

CARRERA: Licenciatura en Ciencias de Datos	CURSO LECTIVO: 2024
CÁTEDRA: Algoritmia y Lógica Computacional	CURSO: 2º año - 1º semestre
DURACIÓN: Semestral	Hs. TOTALES: 80 Hs. Reloj Totales
SEMANAS: 16	Hs. TEÓRICAS: 32 Hs. Reloj Totales Hs. PRÁCTICAS: 48 Hs. Reloj Totales

PROFESOR PROTITULAR: QUINTERO RINCON, ANTONIO

1. OBJETIVOS DE LA ASIGNATURA

Introducir al alumno en el diseño y en el análisis de la complejidad de diferentes tipos de algoritmos con su correspondiente implementación computacional

2. UNIDADES TEMÁTICAS

1. Unidad I:

Calculo de complejidad exacta en algoritmos iterativos. Complejidad asintótica. Notación "O grande". Complejidad por definición. Algebra de "O grande".

2. Unidad II:

2.1. Algoritmos recursivos: Introducción, ejemplos, propiedades y teoremas relativos.

2.2 Teorema maestro.

2.3 Estructuración, análisis e implementación de algoritmos para: cálculo de factoriales, re-ordenamiento de arreglos, multiplicación rápida de matrices y transformada rápida de Fourier.

2.4 Algoritmos Semi-numéricos: Análisis e implementación del algoritmo de la división de enteros y del máximo común divisor. Números primos. Algoritmo de Euclides.

2.5 Algoritmos de búsqueda y de clasificación: Análisis e implementación de algoritmos de búsqueda y clasificación. Teoremas relativos. Análisis de algoritmos recursivos de clasificación. Teoremas. Ejemplos.

2.6 Revisión del concepto de relaciones de recurrencia. Técnicas de cálculo. Relaciones de división y conquista.

3. Unidad III

3.1. Lógica Computacional: lenguaje de la lógica proposicional. Sentencias válidas y tablas de verdad.

Estructuras: estructuras deductivas. Sistemas de Kleene. Deducción. Metodos de deducción y de demostración.

Cálculo de predicados: lógica de predicados: lenguaje de predicados. Reglas semánticas. Teoremas. Modelos y deducción, equivalencia y resolución.

4. Unidad IV (unidad nueva, de Complejidad Computacional Mat416) Computabilidad

4.1 Numeración Godel. Ejemplos. Teoremas.

4.2 Máquinas de Turing y su complejidad

4.3 Clasificación de problemas en tratables y no tratables. Propiedades. Teoremas.

4.4 Problemas NP-completos. Ejemplos. Propiedades. Teorema de Cook

3. BIBLIOGRAFÍA

Algoritmia:

1. DISCRET MATHEMATICS WITH ALGORITHMS, Albertson, M. and Hotchinson, J.P. John Wiley - 1988.
2. CURSO TEORICO-PRÁCTICO DE MATEMATICA DISCRETA, Caputti, T. 2007.
3. CONCRETE MATHEMATICS, Graham, Knuth and Patashnik. Addison Wesley - 1994.
4. MATEMATICAS DISCRETA Y COMBINATORIA, Grimaldi, R.P. Addison Wesley - 1989.
5. THE ART OF PROGRAMMING, Knuth, D.E. Volumen II, Addison Wesley.
6. INTRODUCTION TO LANGUAGES AND THE THEORY OF COMPUTATION, Martin, John C. Mac Grow Hill - 1991.

Lógica Computacional:

7. MATHEMATICAL LOGIC FOR COMPUTER SCIENCE, Ben-Ari, M. Prentice Hall International - 1993.
8. THE DEDUCTIVE FOUNDATIONS OF COMPUTER PROGRAMMING, Zohar, Manna and Waldinger, R. Addison Wesley - 1993.
9. INTRODUCTION TO ALGORITHMS, Cormen. 2001.

4. METODOLOGÍA

Las clases serán teórico-prácticas y fomentarán la participación del alumno mediante el desarrollo de trabajos prácticos individuales y grupales.

5. EVALUACIONES Y CRITERIOS PARA LA APROBACIÓN

La metodología de evaluación para aprobar la cursada de la materia y estar habilitado para rendir el examen final consistirá en cumplir con la asistencia exigida por la Universidad, aprobar el examen parcial en la fecha convenida y cumplimentar las implementaciones computacionales que se exijan.