

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
FACULTAD DE CIENCIAS
CARRERA DE MATEMÁTICO

COMPLEJIDAD COMPUTACIONAL

SEMESTRE: **Séptimo u octavo**

CLAVE:

HORAS A LA SEMANA/SEMESTRE

TEÓRICAS	PRÁCTICAS	CRÉDITOS
5/80	0	10

CARÁCTER: **OPTATIVO.**

MODALIDAD: **CURSO.**

SERIACIÓN INDICATIVA ANTECEDENTE: **Teoría de la Computación.**

SERIACIÓN INDICATIVA SUBSECUENTE: **Ninguna.**

OBJETIVO(S): Introducir al alumno la Complejidad Computacional, a través de la teoría de NP-completez. Dar herramientas para que el alumno pueda continuar a cursos más avanzados de Complejidad y Teoría de la Computación. Contribuir a la formación del alumno en Teoría de la Computación y a aumentar su comprensión de los fundamentos de la Computación.

NUM. HORAS	UNIDADES TEMÁTICAS
10	1. Introducción y conceptos básicos
	1.1 Problemas, algoritmos, complejidad. Motivación.
	1.2 Notación asintótica. codificación, modelos de cómputo.
25	2. La Teoría de NP-completez
	2.1 Máquinas de Turing y la clase P.
	2.2 La clase NP.
	2.3 Relación entre P y NP, transformaciones polinomiales.
	2.4 Definición de NP-completez.
	2.5 Teorema de Cook.
25	3. Demostraciones de problemas NP-completos
	3.1 Problemas básicos: 3SAT, apareamientos, cubierta de vértices, circuito hamiltoniano, clan, partición.
	3.2 Técnicas: restricción, reemplazo local, diseño de componentes.
20	4. Temas selectos
	4.1 Problemas no computables y máquinas de Turing universales.
	4.2 Usando NP-completez para analizar problemas.
	4.3 Enfrentado problemas NP-completos.
	4.4 Otras clases de completud.

BIBLIOGRAFÍA BÁSICA:

1. Garey, M., Johnson, D., *Computers and Intractability - a guide to the theory of NP-completeness*, San Francisco: Freeman and Company, 1979.
2. Gruska, J., *Foundations of Computing*, New York: International Thompson Computer Press, 1997.
3. Hemaspaandra, L., Selman, A. editores, *Complexity Theory Retrospective II*, New York: Springer Verlag, 1997.
4. Hopcroft, J.E., Ullman, J.D., *Introduction to Automata Theory, Languages and Computation*, Reading, Mass.: Addison Wesley, 1979.
5. Lewis, H.R., Papadimitriou, C.H., *Elements of the Theory of Computing*, New Jersey: Prentice-Hall, 1981.
6. Martin, J.C., *Introduction to Languages and the Theory of Computation*, New York: McGraw-Hill, 1991.
7. Papadimitriou, C.H., *Computational Complexity*, Reading, Mass.: Addison Wesley, 1994.
8. Pippinger, N., *Theories of Computability*, Cambridge: Cambridge University Press, 1997.
9. Savage, J., *Models of Computation: Exploring the Power and Limits of Computing*, Reading, Mass.: Addison Wesley, 1997
10. Sipser, M., *Introduction to the Theory of Computation*, Boston: PWS Publishing Company, 1997.

BIBLIOGRAFÍA COMPLEMENTARIA:

1. Gries, David, editor., *Programming in the 1990s, An Introduction to the Calculation of Programs*, New York: Springer Verlag, 1990.
2. Kfoury, A.J., Moll, R. N., Arbib, M. A., *A Programming Approach to Computability*, New York: Springer-Verlag, 1982.

SUGERENCIAS DIDÁCTICAS: Lograr la participación activa de los alumnos mediante exposiciones.

SUGERENCIA PARA LA EVALUACIÓN DE LA ASIGNATURA: Además de las calificaciones en exámenes y tareas se tomará en cuenta la participación del alumno.

PERFIL PROFESIOGRÁFICO: Matemático, físico, actuario o licenciado en ciencias de la computación, especialista en el área de la asignatura a juicio del comité de asignación de cursos.