Universidad de San Carlos de Guatemala
Escuela de Ciencias Físicas y Matemáticas
Programas de Licenciatura

Análisis de Algoritmos

1. Descripción del curso

Nombre: Análisis de algoritmos
Prerrequisitos: M503 - M504
Profesor: Luis Alfredo Alvarado Rodríguez
Código: M603
Créditos: 5
Semestre: Segundo, 2025

El curso se dedica al estudio del diseño y, principalmente, del análisis de algoritmos. Se estudian problemas de matemática pura y aplicada. En particular, se desarrollan técnicas de índole combinatoria para la determinación de cotas del tiempo de ejecución de algoritmos. Se analizan algoritmos famosos como Quicksort y la multiplicación rápida de matrices, problemas relacionados con teoría de números y el concepto de NP-completitud.

2. Competencias

2.1. Competencias generales

2.1.1 Capacidad de abstracción, incluido el desarrollo lógico de teorías matemáticas y las relaciones entre ellas.

2.1.2 Capacidad para formular problemas en lenguaje matemático, de forma tal que se facilite su análisis y solución.

2.1.3 Capacidad para formular problemas, tomar decisiones e interpretar las soluciones en los contextos originales.

2.1.4 Capacidad para contribuir a la construcción de modelos matemáticos a partir de situaciones reales.

2.1.5 Capacidad para utilizar herramientas computacionales para plantear y resolver problemas.

2.1.6 Capacidad para comprender problemas, abstraer lo esencial de ellos y resolverlos.

2.1.7 Capacidad para extraer información cualitativa de datos cuantitativos.

2.1.8 Capacidad para detectar inconsistencias.

2.2. Competencias específicas

a. El estudiante analiza problemas del quehacer científico, encuentra patrones, propone algoritmos y los implementa en algún lenguaje de programación.

b. Capacidad para analizar el tiempo teórico de ejecución de un algoritmo en particular.

c. Comprensión de la teoría general de autómatas.

d. Capacidad para implementar algoritmos eficaces y eficientes.

e. Capacidad para desarrollar software en equipos de dos o más personas, mediante programación estructurada y programación orientada a objetos.

3. Unidades

3.1. Preliminares matemáticos

Descripción: Definición de algoritmo. Necesidad del análisis de algoritmos. Análisis de algoritmos. Caso promedio del análisis de algoritmos. Ejemplo: Quicksort y su análisis (incluido el caso promedio). Aproximaciones asintóticas. Distribuciones. Algoritmos aleatorizados.

Duración: 15 períodos de 50 minutos.

Metodología: Los períodos de clase son magistrales, desarrollando algoritmos en pseudocódigo y diagramas de flujo. Se dan ejemplos de código real implementado en Python.

Evaluación: Se evaluará a través de tareas semanales, ejercicios de programación en Python y un problema en el primer examen parcial.

3.2. Algoritmos recursivos

Descripción: Crecimiento de funciones: notaciones O grande, o pequeña, omega y theta grandes; definiciones, ejemplos y propiedades. Expansión asintótica. Manipulación de expansiones asintóticas. Aproximaciones asintóticas de sumas finitas. Suma de Euler-Maclaurin. Bivariación asintótica. Método de Laplace. Ejemplos del análisis de algoritmos (normal y Poisson).

Duración: 20 períodos de 50 minutos.

Metodología: Los períodos de clase son magistrales, desarrollando algoritmos en pseudocódigo y diagramas de flujo. Se implementa código real en Python.

Evaluación: Se evaluará a través de tareas semanales, ejercicios de programación en Python y dos problemas en el primer examen parcial.

3.3. Algoritmos en teoría de números

Descripción: Árboles binarios. Bosques y árboles. Equivalencia combinatoria de árboles y árboles binarios. Propiedades de árboles. Árboles binarios de búsqueda. Longitud promedio de un camino en árboles de Catalán. Longitud promedio de un camino en árboles binarios de búsqueda. Otros tipos de árboles.

Duración: 20 períodos de 50 minutos.

Metodología: Los períodos de clase son magistrales, desarrollando algoritmos en código Java.

Evaluación: Se evaluará a través de tareas semanales, ejercicios de programación, proyectos de programación en Python/Java y un problema en el segundo examen parcial.

3.4. NP-completitud

Descripción: Inteligencia artificial: introducción. Problemas conocidos: problema del viajante de comercio, satisfacibilidad booleana, programación lineal entera. Búsqueda no informada: búsqueda, búsqueda en amplitud, búsqueda en profundidad y búsqueda de costo uniforme.

Duración: 8 períodos de 50 minutos.

Metodología: Los períodos de clase son magistrales, desarrollando algoritmos en código Python.

Evaluación: Se evaluará a través de tareas semanales, ejercicios de programación, proyectos de programación en Python y un problema en el segundo examen parcial.

4. Evaluación del curso

Los porcentajes asignados a cada uno de los elementos de la evaluación están de acuerdo con el Reglamento General de Evaluación y Promoción del Estudiante de la Universidad de San Carlos de Guatemala.

Proyecto de programación	20 puntos
2 exámenes parciales	30 puntos
Exámenes cortos, tareas y hojas de trabajo	25 puntos
Examen final	25 puntos
Total	**100 puntos**

5. Bibliografía

1. **Wilf, H. S. 'Algorithms and Complexity". A K Peters, Ltd. 2da edición. 2002.**
2. Sedgewick, R. et al. 'An introduction to analysis of algorithms". Addison-Wesley. 2da edición. 2012.
3. Clifford Stein y otros. 'Introduction to algorithms". Cambridge, 2009.
4. Dexter Kozen. 'The design and analysis of algorithms". Springer-Verlag, 1991.
5. Donald Knuth. 'The art of computer programming". Vols 2-3. Addison-Wesley. 1968.
6. Dasgupta. 'Algorithms". 2006.

http://ecfm.usac.edu.gt/programas