



Universidad
Nacional
de Córdoba



FAMAF
Facultad de Matemática,
Astronomía, Física y
Computación

EX-2024-00149385- -UNC-ME#FAMAF

PROGRAMA DE ASIGNATURA	
ASIGNATURA: Algoritmos y Estructuras de Datos II	AÑO: 2024
CARACTER: Obligatoria	UBICACIÓN EN LA CARRERA: 2° año 1° cuatrimestre
CARRERA: Licenciatura en Ciencias de la Computación	
REGIMEN: Cuatrimestral	CARGA HORARIA: 180 horas

ASIGNATURA: Algoritmos y Estructura de Datos	AÑO: 2024
CARACTER: Obligatoria	UBICACIÓN EN LA CARRERA: 3° año 1° cuatrimestre
CARRERA: Licenciatura en Matemática Aplicada	
REGIMEN: Cuatrimestral	CARGA HORARIA: 180 Horas.

FUNDAMENTACIÓN Y OBJETIVOS

Se pretende que el/la estudiante adquiera: capacidad para comprender y describir el problema que resuelve un algoritmo (el “qué”) y diferenciarlo de la manera en que lo resuelve (el “cómo”); capacidad para analizar algoritmos, compararlos según su eficiencia en tiempo y en espacio; capacidad y hábito de identificar abstracciones relevantes al abordar un problema computacional; familiaridad con técnicas de diseño de algoritmos de uso frecuente; familiaridad con la programación (en el lenguaje c, entre otros) de algoritmos y estructuras de datos, familiaridad con la utilización de diversos niveles de abstracción y lenguajes de programación.

CONTENIDO

1- Análisis de Algoritmos

Motivación

Problema de Ordenación. Diferentes maneras de ordenar. Ordenación por selección. El ciclo for. Conteo de operaciones de un programa. Esquemas de conteo. 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. 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.

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

EX-2024-00149385- -UNC-ME#FAMAF

intercalación. Resolución de la recurrencia.

Recurrencias

Recurrencias divide y vencerás. Formulación y resolución. Ejemplos. Demostración de la resolución de recurrencias divide y vencerás.

2- Estructuras 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. Tipos abstractos como concepto que surge de un problema a resolver. Chequeo de paréntesis balanceados: TAD Contador, operaciones, ecuaciones. Chequeo de delimitadores balanceados: TAD Pila, operaciones, ecuaciones. Representaciones posibles de contadores. Ejemplo: versión iterativa de la ordenación por intercalación usando una pila. Ejemplo: evaluación de expresiones en notación polaca inversa usando una pila. TAD Lista. Operaciones. Ecuaciones. Representaciones usando arreglos. Representaciones de pilas usando arreglos y listas. Transmisión de datos: 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. Diccionarios: 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.

3- 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

EX-2024-00149385- -UNC-ME#FAMAF

n-ésima de un número. Multiplicación de grandes números.

Backtracking

Motivación: algoritmo para salir de un laberinto. Problema de la moneda. Problema de la mochila. Problema de los caminos de costo mínimo.

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.

Recorrida de grafos y más backtracking

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 las ocho reinas. Podas graduales al grafo de búsqueda.

BIBLIOGRAFÍA

BIBLIOGRAFÍA BÁSICA

Fundamentos de Algoritmia, Gilles Brassard, Paul Bratley. Prentice-Hall, 1997.

Fundamentals of Algorithmics. Gilles Brassard, Paul Bratley. Prentice-Hall, 1995.

Introduction to Algorithms. Thomas Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein. Cambridge, 2009.

Introduction to Algorithms: A Creative Approach. Udi Manber. Addison-Wesley, 1989.

BIBLIOGRAFÍA COMPLEMENTARIA

Programación Metódica. José Luis Balcázar. McGraw-Hill, 1993.

Matemática Discreta. Norman L. Biggs. Vives V., 1998

Cálculo de Programas. Javier Blanco, Silvina Smith, Damián Barsotti. Universidad Nacional de Córdoba, 2008.

Programming: the Derivation of Algorithms. Anne Kaldewaij. Prentice-Hall, 1990.

EVALUACIÓN

FORMAS DE EVALUACIÓN

Se toman dos evaluaciones parciales (con sus respectivos recuperatorios) evaluando los contenidos teóricos y prácticos. Se evalúan dos trabajos prácticos de implementación en máquina (con sus respectivos recuperatorios).

El examen final consiste de una evaluación escrita teórico-práctica y una evaluación de implementación en máquina, es necesario aprobar ambas.

REGULARIDAD

- Aprobación de los dos parciales. En caso de desaprobado uno de ellos, debe aprobarse su recuperatorio.

- Aprobar los dos trabajos prácticos de implementación en máquina, o uno de ellos y el recuperatorio del otro.