



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO  
FACULTAD DE INGENIERÍA



PROGRAMA DE ESTUDIO

**ESTRUCTURA DE DATOS Y ALGORITMOS II**

**3**

**10**

Asignatura

Clave

Semestre

Créditos

**INGENIERÍA ELÉCTRICA**

**INGENIERÍA  
EN COMPUTACIÓN**

**INGENIERÍA  
EN COMPUTACIÓN**

División

Departamento

Licenciatura

**Asignatura:**

Obligatoria

Optativa

**Horas/semana:**

Teóricas

Prácticas

Total

**Horas/semestre:**

Teóricas

Prácticas

Total

**Modalidad:** Curso teórico-práctico

**Seriación obligatoria antecedente:** Estructura de Datos y Algoritmos I

**Seriación obligatoria consecuente:** Estructuras Discretas

**Objetivo(s) del curso:**

El alumno diseñará algoritmos para la resolución de problemas de la ciencia e ingeniería en computación.

**Temario**

NÚM.	NOMBRE	HORAS
1.	Algoritmos de ordenamiento	12.0
2.	Algoritmos de búsqueda	8.0
3.	Algoritmos de grafos	6.0
4.	Árboles	6.0
5.	Archivos	8.0
6.	Introducción a los algoritmos paralelos	24.0
		64.0
	Actividades prácticas	32.0
	Total	96.0

## 1 Algoritmos de ordenamiento

**Objetivo:** El alumno diseñará los métodos más importantes de algoritmos para efectuar ordenamientos en la computadora.

**Contenido:**

- 1.1 Ordenamiento.
  - 1.1.1 Bubble Sort.
  - 1.1.2 Heapsort.
  - 1.1.3 QuickSort.
  - 1.1.4 Countig Sort.
  - 1.1.5 Radix Sort.
  - 1.1.6 Merge Sort.

## 2 Algoritmos de búsqueda

**Objetivo:** El alumno aplicará el método de búsqueda apropiado a conjuntos de datos residentes, tanto en la memoria principal, como en la memoria secundaria para generar algoritmos que resuelvan búsquedas.

**Contenido:**

- 2.1 Generalidades.
- 2.2 Definición de la operación de búsqueda.
- 2.3 Búsqueda por comparación de llaves.
  - 2.3.1 Lineal.
  - 2.3.2 Binaria.
  
- 2.4 Búsqueda por transformación de llaves.
  - 2.4.1 Funciones de hash.
  - 2.4.2 Colisiones.

## 3 Algoritmos de grafos

**Objetivo:** El alumno aplicará las formas de representar y operar los grafos y listas lineales para representarlos en la computadora.

**Contenido:**

- 3.1 Representación de grafos.
- 3.2 Búsqueda por expansión.
- 3.3 Búsqueda por profundidad.

## 4 Árboles

**Objetivo:** El alumno aplicará las formas de representar y operar las listas lineales para representarlos en la computadora.

**Contenido:**

- 4.1 Notaciones: infija, prefija, sufija.
- 4.2 Árboles binarios.
  - 4.2.1 Definiciones y operaciones.
  - 4.2.2 Transformación de árboles a árboles binarios.
  - 4.2.3 Recorrido de árboles.
  - 4.2.4 Representación en la computadora.

- 4.3 Árboles B.

- 4.3.1 Árboles B.
- 4.3.2 Árboles B+, algoritmos.
- 4.3.3 Árboles B+ prefijos simples, algoritmos.

## 5 Archivos

**Objetivo:** El alumno interpretará las organizaciones básicas de los archivos, las operaciones que se pueden realizar sobre ellos y su representación mediante diferentes medios de almacenamiento secundario.

**Contenido:**

- 5.1 Generalidades.
- 5.2 Definición y operaciones.
- 5.3 Organización de archivos.
  - 5.3.1 Organización lógica.
  - 5.3.2 Organización física.
  
- 5.4 Acceso a archivos.
  - 5.4.1 Acceso lógico.
  - 5.4.2 Acceso físico.
  
- 5.5 Sistema de archivos.

## 6 Introducción a los algoritmos paralelos

**Objetivo:** El alumno clasificará los elementos a considerar en el diseño y análisis de algoritmos paralelos versus algoritmos seriales para su programación.

**Contenido:**

- 6.1 Niveles de paralelismo. Granularidad.
- 6.2 Algoritmos con memoria compartida.
  - 6.2.1 Carrera de datos.
  - 6.2.2 Inconsistencia de datos.
  - 6.2.3 Modelo PRAM.
  
- 6.3 Técnicas de desarrollo de algoritmos.
  - 6.3.1 Rediseño de estructuras de datos.
  - 6.3.2 Rediseño de algoritmos.
  
- 6.4 Análisis de desempeño de algoritmos paralelos.
  - 6.4.1 Trabajo y profundidad.
  - 6.4.2 Ejemplos clásicos.

---

---

### Bibliografía básica

AHO, Alfred, ULLMAN, Jeffrey, et al.  
*Data Structures and Algorithms*  
New Jersey  
Addison-Wesley, 1983

### Temas para los que se recomienda:

1-5

BAASE, Sara, VAN GELDER, Allen <i>Computer Algorithms: Introduction to Design and Analysis</i> 3rd edition San Diego Addison-Wesley, 1999	1-5
BLAIR-CHAPPELL, Stephen, STOKES, Andrew <i>Parallel Programming with Intel Parallel Studio XE</i> Indiana Wrox, 2012	6
CORMEN, Thomas <i>Algorithms Unlocked</i> Cambridge MA, USA The MIT Press, 2013	1-5
CORMEN, Thomas, LEISERSON, Charles, et al. <i>Introduction to Algorithms</i> 3rd edition MA, USA The MIT Press, 2009	1-5
JEFFERS, James, REINDERS, James <i>Intel Xeon Phi Coprocessor High Performance Programming</i> Waltham MA, USA Morgan Kaufmann, 2013	6
KIRK, David, HWU, Wen-mei <i>Programming Massively Parallel Processors: A Hands-on Approach</i> 2nd edition Waltham MA, USA Morgan Kaufmann, 2012	6
KNUTH, Donald <i>The Art of Computer Programming</i> New Jersey Addison-Wesley Professional, 2011 Volumes 1-4A	1-5
PACHECO, Peter <i>An Introduction to Parallel Programming</i> MA, USA Morgan Kaufmann, 2011	6
RAUBER, Thomas, RÜNGER, Gudula <i>Parallel Programming: for Multicore and Cluster Systems</i> Heidelberg Springer, 2010	6

**Bibliografía complementaria****Temas para los que se recomienda:**

BRASSARD, Gilles, BRATLEY, Paul

*Fundamentals of Algorithmics*

1-5

New Jersey

Prentice Hall, 1995

CAMPBELL, Colin, JOHNSON, Ralph, et al.

*Parallel Programming with Microsoft® .NET: Design Patterns*

6

*for Decomposition and Coordination on Multicore Architectures (Patterns & Practices)* Washington

Microsoft Press, 2010

HERLIHY, Maurice, SHAVIT, Nir

*The Art of Multiprocessor Programming*

6

Waltham MA

Morgan Kaufmann, 2012

KINGSTON, Jeffrey

*Algorithms and Data Structures: Design, Correctness,*

1-5

*Analysis* 2nd edition

New Jersey

Addison-Wesley, 1997

KOZEN, Dexter C.

*The Design and Analysis of Algorithms*

1-5

Ithaca NY

Springer, 1992

PACHECO, Peter

*Parallel Programming with MPI*

6

San Francisco CA, USA

Morgan Kaufmann, 1996

SZNAJDLEDER, Pablo

*Algoritmos a fondo: con implementación en C y JAVA*

Todos

Buenos Aires

Alfaomega, 2012

**Sugerencias didácticas**

Exposición oral	<input checked="" type="checkbox"/>
Exposición audiovisual	<input checked="" type="checkbox"/>
Ejercicios dentro de clase	<input checked="" type="checkbox"/>
Ejercicios fuera del aula	<input checked="" type="checkbox"/>
Seminarios	<input checked="" type="checkbox"/>
Uso de software especializado	<input type="checkbox"/>
Uso de plataformas educativas	<input type="checkbox"/>

Lecturas obligatorias	<input checked="" type="checkbox"/>
Trabajos de investigación	<input checked="" type="checkbox"/>
Prácticas de taller o laboratorio	<input checked="" type="checkbox"/>
Prácticas de campo	<input type="checkbox"/>
Búsqueda especializada en internet	<input type="checkbox"/>
Uso de redes sociales con fines académicos	<input type="checkbox"/>

**Forma de evaluar**

Exámenes parciales	<input checked="" type="checkbox"/>
Exámenes finales	<input checked="" type="checkbox"/>
Trabajos y tareas fuera del aula	<input checked="" type="checkbox"/>

Participación en clase	<input checked="" type="checkbox"/>
Asistencia a prácticas	<input checked="" type="checkbox"/>

**Perfil profesiográfico de quienes pueden impartir la asignatura**

Licenciatura en Ingeniería en Computación, Ciencias de Computación, Matemáticas Aplicadas o una carrera similar. Deseable haber realizado estudios de posgrado, contar con conocimientos y experiencia en el área de ciencias de la computación, contar con experiencia docente o haber participado en cursos o seminario de iniciación en la práctica docente