



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO  
FACULTAD DE ESTUDIOS SUPERIORES ACATLÁN

PLAN DE ESTUDIOS DE LA LICENCIATURA EN  
MATEMÁTICAS APLICADAS Y COMPUTACIÓN

PROGRAMA DE ASIGNATURA



SEMESTRE: 7 (SÉPTIMO)

Análisis de Algoritmos

CLAVE:

MODALIDAD	CARÁCTER	TIPO	HORAS AL SEMESTRE	HORAS SEMANA	HORAS TEÓRICAS	HORAS PRÁCTICAS	CRÉDITOS
Curso	Optativa	Teórica	64	4	4	0	8

ETAPA DE FORMACIÓN	Terminal
CAMPO DE CONOCIMIENTO	Ciencias de la Computación

SERIACIÓN	Indicativa
ASIGNATURA(S) ANTECEDENTE	Teoría de Gráficas, Probabilidad, Optimización II
ASIGNATURA(S) SUBSECUENTE(S)	Ninguna

**Objetivo general:** El alumno reconocerá las herramientas matemáticas necesarias para el diseño y la evaluación de la complejidad de un algoritmo.

Índice Temático		Horas	
Unidad	Tema	Teóricas	Prácticas
1	Introducción a la teoría de algoritmos	8	0
2	Técnicas básicas del análisis de algoritmos	24	0
3	Paradigmas principales en el diseño de algoritmos	12	0
4	Temas selectos de análisis de algoritmos	20	0
Total de horas:		64	0
Suma total de horas:		64	

HORAS		UNIDAD	CONTENIDO
T	P		
8	0	1	<p><b>INTRODUCCIÓN A LA TEORÍA DE ALGORITMOS</b></p> <p><b>Objetivo particular:</b> El alumno explicará los conceptos de algoritmo, problema computacional, modelo de cómputo y complejidad computacional.</p> <p><b>Temas:</b></p> <ul style="list-style-type: none"> <li>1.1 Breve historia de la teoría de algoritmos             <ul style="list-style-type: none"> <li>1.1.1 El décimo problema de Hilbert</li> <li>1.1.2 La tesis de Church-Turing</li> </ul> </li> <li>1.2 Modelos de cómputo             <ul style="list-style-type: none"> <li>1.2.1 Máquinas de Turing</li> <li>1.2.2 El modelo RAM</li> </ul> </li> <li>1.3 Problemas computacionales y algoritmos</li> <li>1.4 Complejidad Computacional             <ul style="list-style-type: none"> <li>1.4.1 Complejidad en tiempo</li> <li>1.4.2 Complejidad en espacio</li> </ul> </li> <li>1.5 Análisis en el peor caso, mejor caso y caso promedio</li> <li>1.6 Verificación formal de un algoritmo             <ul style="list-style-type: none"> <li>1.6.1 Verificación por invariantes</li> </ul> </li> </ul>
24	0	2	<p><b>TÉCNICAS BÁSICAS DEL ANÁLISIS DE ALGORITMOS</b></p> <p><b>Objetivo particular:</b> El alumno ejercitara el uso de la notación asintótica y la solución de relaciones de recurrencia para la descripción de la complejidad de un algoritmo.</p> <p><b>Temas:</b></p> <ul style="list-style-type: none"> <li>2.1 Análisis Asintótico             <ul style="list-style-type: none"> <li>2.1.1 Notación O-grande</li> <li>2.1.2 Notación <math>\Omega</math></li> <li>2.1.3 Notación <math>\Theta</math></li> </ul> </li> <li>2.2 Métodos de solución de relaciones de recurrencia</li> <li>2.3 Recurrencias de algoritmos Divide y vencerás</li> <li>2.4 Cota inferior de ordenamiento</li> <li>2.5 Algoritmos óptimos de ordenación: Heap sort y Quik sort</li> <li>2.6 Algoritmos lineales de ordenación con información extra: Counting sort</li> <li>2.7 Búsquedas binarias y árboles de búsqueda balanceados</li> </ul>
12	0	3	<p><b>PARADIGMAS PRINCIPALES EN EL DISEÑO DE ALGORITMOS</b></p> <p><b>Objetivo particular:</b> El alumno identificará los principales paradigmas para diseñar un algoritmo.</p> <p><b>Temas:</b></p> <ul style="list-style-type: none"> <li>3.1 Algoritmos Codiciosos (Greedy)</li> <li>3.2 Algoritmos de Programación Dinámica</li> <li>3.3 Algoritmos Backtracking y Branch and Bound</li> </ul>

20	0	4	<b>TEMAS SELECTOS DE ANÁLISIS DE ALGORITMOS</b>
			<p><b>Objetivo Particular:</b> El alumno describirá temas avanzados de las diferentes ramas de la teoría de algoritmos.</p> <p><b>Temas:</b></p> <ul style="list-style-type: none"> <li>4.1 Algoritmos en Teoría de Gráficas</li> <li>4.2 Introducción a la clase de problemas NP-Completos</li> <li>4.3 Algoritmos de Aproximación</li> <li>4.4 Introducción a la Geometría Computacional</li> <li>4.5 Introducción al análisis probabilístico y algoritmos aleatorios</li> </ul>

#### Referencias básicas:

- Cormen, T. H., Leiserson, C. E., Rivest, R. L., Stein, C. Cambridge. (2009). *Introduction to Algorithms* (3 ed). Massachusetts London, England: The MIT Press.
- Dasgupta, S., Papadimitriou, C. y Vazirani, U. (2001). *Algorithms*. U. S. A.: McGraw-Hill.
- Dave, P. H., Dave, H. B. (2008). *Design and Analysis of Algorithms*. India: Pearson Education.
- Goodrich, M. T., Tamassia, R. (2002). *Algorithm Design. Foundations, Analysis and Internet Example*. USA: John Wiley & Sons.
- Kleinberg, J., Tardos, E. (2005). *Algorithms Design*. U. S. A.: Addison Wesley.
- Sedgewick, R., Flajolet, Ph. (1995). *An Introduction to the Analysis of Algorithms*.
- Skiena, S. (1998). *The Algorithms Design Manual*. U.S.A.: Springer.

#### Referencias complementarias:

- Camacho, S. (1996). *Análisis de algoritmos*. México: U.N.A.M., E.N.E.P. Acatlán.
- D. L. Kreher y Stinson, D. R. (1999). *Stinson Combinatorial Algorithms: generation, enumeration and search*. USA: CRC Press.
- De Berg, M., Van Krefeld, M., Overmars, M, Schwarzkopf, O. (2008). *Computational Geometry: Algorithms and Applications* (3th). Alemania: Springer.
- Greene, D. H., Knuth D. E. (1990). *Mathematics for the Analysis of Algorithms Vol1* (3 ed).USA: Progress in Computer Science and Applied Logic.
- Knuth, D. et al. (1994). *Concrete Mathematics* (2 ed). USA: Addison-Wesley Professional.
- Knuth, D. (1998). *The art of computer programming*, vol. I, II, III. E.U.A: Addison-Wesley.
- M. Garey y D. Johnson, *Computers and Intractability: a Guide to the Theory of Np-Completeness*, W. H. Freeman.
- Mitzenmacher, M., Upfal, E. (2005). *Probability and Computing: Randomized: Algorithms and Probabilistic Analysis*. USA: Cambridge University Press.
- Motwani, R. y Raghavan, P. (1995). *Randomized Algorithms*. USA: Cambridge University Press.
- Vazirani, V. V. (2010). *Approximation Algorithms*. USA: Springer.

<b>Sugerencias didácticas:</b>	<b>Sugerencias de evaluación del aprendizaje:</b>
Analizar y producir textos	Examen final oral o escrito
Utilizar tecnologías multimedia	Exámenes parciales
Resolver ejercicios dentro y fuera de clase	Informes de prácticas
Estudiar casos	Informes de investigación
Instrumentar técnicas didácticas como exposición audiovisual, exposición oral, interrogatorio y técnicas grupales de trabajo colaborativo, entre otros	Participación en clase
Realizar visitas de observación	Rúbricas
Usar recursos didácticos en línea	Solución de ejercicios
Desarrollo de programas mediante el uso de paquetes computacionales aplicando los métodos estudiados en el curso.	Trabajos y tareas
En la unidad 4 demostrar la intratabilidad computacional de algunos problemas y diseñar algoritmos que den solución aproximada a estos.	Proyecto práctico

**Perfil Profesiográfico:** El profesor que imparta la asignatura deberá tener el título de licenciado en Matemáticas Aplicadas y Computación o carrera afín, con experiencia profesional y docente en la materia, contar con actualización en el área y preferentemente tener estudios de posgrado.