

PROGRAMA DE ASIGNATURA

ASIGNATURA: Algoritmos y Estructuras de Datos II	AÑO: 2012
CARÁCTER: Obligatoria	
CARRERA: Licenciatura en Ciencias de la Computación	
RÉGIMEN: cuatrimestral	CARGA HORARIA: 120 hs.
UBICACIÓN en la CARRERA: Segundo año – Primer cuatrimestre	

FUNDAMENTACIÓN Y OBJETIVOS

Se pretende que el alumno adquiera:

- capacidad para analizar algoritmos
- capacidad y hábito de identificar abstracciones al abordar un problema
- familiaridad con técnicas frecuentes de diseños de algoritmos

CONTENIDO

Unidad I: Análisis de algoritmos

Motivación

Problema de Ordenación. Ordenación por selección. El ciclo for. Conteo de operaciones de un programa. Definición en símbolos (ops). Conteo de comparaciones de la ordenación por selección. Incidencia del crecimiento del tamaño de los datos en la performance del algoritmo. Introducción del término “del orden de”. Ordenación por inserción. Conteo. Peor caso, mejor caso y caso medio.

La notación O

Significado de peor caso y caso medio. Operaciones elementales. Análisis aproximado. La notación O . Definición formal. Ejemplos. Insignificancia de las constantes aditivas y multiplicativas. Reflexividad y transitividad. Igualdad entre los O 's de funciones. Equivalencia entre logaritmos de diferente base. Regla del límite. Jerarquía: logaritmos, polinomios, exponenciales, factoriales. El O de la suma y el producto. El O de un polinomio. Terminología: funciones y algoritmos logarítmicos, cuadráticos, cúbicos, polinomiales, exponenciales. Balance entre tiempo y espacio de los algoritmos.

Notación complementaria

Las notaciones Ω y Θ . Definición formal. Equivalencia entre expresiones con Ω , Θ y O . Distinción entre Ω , Θ y O por un lado y el análisis del mejor caso, caso medio y

peor caso por el otro. Notación $f(n) + O(g(n))$, $f(n) * O(g(n))$, $f(n)^{O(g(n))}$, etc.

Ejemplos

Búsqueda lineal. Análisis de mejor caso, peor caso y caso medio. Búsqueda lineal en un arreglo ordenado. Análisis de mejor caso, peor caso y caso medio. Búsqueda binaria. Análisis de mejor caso, peor caso y caso medio. Contraste entre el algoritmo lineal y el logarítmico cuando el tamaño de la entrada crece.

Motivación de la recurrencias

Transformación gradual de la ordenación por selección en la ordenación por intercalación. Versión funcional de la ordenación por intercalación. Versión imperativa. Análisis de la ordenación por intercalación. Resolución de la recurrencia.

Recurrencias

Recurrencias homogéneas. Formulación, resolución y demostración. Ejemplos. Recurrencias no homogéneas. Formulación y resolución. Ejemplos. Recurrencias divide y vencerás. Formulación y resolución. Ejemplos. Funciones eventualmente no decrecientes. Funciones i -uniformes y uniformes. Regla de la uniformidad. Ejemplos. Demostración de la resolución de recurrencias divide y vencerás.

Unidad II: Estructura de datos

Introducción

Importancia de la elección de estructuras de datos adecuadas. Los tipos concretos como concepto relativo a un lenguaje de programación. Los tipos abstractos como concepto asociado a un problema que se quiere resolver. Tipos abstractos y sus diferentes representaciones.

Estructuras concretas

Estructuras concretas más comunes en los lenguajes de programación. Arreglos. Operaciones para manipularlos. Almacenamiento en memoria. Representación gráfica. Eficiencia de las operaciones. Diferentes tipos de índices. Tipos enumerados. Ciclo for generalizado. Listas como tipos concretos. Operaciones para manipularlos. Almacenamiento en memoria. Representación gráfica. Eficiencia de las operaciones. Registros. Operaciones para manipularlos. Almacenamiento en memoria. Representación gráfica. Problema de aliasing.

Tipos abstractos de datos (TAD's)

Tipos abstractos más usuales. TAD Lista. Operaciones. Ecuaciones. Representaciones usando arreglos. TAD Pila. Operaciones. Ecuaciones. Ejemplo: versión iterativa de la ordenación por intercalación usando una pila. Ejemplo: evaluación de expresiones en notación polaca inversa usando una pila. Representaciones usando arreglos y listas. TAD cola. Operaciones. Ecuaciones. Representaciones usando arreglos y listas. Listas enlazadas. Representación gráfica. Representaciones de listas, pilas y colas usando listas enlazadas, listas enlazadas con puntero al último y listas circulares. Aliasing y errores usuales al programar con punteros. Manejo de memoria en ejecución. TAD árbol binario. Representación gráfica. Operaciones. Ecuaciones. Terminología botánica y genealógica. Posiciones. Subárbol correspondiente a una posición. Posiciones de un

árbol. Elemento alojado en una posición de un árbol. Representación usando punteros. Árboles binario de búsqueda (ABB). Operaciones: versiones recursiva e iterativa. Eficiencia. TAD cola de prioridades. Operaciones. Ecuaciones. Heap. Implementación de cola de prioridades usando un heap. Eficiencia de las operaciones. Heap usando arreglos. Eficiencia. Ordenación con heap. Eficiencia. Ordenación con heap sin arreglo auxiliar.

Otras estructuras

Problema unión-find. Inicialización virtual.

Unidad III: Estrategias conocidas de resolución de problemas

Uso de heurísticas en algoritmos. Estrategias de diseño de algoritmos.

Algoritmos voraces

Propiedades generales de los algoritmos voraces (o greedy o glotones o golosos). Esquema general. Problema de la moneda simplificado. Problema de la mochila simplificado. Problema del camino de costo mínimo. Algoritmo de Dijkstra. Problema del árbol generador de costo mínimo. Algoritmos de Prim y de Kruskal.

Divide y vencerás

Propiedades generales de la técnica divide y vencerás. Esquema general. Búsqueda binaria. Ordenación por intercalación. Ordenación rápida (quicksort). Cálculo eficiente de la potencia n -ésima de un número. Multiplicación de grandes números.

Programación dinámica

Funciones recursivas potencialmente exponenciales. Confección de tablas. Fibonacci. Problema de la moneda. Problema de la mochila. Funciones con memoria. Revisión de los problemas de la moneda y de la mochila. Problema de los caminos de costo mínimo. Algoritmo de Floyd. Cómputo de números combinatorios. Reducción del espacio necesario para las tablas.

Backtracking

Motivación: algoritmo para salir de un laberinto. Recorrida de árboles binarios. Pre-orden, in-orden y pos-orden de izquierda a derecha y de derecha a izquierda. In-orden para listar ordenadamente un ABB. Recorrida de árboles finitarios. Precondicionamiento. Pre-orden y pos-orden para resolver el problema del ancestro. Recorrida de árboles dirigidos o no. DFS recursivo e iterativo con pila. BFS con cola. Grafos implícitos. Problema de la mochila con pesos reales. Problema de las ocho reinas. Podas graduales al grafo de búsqueda.

Trabajos prácticos especiales

Desarrollo de cuatro trabajos de laboratorio

BIBLIOGRAFÍA

BIBLIOGRAFÍA BÁSICA

- Brassard & Bratley. Fundamentos de Algoritmia.

- Brassard & Bratley. Fundamentals of Algorithmics.
- Cormen, Leiserson, Rivest & Stein. Introduction to Algorithms.
- Manber. Introduction to Algorithms: A Creative Approach.
- Fridlender. Notas de Algoritmos y Estructuras de Datos II.
- Gramaglia. Algoritmos y Estructuras de Datos II: Guía de Ejercicios Resueltos.
- Tiraboschi. Algoritmos y Estructuras de Datos II: Guía de Ejercicios.

BIBLIOGRAFÍA COMPLEMENTARIA

- Balcázar. Programación Metódica.
- Biggs. Matemática Discreta.
- Blanco & Smith. Cálculo de Programas
- Kaldewaij. Programming: the Derivation of Algorithms.

METODOLOGÍA DE TRABAJO

El desarrollo de los contenidos se basa fuertemente en la modalidad de clase magistral, empleando en este tipo de actividad un 33% del tiempo de aula. En las mismas se estimula la participación de los alumnos a través del análisis de casos de estudio y preguntas, vinculando la actividad con la práctica.

Para el desarrollo de actividades prácticas (33%) se trabaja con una guía de ejercicios adoptando una modalidad de resolución grupal o individual según la preferencia del alumno, bajo la supervisión y orientación de los docentes y atendiendo a los tiempos de cada alumno o grupo de alumnos.

Las actividades de implementación en máquina se organizan con la modalidad de clase magistral seguida de trabajo grupal supervisado con entregas pautadas.

EVALUACIÓN

FORMAS DE EVALUACIÓN

Se toman tres evaluaciones parciales, uno para cada módulo principal de la materia. Los exámenes evalúan contenidos teóricos y prácticos.

Requisitos para obtener la regularidad

- Aprobación de dos de los tres exámenes parciales.

Requisitos para la aprobación:

- Evaluación escrita integral de la materia, de contenidos teóricos y prácticos. En caso de ser necesario, se toma un examen oral.

Promoción

- Aprobación de todos los parciales con nota mínima 6(seis) y promedio no inferior a 7(siete).