

PROGRAMA DE MATERIA

DATOS DE IDENTIFICACIÓN

| | | | | | |
|---|--|-----------|----------------------------|----------------------|-------|
| MATERIA: | TEORÍA DE LA COMPLEJIDAD COMPUTACIONAL | | | | |
| CENTRO ACADÉMICO: | CIENCIAS BÁSICAS | | | | |
| DEPARTAMENTO ACADÉMICO: | CIENCIAS DE LA COMPUTACIÓN | | | | |
| PROGRAMA EDUCATIVO: | INGENIERÍA EN COMPUTACIÓN INTELIGENTE | | | | |
| AÑO DEL PLAN DE ESTUDIOS: | 2017 | SEMESTRE: | 6° | CLAVE DE LA MATERIA: | 25785 |
| ÁREA ACADÉMICA: | IAyFC | | PERIODO EN QUE SE IMPARTE: | ENERO-JULIO | |
| HORAS SEMANA T/P: | 2T/3P | | CRÉDITOS: | 7 | |
| MODALIDAD EDUCATIVA EN LA QUE SE IMPARTE: | PRESENCIAL | | NATURALEZA DE LA MATERIA: | TEÓRICO- PRÁCTICO | |
| ELABORADO POR: | EEPL, FSQA, FJOZ, JCPG | | | | |
| REVISADO Y APROBADO POR LA ACADEMIA DE: | IAyFC | | FECHA DE ACTUALIZACIÓN: | Enero 2024 | |

DESCRIPCIÓN GENERAL

En este curso teórico-práctico el estudiante entenderá los elementos fundamentales de la teoría de la Computación y límites del uso de las computadoras en la solución de problemas, usando los conceptos de máquina de Turing y los resultados básicos de este esquema de computación, las técnicas más usadas en el diseño de algoritmos y por último la revisión de algunas clases de complejidad de problemas. Se desarrollarán las habilidades de interpretar y manejar símbolos abstractos, deducir e inducir utilizando lenguaje matemático; identificar, modelar e implementar la solución de problemas. Esta materia es simultánea a Aprendizaje Inteligente y Minería de Datos y es antecedente de Metaheurísticas I y Metaheurísticas II.

OBJETIVO (S) GENERAL (ES)

Al finalizar el curso, el estudiante interpretará las teorías sobre la que se fundamentan las Ciencias de la Computación en cuanto al manejo de la máquina de Turing y los resultados básicos de este esquema de computación; asimismo, aplicará las técnicas más usadas para el diseño de algoritmos, calculará la complejidad de tiempo de un algoritmo e identificará y categorizará los problemas de acuerdo a su complejidad con la finalidad de poseer recursos para la medida de la complejidad de los problemas y sus soluciones computacionales con actitudes y valores como colaboración, crítica, organización, compromiso, disciplina, creatividad y autonomía.

CONTENIDOS DE APRENDIZAJE

| UNIDAD TEMÁTICA I: Funciones de orden y análisis de algoritmos (25 horas aprox.) | | |
|--|---|---------------------|
| OBJETIVOS PARTICULARES | CONTENIDOS | FUENTES DE CONSULTA |
| Que el alumno evalúe la eficiencia de algoritmos mediante el uso de funciones, | 1. Introducción. 1.1. Definición de algoritmo. 1.2. Compatibilidad. | 2,4,7,9. |

PROGRAMA DE MATERIA

| | | |
|---|--|--|
| <p>con la finalidad de determinar su complejidad.</p> <p>Que el alumno compare dos algoritmos en función del espacio (memoria que utiliza) y/o tiempo (lo que tarda en ejecutarse).</p> <p>Determinar el peor caso, mejor caso y el caso promedio de un algoritmo, así como los problemas con los que se comporta de alguna de las tres formas mencionadas.</p> | <p>1.3. Elección de un algoritmo.</p> <p>1.4. Límites de tiempo.</p> <p>1.5. Tamaño de una instancia.</p> <p>2. Métodos para el análisis de la complejidad.</p> <p>2.1. Tiempo de ejecución de un algoritmo.</p> <p>2.2. Mejor caso, caso medio y peor caso.</p> <p>2.3. Notación asintótica.</p> <p>2.4. Funciones de orden. (Órdenes de eficiencia)</p> <p>2.5. Notación O grande, Ω, Θ.</p> <p>3. Ejercicios.</p> | |
|---|--|--|

UNIDAD TEMÁTICA II: Clasificación de los problemas (25 horas aprox.)

| OBJETIVOS PARTICULARES | CONTENIDOS | FUENTES DE CONSULTA |
|---|---|---------------------|
| <p>Que el alumno conozca y entienda los fundamentos del modelo de la máquina de Turing y comprenda su importancia en la formalización del concepto de algoritmo.</p> <p>Que el alumno identifique los problemas no tratables.</p> <p>Que el alumno pueda diseñar e implementar un algoritmo para resolver los problemas no tratables.</p> | <p>1. Introducción.</p> <p>2. Decidibilidad de los problemas.</p> <p>3. Definición de la máquina de Turing (MT).</p> <p>3.1 Notación para la máquina de Turing.</p> <p>3.2 Lenguaje de una máquina de Turing.</p> <p>3.3 Máquinas de Turing y parada.</p> <p>3.4 Máquinas de Turing No Deterministas (MTN).</p> <p>4. Clasificación de los problemas según su complejidad.</p> <p>4.1 Problemas P.</p> <p>4.2 Problemas NP.</p> <p>4.3 Relación entre los problemas P y NP.</p> <p>4.4 Problemas NP-Complejos.</p> <p>5. Clasificación de problemas no tratables.</p> <p>5.1 Definición.</p> <p>5.2 Representación formal.</p> <p>5.3 Enfoques para la solución de los problemas.</p> <p>6. Ejemplos.</p> | 1,2,3 |

UNIDAD TEMÁTICA III: Técnicas de diseño de algoritmos y estudio de su complejidad (30 horas aprox.)

| OBJETIVOS PARTICULARES | CONTENIDOS | FUENTES DE CONSULTA |
|---|--|---------------------|
| <p>Que el alumno conozca y aplique técnicas de diseño de algoritmos.</p> <p>Que el alumno conozca la complejidad de los diferentes algoritmos.</p> <p>Que el alumno identifique el mejor algoritmo/técnica para la solución de un determinado problema.</p> | <p>1. Introducción</p> <p>2. Enfoque Divide y Vencerás (Divide and Conquer algorithms).</p> <p>2.1. Características generales.</p> <p>2.2. Modelo.</p> <p>2.3. Ejemplos.</p> <p>3. Algoritmo Voraz (Greedy algorithms).</p> <p>3.1. Características generales.</p> <p>3.2. Modelo.</p> <p>3.3. Ejemplos.</p> | 2,4,5,6,7. |

PROGRAMA DE MATERIA

| | | |
|--|--|--|
| | 4. Programación Dinámica (Dynamic Programming). 4.1. Características generales. 4.2. Modelo. 4.3. Ejemplos. | |
|--|--|--|

METODOLOGÍA DE ENSEÑANZA - APRENDIZAJE

Dadas las características del curso, el maestro utilizará una metodología analítica y lógica, presentando conceptos y principios que parten de lo simple a lo complejo, motivando al alumno a una constante participación en clase; apoyándose con tareas dirigidas y el uso del laboratorio para practicar los conocimientos teóricos, donde la coordinación y guía del maestro es fundamental.

RECURSOS DIDÁCTICOS

El maestro hará uso del equipo computacional ubicado en el laboratorio asignado, así como del manejo de pantallas, aula virtual, pizarrón, páginas de Internet y software que puedan ser de utilidad para la impartición de la materia.

EVALUACIÓN DE LOS APRENDIZAJES

La evaluación tiene los siguientes lineamientos:

Dos evaluaciones parciales, cada uno con una ponderación del 25% de la calificación final

Una tercera evaluación con ponderación del 30% de la calificación final.

Trabajo final que representa el 20% de la calificación final.

Nota: Cada evaluación parcial está conformada por 80% del examen y 20% de tareas y participación en clase.

FUENTES DE CONSULTA

BÁSICAS:

1. Computers and Intractability: a Guide to the theory of Np-Completeness. Michael R. Garey y David S. Johnson. 1979. Clasificación 005.1 G229c. (AE)
2. Introduction to Algorithms. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein. The MIT Press and McGraw-Hill Higher Education. Tercera Edición. 2009. Clasificación 005.1 C811i, (MLZ)
3. Teoría de Autómatas, lenguajes y computación. John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman. Pearson, Addison Wesley. Tercera edición. 2008. Clasificación. 511.3 H7911i, (AE)
4. The Algorithm Design Manual. Steven S. Skiena. Springer. Segunda Edición. 2008. Colocación 005.1 S6281a 23

COMPLEMENTARIAS:

5. Combinatorial Optimization: Algorithms and Complexity. Christos H. Papadimitriou, Kenneth Steiglitz. Dover publication Inc. 1998.
6. Practical Analysis of Algorithms. Dana Vrajitoru, William Knight. Springer. 2014.
7. Elements of Theory Computation Harry R. Lewis, Christos H. Papadimitriou. Prentice-Hall, Segunda edición. 1998
8. Computational Complexity: A Modern Approach. Sanjeev Arora, Boaz Barak. Cambridge University Press. Online draft. January 2007.
9. Algorithms. Erickson Jeff. Primera Edición. Online free book. Junio 2019.