



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

Licenciatura en Ciencias de la Computación

Facultad de Ciencias

Programa de la asignatura



Denominación de la asignatura:

Estructuras de Datos

| | | | |
|-------------------------------|---|--------------------------------------|----------------------------|
| Clave: 1222 | Semestre: 2 | Eje temático: Programación | No. Créditos: 12 |
| Carácter: Obligatoria | Horas | | Horas por semana |
| Tipo: Teórico-Práctica | Teoría: | Práctica: | Total de Horas |
| | 3 | 6 | |
| Modalidad: Curso | Duración del programa: Semestral | | |

Asignatura con seriación obligatoria antecedente: Ninguna

Asignatura con seriación obligatoria subsecuente: Análisis de Algoritmos, Fundamentos de Bases de Datos; Computación Distribuida; Lenguajes de Programación

Asignatura con seriación indicativa antecedente: Estructuras Discretas; Introducción a Ciencias de la Computación

Asignatura con seriación indicativa subsecuente: Modelado y programación; Autómatas y Lenguajes Formales; Análisis de Algoritmos

Objetivos generales:

Comprender el papel fundamental que cumple la abstracción de datos en la elaboración de modelos correctos y completos para resolver problemas por medio de la computadora.

Conocer el panorama de las estructuras de datos más usuales, sus características y las diferentes maneras de instrumentarlas en un lenguaje de programación orientado a objetos.

Analizar la complejidad en tiempo y espacio de algoritmos elementales.

Conocer y aplicar los algoritmos más comunes de búsqueda y ordenamiento.

Conocer los elementos de criterio elementales para elaborar programas correctos y eficientes en función del contexto.

| Índice temático | | | |
|-----------------------------|--|------------|-----------|
| Unidad | Temas | Horas | |
| | | Teóricas | Prácticas |
| I | Tipos de datos abstractos | 4 | 9 |
| II | Elementos de complejidad algorítmica | 3 | 6 |
| III | Elementos de corrección de algoritmos | 2 | 3 |
| IV | Arreglos | 1 | 3 |
| V | Recursión | 4 | 8 |
| VI | Listas | 3 | 6 |
| VII | Pilas | 3 | 6 |
| VIII | Colas | 3 | 6 |
| IX | Árboles | 2 | 4 |
| X | Árboles binarios | 3 | 6 |
| XI | Árboles binarios de búsqueda | 3 | 6 |
| XII | Árboles de búsqueda balanceados | 3 | 6 |
| XIII | Funciones y tablas de dispersión (<i>hash</i>) | 3 | 6 |
| XIV | Heaps | 2 | 3 |
| XV | Algoritmos de ordenamiento | 3 | 6 |
| XVI | Algoritmos en gráficas | 6 | 12 |
| Total de horas: | | 48 | 96 |
| Suma total de horas: | | 144 | |

| Contenido temático | |
|---|--|
| Unidad | Tema |
| I Tipos de datos abstractos | |
| I.1 | Definición, especificación e instrumentación de tipos de dato abstractos (TDAs). |
| I.2 | Conceptos de clase, objeto y estructura de datos y su relación con TDA. |
| I.3 | Encapsulación, acoplamiento y cohesión. |
| II Elementos de complejidad algorítmica | |
| II.1 | Análisis asintótico, notación $O()$. |
| II.2 | Recurrencias. |
| II.3 | Ejemplos de cálculo de complejidad. |
| III Elementos de corrección de algoritmos | |
| III.1 | Pruebas basadas en invariantes. |
| III.2 | Pruebas basadas en las propiedades de un TDA. |
| IV Arreglos | |
| IV.1 | Instrumentación de arreglos: polinomios de direccionamiento vectores de Iliffe. |
| IV.2 | Arreglos empacados. |
| V Recursión | |
| V.1 | Estrategia <i>divide y vencerás</i> , solución de problemas mediante recursión. |
| V.2 | Búsqueda con retroceso mínimo (<i>backtrack</i>). |
| VI Listas | |
| VI.1 | TDA Lista, definición, alternativas de instrumentación. |
| VI.2 | Inserción, remoción y recuperación de elementos en una Lista. |

| | |
|---|--|
| VI.3 | Variantes de lista: ligadura doble, circular, etc. |
| VI.4 | Complejidad de operaciones, iterador. |
| VI.5 | Aplicaciones. |
| VII Pilas | |
| VII.1 | TDA Pila, definición, alternativas de instrumentación. |
| VII.2 | Inserción, remoción y recuperación de elementos en una Pila. |
| VII.3 | Aplicaciones. |
| VIII Colas | |
| VIII.1 | TDA Cola, definición, alternativas de instrumentación. |
| VIII.2 | Inserción, remoción y recuperación de elementos en una Cola. |
| VIII.3 | Aplicaciones. |
| IX Árboles | |
| IX.1 | Concepto de Árbol, conceptos relacionados, TDA, instrumentación y aplicaciones. |
| X Árboles binarios | |
| X.1 | Conceptos y cotas relacionadas con árboles binarios, TDA. |
| X.2 | Instrumentaciones de árboles binarios. |
| X.3 | Recorridos. |
| X.4 | Aplicaciones. |
| XI Árboles binarios de búsqueda | |
| XI.1 | Definición, propiedades y TDA. |
| XI.2 | Algoritmos de inserción, remoción y recuperación de elementos, complejidad. |
| XI.3 | Aplicaciones. |
| XII Árboles de búsqueda balanceados | |
| XII.1 | Motivación, definición, propiedades y TDA. |
| XII.2 | Árboles AVL, árboles rojinegros, definición, propiedades y TDA. |
| XII.3 | Algoritmos de inserción, remoción y recuperación de elementos conservando el balanceo. |
| XII.4 | Análisis de complejidad. |
| XII.5 | Aplicaciones. |
| XIII Funciones y tablas de dispersión (<i>hash</i>) | |
| XIII.1 | Motivación, diseño de funciones de dispersión. |
| XIII.2 | Colisiones y su manejo. |
| XIV Heaps | |
| XIV.1 | Definición, propiedades y TDA. |
| XIV.2 | Algoritmos de inserción, remoción y recuperación de elementos. |
| XIV.3 | Análisis de complejidad. |
| XIV.4 | Aplicaciones. |
| XV Algoritmos de ordenamiento | |
| XV.1 | Métodos elementales de complejidad cuadrática: inserción, selección, burbuja. |
| XV.2 | Métodos eficientes: <i>Shellsort</i> , <i>Heapsort</i> , <i>Quicksort</i> y <i>Mergesort</i> . |
| XV.3 | Análisis de complejidad. |
| XVI Algoritmos en gráficas | |
| XVI.1 | Representación de gráficas, complejidad de espacio. |

| | |
|-------|--|
| XVI.2 | Recorridos elementales: en amplitud (BFS) y a profundidad (DFS). |
| XVI.3 | Rutas más cortas: algoritmo de Dijkstra y Floyd. |
| XVI.4 | Árbol generador de peso mínimo, algoritmos de Prim y Kruskal. |

Bibliografía básica:

1. Carrano, Frank M. y Janet J. Prichard, *Data Abstraction and Problem Solving with Java*, 2a ed., Addison Wesley, 2005.
2. Goldman Sally y Kenneth J. Goldman, *A Practical Guide to Data Structures and Algorithms Using Java*, Chapman & Hall - CRC Press, 2007.
3. Skiena, Steven S., *The Algorithm Design Manual*, 2a ed. Springer, 2008.

Bibliografía complementaria:

1. Brass, Peter, *Advanced Data Structures*, Cambridge University Press, 2008.
2. Cormen, Thomas H., Charles E. Leiserson, Ronald R. Rivest y Clifford Stein, *Introduction to Algorithms*, 3a ed. MIT Press, 2009.
3. Gamma, Erich, Richard Helm, Ralph Johnson y John M. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley Professional, 1994.
4. Preiss, Bruno R., *Data Structures and Algorithms with Object-Oriented Design Patterns in Java*, Wiley, 1999.

| | | | |
|-----------------------------------|-----|---|-----|
| Sugerencias didácticas: | | Métodos de evaluación: | |
| Exposición oral | (X) | Exámenes parciales | (X) |
| Exposición audiovisual | (X) | Examen final escrito | (X) |
| Ejercicios dentro de clase | (X) | Trabajos y tareas fuera del aula | (X) |
| Ejercicios fuera del aula | (X) | Exposición de seminarios por los alumnos | () |
| Seminarios | () | Participación en clase | () |
| Lecturas obligatorias | () | Asistencia | () |
| Trabajo de investigación | () | Seminario | () |
| Prácticas de taller o laboratorio | (X) | | |
| Prácticas de campo | () | Otras: Prácticas de laboratorio. Proyecto final de programación. | |
| Otras: _____ | | | |

Perfil profesiográfico:

Egresado preferentemente de la Licenciatura en Ciencias de la Computación o Matemático con especialidad en Computación con amplia experiencia de programación. Es conveniente que posea estudios de posgrado en la disciplina. Con experiencia docente.