritmos

La eficiencia de los algoritmos. Notaciones O y Ω . Análisis teórico y práctico del tiempo de ejecución de un algoritmo. Algunos ejemplos prácticos.

Tema 3. Resolución de Recurrencias Asintóticas

Resolución de recurrencias: Método de la función característica.





Programa



Capítulo 2. Algoritmos Divide y Vencerás.

Tema 4. Método general DV

El método general. Determinación del umbral. Algoritmos de búsqueda. Algoritmos de ordenación. Otras aplicaciones

Tema 5. Algoritmos de Búsqueda y Ordenación.

Algoritmos de búsqueda. Algoritmos de ordenación. Ordenación por mezcla. Quicksort. Discusión del método

Tema 6. Aplicaciones de la Técnica Divide y Vencerás.

Algoritmos para la multiplicación de enteros, de polinomios y de matrices: Algoritmo de Strassen. El problema del "skyline"



Programa

Capítulo 3. Algoritmos Greedy.

Tema 7. El Enfoque Greedy. Fundamentos teóricos.

Algoritmos greedy. Repaso de conceptos básicos de grafos

Tema 8. Algoritmos Greedy para Grafos.

Algoritmos para el árbol generador minimal: Algoritmos de Kruskal y Prim. Algoritmos para caminos mínimos: Algoritmo de Dijkstra.

Tema 9. Algoritmos greedy como heurísticas

Problemas de coloreo, del Viajante de Comercio y de la Mochila: Algoritmo de Martello y Toth.





Programa



Capítulo 4. Algoritmos Basados en Programación Dinámica.

Tema 10. Programación Dinámica y Principio del Óptimo.

El enfoque de la Programación Dinámica. El Principio del Óptimo de Bellman. La P.D. frente a otros enfoques

Tema 11. Algoritmos Elementales que usan Programación Dinámica.

El Problema del Camino Mínimo: Algoritmo de Floyd. El Problema de la Mochila. El Problema del Viajante de Comercio.

Tema 12. Aplicaciones de la Programación Dinámica

Multiplicación encadenada de matrices. El enfoque tabular de la PD



Programa

Capítulo 5. Algoritmos para la Exploración de Grafos.

Tema 13. Grafos Implícitos.

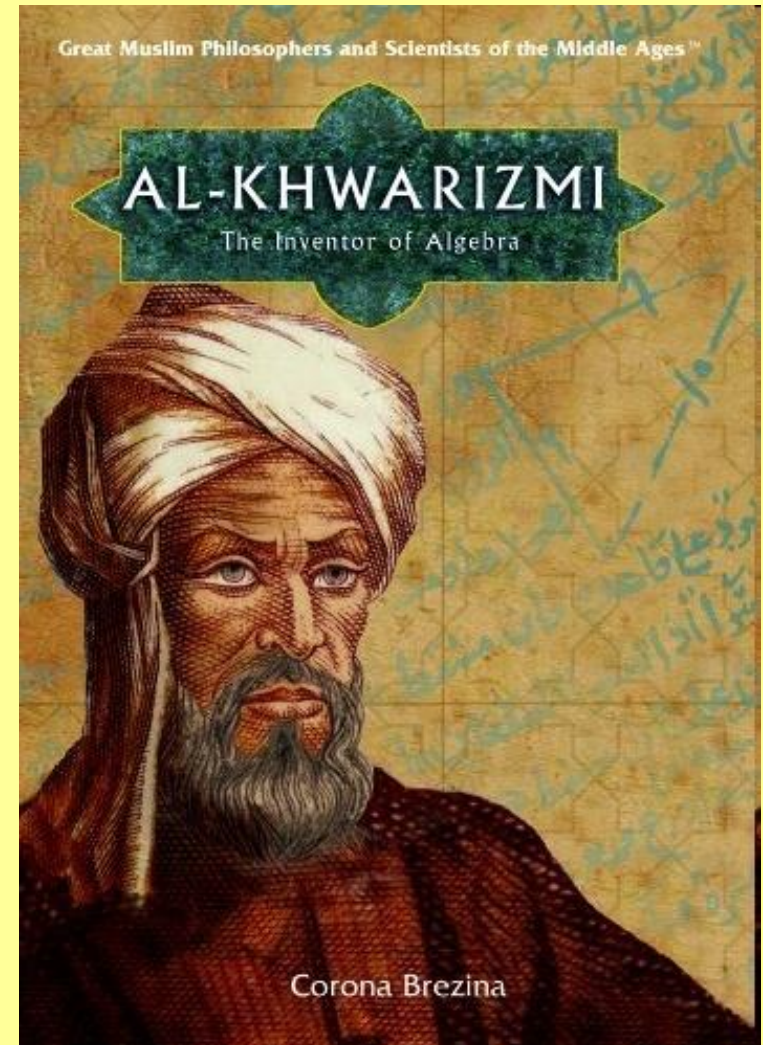
Grafos Y/O. Árboles para Juegos. Algoritmos de solución para juegos elementales.

Tema 14. La técnica "Backtracking"

El método "Backtracking". Espacio de estados. Algoritmo general Backtracking. El problema de las 8 reinas. El problema de la suma de subconjuntos. Circuitos Hamiltonianos.

Tema 15. El método "Branch and Bound"

Método general "Branch and Bound". El Problema de la Mochila. Problema del Viajante de Comercio.





Programa



Capítulo 6. Otras Metodologías Algorítmicas.

Tema 16. Algoritmos de Precomputación

Algoritmos numéricos para la evaluación de polinomios, para la multiplicación de matrices y para la resolución de sistemas de ecuaciones lineales.

Tema 17. Algoritmos de Transformación Algebraica

La Transformada de Fourier.
Algoritmos de cálculo.



Metodología

- **Lecciones magistrales** (Clases teóricas-expositivas) (grupo grande)
 - 45 horas presenciales (1.8 ECTS)
 - Competencias: R6, E8, E9, CB4, T2
- **Actividades prácticas** (Clases prácticas de laboratorio y seminarios) (grupo pequeño)
 - 12 horas presenciales (0.48 ECTS)
 - Competencias: R6, E8, E9, CB4, T2
- **Tutorías académicas** (grupo pequeño)
 - 3 horas presenciales (0.12 ECTS)
 - Competencias: R6, E8, E9, CB4, T2
- **Actividades no presenciales individuales** (Estudio y trabajo autónomo)
 - Competencias: R6, E8, E9, CB4, T2
- **Actividades no presenciales grupales** (Estudio y trabajo en grupo)
 - Competencias: R6, E8, E9, CB4, T2





Sistema de evaluación

- **Convocatoria Ordinaria:**

- La evaluación de la asignatura se divide en dos bloques fundamentales: Parte Teórica (**T**) y Parte Práctica (**P**):
- Para la parte teórica se realizará un examen final y, opcionalmente, eventuales entregas y presentaciones de ejercicios, o trabajos propuestos por el profesor. La ponderación de este bloque es del 70% (**T**)
- Para la parte práctica se realizarán prácticas de laboratorio, resolución de problemas y desarrollo de proyectos (individuales o en grupo), y se valorarán las entregas de los informes/memorias y presentaciones realizados por los alumnos. La ponderación de este bloque es del 30% (**P**)
- La calificación global (**G**) es la calificación numérica obtenida mediante la suma de las anteriores calificaciones (**G = T+P**).

- **Regimen de Asistencia**

- La asistencia a las clases teóricas no será obligatoria, aunque la participación activa en clase y la entrega de ejercicios planteados por el profesor se tendrá en cuenta dentro del sistema de evaluación continua de la asignatura.
- Se requerirá, siguiendo el sistema de evaluación continua, que el estudiante asista al menos a alguna de las sesiones prácticas dentro de los límites de entrega de cada práctica y eventualmente defienda ante el profesor el resultado de la correspondiente práctica.



Sistema de evaluación

- **Convocatorias Extraordinarias:**

- Esta modalidad de evaluación se realizará en un único acto académico en la fecha establecida por el Centro y consistirá en un examen escrito (evaluado de 0 a 10) que incluirá preguntas tanto de tipo teórico como práctico que garanticen que el alumno ha adquirido la totalidad de las competencias descritas en esta misma guía docente. La valoración de los bloques de teoría y de prácticas se mantiene en el margen del 70% para teoría y 30% para prácticas. Adicionalmente, para aquellos alumnos que superen la parte práctica por evaluación continua, pero no la parte teórica, se mantendrá la calificación obtenida en prácticas sólo para la convocatoria extraordinaria inmediatamente posterior a la ordinaria.

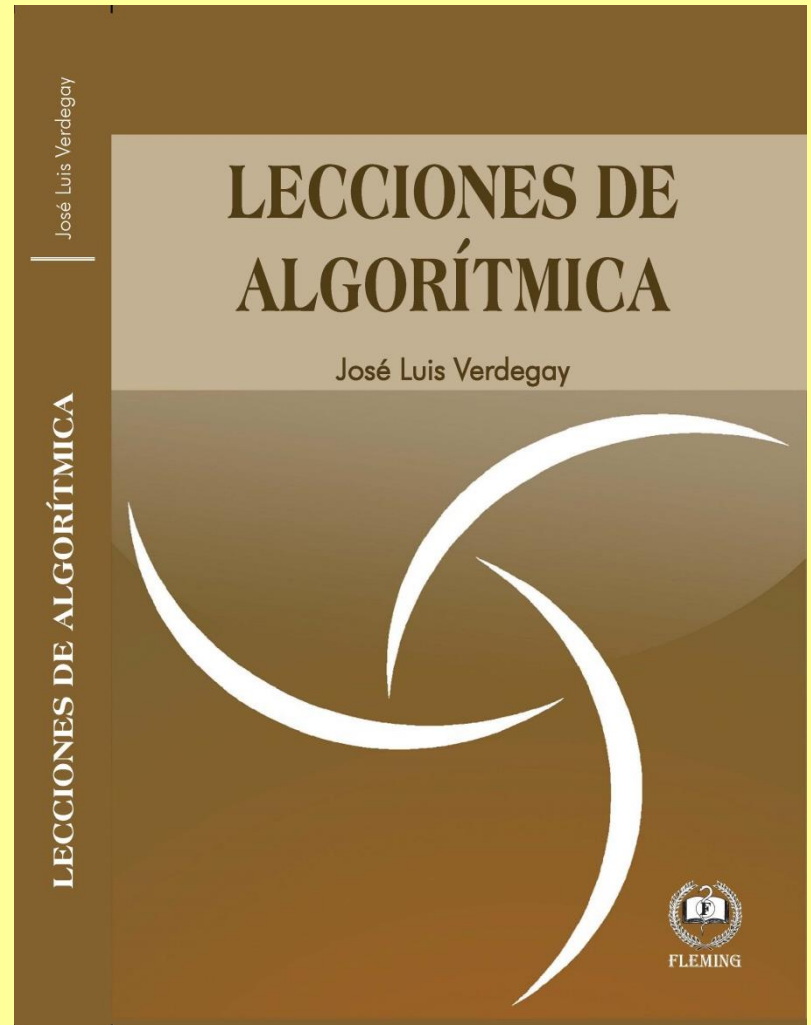
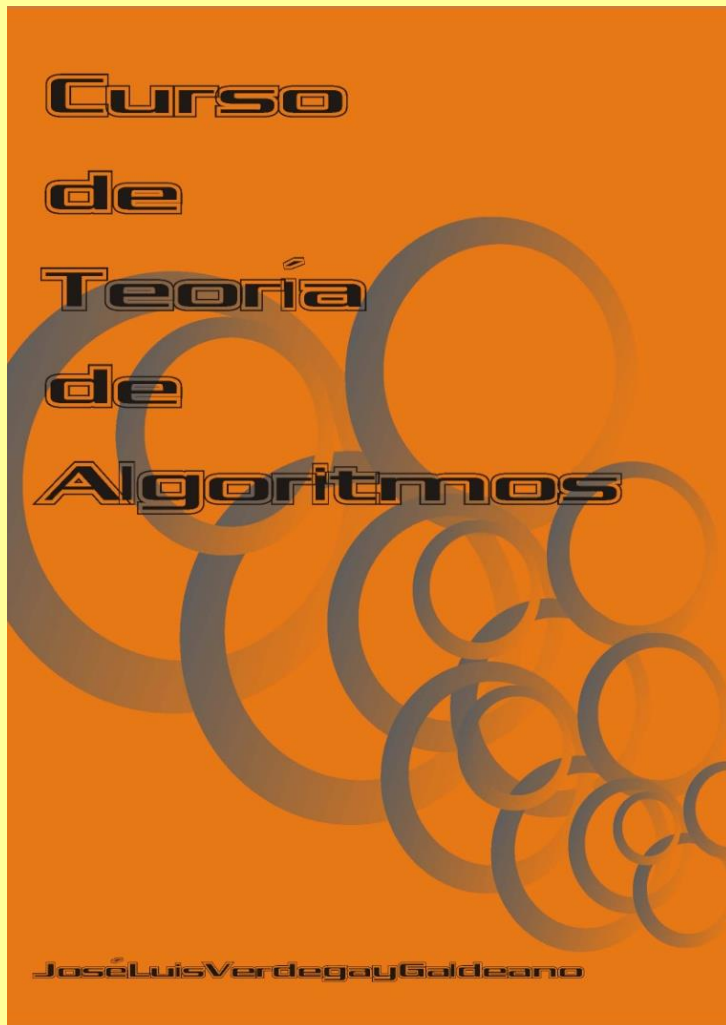
- **Evaluación única final**

- Esta modalidad de evaluación se realizará en un único acto académico en la fecha establecida por el Centro y consistirá en un examen escrito (evaluado de 0 a 10) que incluirá preguntas tanto de tipo teórico como práctico que garanticen que el alumno ha adquirido la totalidad de las competencias descritas en esta misma guía docente. La valoración de los bloques de teoría y de prácticas será del 70% para teoría y de 30% para prácticas.

Conforme a lo establecido en la normativa de Evaluación y de Calificación de los estudiantes de la UGR.

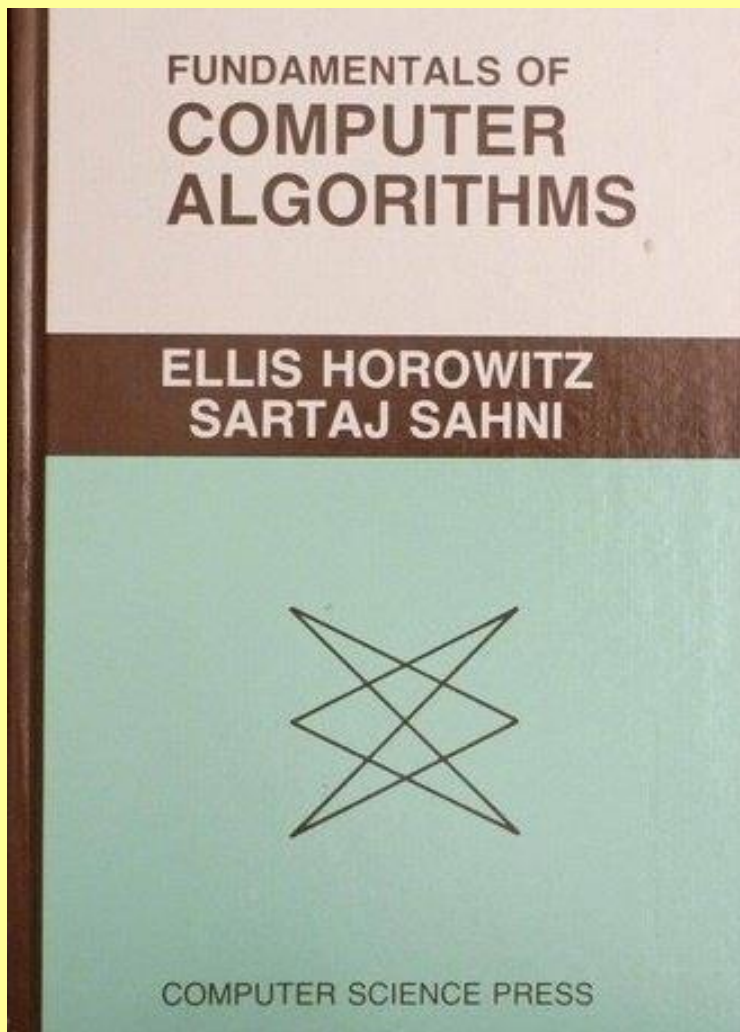


Bibliografía básica





Bibliografía de apoyo



G. Brassard y P. Bratley: Fundamentos de Algoritmos. Prentice Hall

<https://docs.google.com/file/d/0B4tyWbCUB5eFaVpx0XI3MLJtMzg/edit?pref=2&pli=1>

T.H. Cormen, C.E. Leiserson y R. L. Rivest: Introduction to Algorithms. The MIT Press

<http://is.ptithcm.edu.vn/~tdhuy/Programming/Introduction.to.Algorithms.pdf>

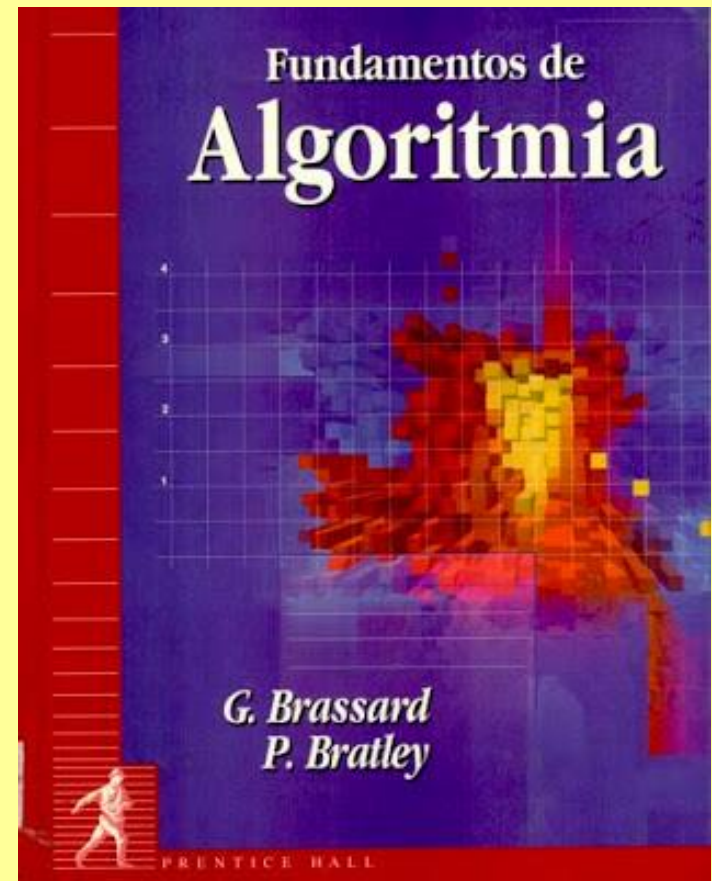
E. Horowitz y S. Sahni: Fundamentals of Computer Algorithms. Comp Science Press

<http://gvpcse.helplena.co/pdf/sahni.pdf>



Bibliografía de consulta

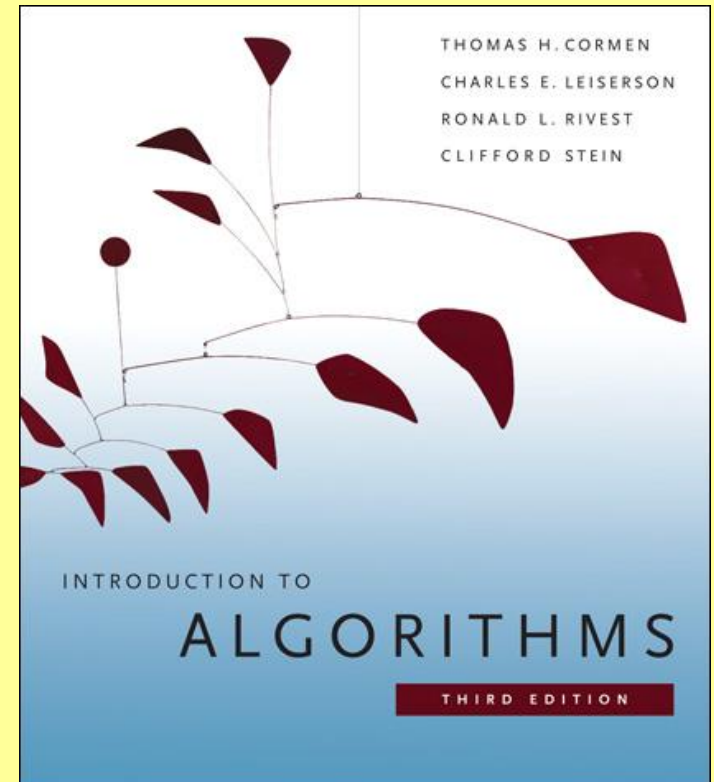
1. M.H. Alsuwaiyel: Algorithms. Design Techniques and Analysis. World Sci. (2000)
2. A. V. Aho, J.E. Hopcroft y J.D. Ullman: Data Structures and Algorithms. Addison Wesley (1982)
3. S. Baase: Computer Algorithms. Introduction to Design and Analysis. Addison Wesley (1988)
4. G. Brassard y P. Bratley: Fundamentos de Algoritmia. Prentice Hall (1997)
5. B. Christian and T. Griffiths. Algorithms to Live. The Computer Science of Human Decisions. HarperCollins Pub. Ltd. (2016)
6. Dasgupta, Papadimitriou y Vazirani: Algorithms. McGraw-Hill Science/Engineering (2006)
7. D. Harel: Algorithmics (The spirit of computing). Addison Wesley Pub. Co. (1987)
8. S. Skiena: The Algorithm Design Manual. Springer Verlag (1998).



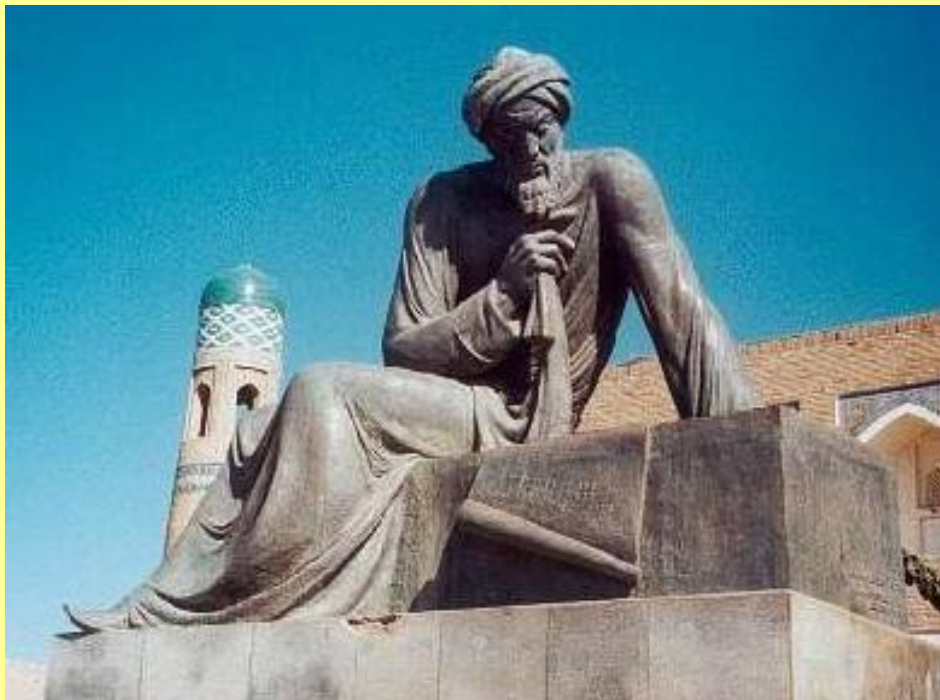


Bibliografía de consulta

9. D.E. Knuth. El Arte de Programar Ordenadores. Vol 1. Reverté, 1985.
10. U. Manber. Introduction to Algorithms. Addison-Wesley, 1989.
11. Ming-Yang Kao (ed.). Encyclopedia of Algorithms. Second edition Springer, 2016
12. J.J. McConnell: Analysis of Algorithms. Jones and Bartlett Pub. (2004)
13. I. Parberry. Problems on Algorithms. Prentice Hall (1995).
14. R. Sedgewick. Algorithms. Second edition. Addison-Wesley, 1988.
15. R. Sedgewick y P. Flajolet. An Introduction to Analysis of Algorithms. Addison-Wesley, 1997.
16. J.D. Smith: Design and Analysis of Algorithms. PWS-KENT Publishing Co. (1989).



Forma de aprobar la asignatura



- Estudiar todos los días un poco
- Contrastar y ampliar lo estudiado con el libro de la asignatura, los apuntes en **PRADO** y los otros que se han recomendado
- Asistir a clase regularmente, participar activamente y **tomar apuntes**.
- Hacer tantos **ejercicios numéricos** como puedan, en solitario y en grupo
- No memoricen.
- Razonen y comprendan lo que se explica

Hay un horario oficial de tutorías, pero me pueden localizar cuando les venga mejor (dentro de un orden)



Material de ayuda

- En la página:
<http://modo.ugr.es/algoritmos>
pueden encontrar:
 - Ejercicios
 - Exámenes de otros años
 - Animaciones algorítmicas...
 - Todos tienen acceso
- Depositaré material adicional en **PRADO**



The screenshot shows the website for 'Teoría de Algoritmos' at the University of Granada. The header includes navigation links: Principal, Teoría, Prácticas, and Enlaces. The page title is 'TEORÍA DE ALGORITMOS' with the subtitle 'MATERIAL DE APOYO DOCENTE PARA LA ASIGNATURA'. A central announcement box states: 'La revisión del examen conlleva la posibilidad de ajuste (subida o bajada) de la nota.' To the right is a portrait of a historical figure and a Google search bar. The left sidebar lists course details: 'Curso: Segundo', 'Titulación: Ingeniería Informática', 'Cuatrimestre: Primero', 'Tipo: Troncal', 'Créditos: -Teoría: 6, -Prácticas: 3', 'Area: Ciencias de la Computación e Inteligencia Artificial', 'Tutorías: Lunes de 17 a 19, Jueves 10 a 12', and 'Profesores: - J.Luis Verdegay, - David Pelta'. The main content area contains a welcome message and information about the website's purpose and resources.

<http://modo.ugr.es/algoritmos/seccion.php?id=teoria>



José Luis Verdegay

Catedrático

Departamento de Ciencias de la Computación e Inteligencia Artificial
Despacho 7

e-mail: verdegay@ugr.es

<https://es.linkedin.com/in/jverdegay>

http://es.wikipedia.org/wiki/Jos%C3%A9_Luis_Verdegay

<http://scholar.google.com/citations?user=7AjYKlcAAAAJ&hl=en>