



Planificaciones

7504 - Algoritmos y Programación II

Docente responsable: CALVO PATRICIA MABEL

OBJETIVOS

Que el alumno maneje de modo correcto el diseño de un TDA de acuerdo a las exigencias de una determinada situación problemática, dentro del paradigma de la orientación a objetos

Que el alumno aplique estrategias de solución de problemas complejos, definiéndolas en forma abstracta.

Que el alumno pueda plantear y desarrollar distintas implementaciones para un TDA eligiendo la más adecuada en función de su eficiencia temporal y espacial

Que el alumno aplique de modo conveniente los conceptos de complejidad computacional para garantizar la eficiencia del producto final.

CONTENIDOS MÍNIMOS

-

PROGRAMA SINTÉTICO

Unidad 1 Principios de la programación orientada a objetos: abstracción encapsulamiento, herencia, polimorfismo.

Unidad 2 El TDA Vector. TDA Lista. Primitivas e implementaciones diversas estáticas y dinámicas de cada TDA con orientación a objetos.

Unidad 3 TDA pila y clase pila, TDA cola y clase cola: implementaciones varias. Aplicaciones de pilas y colas.

Unidad 4 Complejidad de algoritmos. Concepto de Big Omicron, Big Omega; y Big Theta. Recurrencias básicas.

Unidad 5 Recursividad. Concepto y diseño de algoritmos recursivos. Tipos de recursividad. Recursividad 'de cola'. Estrategia 'Dividir y conquistar': análisis y aplicaciones. Métodos para la eliminación de la recursividad. Determinación de la complejidad de algoritmos recursivos.

Unidad 6 Métodos de búsqueda y ordenamiento de vectores: Búsqueda secuencial y binaria. Ordenamientos lentos, Shell, y rápidos (Mergesort, Quicksort). Análisis de la complejidad de cada uno.

Unidad 7 El TDA Conjunto. La estructura árbol binario de búsqueda. Balanceo de árboles. Árboles AVL. Backtracking.

El TDA Cola con prioridad y la implementación con árboles "heap". Ordenamiento Heap.

Unidad 8 Árboles multicamino. Árboles B. Concepto de árboles digitales. Tries.

Unidad 9 Grafos no dirigidos y dirigidos. Recorridos básicos, prueba de aciclicidad, recorridos topológicos.

Análisis de la estrategia voraz, Algoritmo de caminos mínimos de Dijkstra.

Estrategia de programación Dinámica. Algoritmo para la obtención de los valores de los caminos mínimos entre todos los pares de vértices (Floyd). Cerradura transitiva y Algoritmo de Warshall.

Árbol de expansión de coste mínimo, y algoritmos que permiten obtenerlo (Prim, Kruskal).

Unidad 10 Tablas de dispersión. Concepto de hashing y aplicaciones El radix sort.

PROGRAMA ANALÍTICO

Unidad 1:

Concepto de TDA: especificación e implementación. Introducción a la programación orientada a objetos. Concepto de clase. Principios de pruebas de programa. Mantenimiento de programas.

Unidad 2:

TDA Vector. TDA Lista. Primitivas e implementaciones diversas con arrays y con estructuras de listas

ligadas.
Plantillas de clases.

Unidad 3:
TDA pila , TDA cola: implementaciones varias con arrays y con estructuras de listas ligadas. Aplicaciones de pilas y colas.

Unidad 4:
Complejidad temporal y espacial de algoritmos. Medidas asintóticas: notación Big Omicron, Big Omega y Big Theta. Propiedades. Orden de complejidad de un algoritmo. Recurrencias básicas, resolución de diversos casos. El teorema Maestro.

Unidad 5:
Recursividad. Principios de la recursividad. Tipos de recursividad. Diseño de algoritmos recursivos. Pilas y recursividad. Recursividad de cola y métodos para la eliminación de la recursividad. Análisis de la estrategia "Dividir y conquistar" ("Divide and Conquer").

Unidad 6:
Búsqueda y Ordenamiento de vectores. Búsqueda secuencial. Búsqueda binaria. Ordenamiento Shell, Mergesort, Quicksort, Radix sort. Comparación de la eficiencia de distintos métodos de ordenamiento interno.

Unidad 7:
El TDA Conjunto. El TDA Diccionario. Primitivas e implementaciones diversas. Estructura de árbol binario de búsqueda. Balanceo de árboles. Árboles AVL. Implementación del TDA Conjunto en estructuras de árboles y en mapas de bits. El TDA Cola con prioridad y su implementación en montículos (árboles "Heap"). Ordenamiento por montículo (Heapsort). Estudio de la complejidad de los algoritmos utilizados en la clase árbol.

Unidad 8:
Árboles multicamino. Concepto de árbol B. Concepto de árbol digital (Trie). Aplicaciones para la implementación del TDA Conjunto y del TDA Diccionario.

Unidad 9;
Grafos no dirigidos y grafos dirigidos. Concepto, implementaciones. Recorridos en profundidad y en anchura. Aciclicidad, recorridos topológicos.
Algoritmo de caminos mínimos con un solo origen (Dijkstra), caminos mínimos entre todos los pares de vértices (Floyd), cerradura transitiva (Warshall).
Árbol de expansión de coste mínimo (algoritmos de Dijkstra, Kruskal y Prim). Análisis de la estrategia "Voraz"("Greedy") y de "Programación Dinámica".

Unidad 10:
Tablas de dispersión. El hashing: concepto, implementaciones, hashing abierto y cerrado; aplicaciones en la implementación del TDA Conjunto y del TDA Diccionario. Hashing sort.

BIBLIOGRAFÍA

1. 'Data structures and program design in C++' Kruse, Robert L., Ryba, Alexander J. Prentice-Hall
2. 'El Arte de Programar Computadoras' Knuth, Donald Volumen 1.
3. 'Construcción de Software orientado a objetos'. Meyer, Bertrand. Prentice Hall
4. 'El lenguaje de programación C++' Stroustrup, Bjarne Addison Wesley
5. 'Algoritmos y estructuras de datos' Weiss, M. Addison Wesley
6. 'Files structures' Folk, Zoellick Addison Wesley
7. 'C++ pointers and dynamic memory management' Daconta Wiley & Sons
9. 'Data abstraction and object-oriented programming in C++' Gorlen, Orlow, Plexico Wiley & Sons
10. 'C++, cómo programar'. Deitel y Deitel Pearson
11. 'Estructuras de Datos y Algoritmos', Aho, Hopcroft , Ullman
12. 'Técnicas de Diseño de Algoritmos' Guerequeta , Vallecillo, Univ de Málaga
13. 'The Practice of Programming', Kernighan
14. 'Estructuras de Datos, especificación, diseño e implementación' Franch, ediciones UPC
15. 'Thinking in C++', B. Eckel

16. Drozdek, A., "EStructuras de Datos y Algoritmos"

17. Cormen, T., Leiserson, C., Rivest, R., Stein C., "Introducción a los Algoritmos"

RÉGIMEN DE CURSADA

Metodología de enseñanza

Exposición de los temas del programa, realización de ejercicios tipo, resolución de situaciones problemáticas novedosas en forma individual y grupal, búsqueda y análisis de información en publicaciones y en la red.

Modalidad de Evaluación Parcial

Escrita. Resolver un conjunto de ejercicios teórico - prácticos.

CALENDARIO DE CLASES

Semana	Temas de teoría	Resolución de problemas	Laboratorio	Otro tipo	Fecha entrega Informe TP	Bibliografía básica
<1> 09/03 al 14/03	Introd POO	x	x			3,4,7,9
<2> 16/03 al 21/03	TDA Vector	x	x			1,4,7
<3> 23/03 al 28/03	TDA Lista	x	x		TP 0	1,4,7,8,9
<4> 30/03 al 04/04	TDA Pila-TDA Cola	x	x			1,4,7,9
<5> 06/04 al 11/04	Recursividad. Estrategia Divide y Vencerás	x				1,2,4,7
<6> 13/04 al 18/04	Complejidad Algoritmica	x			TP 1	2,3,8,9,10, 12,13
<7> 20/04 al 25/04	Complejidad de los principales métodos de Ordenamiento, búsqueda y de algoritmos diversos	x				1,2,37,10,11,12,13
<8> 27/04 al 02/05	TDA Conjunto. Arboles Generales y Binarios					1,7,8,9,10
<9> 04/05 al 09/05	Balanceo de árboles binarios. Arboles AVL. Arboles multivías. Arboles B. Conceptos de Backtracking	x				1,7,8,9,10
<10> 11/05 al 16/05	Arboles digitales (Tries) Colas con prioridad, árboles heap.	x			TP 2	1,7,8,9,10
<11> 18/05 al 23/05	Hashing	x	x			1,7
<12> 25/05 al 30/05	Grafos	x	x			1,5,6,7,9
<13> 01/06 al 06/06	Grafos, estrategias voraces y programación dinámica	x				1,5,6,7,9
<14> 08/06 al 13/06	Grafos		x		TP 3	1,5,6,7,9
<15> 15/06 al 20/06	Conceptos básicos de herencia y polimorfismo	x				1,14
<16> 22/06 al 27/06	Revisión y consultas	x				

CALENDARIO DE EVALUACIONES

Evaluación Parcial

Oportunidad	Semana	Fecha	Hora	Aula
1º	9	14/10	19:00	
2º	13	07/11	19:00	
3º	15	24/11	19:00	
4º				