



Planificaciones

7541 - Algoritmos y Programación II

Docente responsable: BUCHWALD MARTIN

OBJETIVOS

Que el estudiante, ante un problema, pueda indicar cuáles son los los Tipos de Datos Abstractos (TDAs) involucrados. Especificar correctamente las operaciones propias de cada TDA. Manipular e Implementar adecuadamente distintas estructuras de datos. Aplicar adecuadamente estrategias de resolución de problemas. Dadas las especificaciones de un TDA, pueda plantear diversas implementaciones. utilizar elementos del análisis de algoritmos para comparar los algoritmos correspondientes a las distintas implementaciones. Elegir la implementación más eficiente para un TDA. Justificar rigurosamente la elección de una implementación para un TDA. Elaborar productos de software adecuadamente documentados. Trabajar en equipo de manera proactiva y eficiente.

CONTENIDOS MÍNIMOS

Eficiencia de algoritmos. Métodos de diseño de algoritmos. Tipos Abstractos de Datos. Implementaciones eficientes de los Tipos Abstractos de Datos. Resolución de problemas usando Tipos Abstractos de Datos. Análisis de la eficiencia de esas soluciones.

PROGRAMA SINTÉTICO

Punteros y Memoria dinámica. Administración de memoria dinámica. Comparación con memoria estática. Cuestiones adicionales como: pasaje de funciones por parámetro, aritmética de punteros, piping y redirección de entrada, salida y errores. Herramientas (Valgrind, GDB, Makefile, uso Unix/Linux). Complejidad Algorítmica (tiempo, y también espacio). División y Conquista. Conceptos de Tipos de Datos Abstractos. TDAs Básicos: TDA Vector, Pila, Cola, Lista enlazada. Implementaciones sobre arreglos y estructuras enlazadas. Algoritmos de ordenamiento: Ordenamientos Comparativos. Ordenamientos no comparativos: Counting Sort, Radix Sort, Bucket Sort. Características de ordenamientos: estabilidad, in place TDA Conjunto y Diccionario. Tablas de Hash. Resolución de colisiones con Hashing Abierto y Cerrado (probing lineal, cuadrático, multiple hashing, etc). Cuckoo Hashing. Hopscotch Hashing. Hashing Perfecto. Árboles. Árboles Binarios. Recorridos sobre árboles. Algoritmos sobre árboles. Árboles Binarios de Búsqueda, como implementación de diccionarios. Árboles auto balanceados (AVL, aunque podría darse rojo negro). Árboles B y B+.

TDA Cola de Prioridad. Implementación con heaps binarios. Representación en árbol y en arreglo. Construcción de Heap a partir de un arreglo. Heapsort.

TDA Grafo. Aplicaciones. Operaciones y Representaciones (matriz de incidencia, matriz de adyacencia, listas de adyacencias, y combinaciones). Recorridos sobre Grafos. BFS. DFS. Ejemplos de uso de BFS y DFS. Orden topológico. Grafos bipartitos. Algoritmos sobre grafos: Componentes fuertemente conexas (Algoritmo de Tarjan). Puntos de Articulación, otros. Caminos mínimos: BFS y Algoritmo de Dijkstra. Algoritmo de Bellman-Ford. Backtracking y aplicaciones sobre grafos. Árboles de tendido mínimo. Algoritmos de Prim y Kruskal (TDA UnionFind). Flujo sobre una red: Algoritmo de Ford-Fulkerson.

PROGRAMA ANALÍTICO

Unidad 1: Introducción general.

Punteros y Memoria dinámica. Administración de memoria dinámica. Comparación con memoria estática. Cuestiones adicionales como: pasaje de funciones por parámetro, aritmética de punteros, piping y redirección de entrada, salida y errores. Herramientas (Valgrind, GDB, Makefile, uso Unix/Linux).

Unidad 2: Complejidad Algorítmica y Recursividad

Complejidad Algorítmica (tiempo, y también espacio). División y Conquista. Teorema Maestro.

Unidad 3: Tipos de Datos Abstractos.

Conceptos de Tipos de Datos Abstractos. TDAs Básicos: TDA Vector, Pila, Cola, Lista enlazada (y sus variantes). Implementaciones sobre arreglos y estructuras enlazadas. Análisis de complejidad de cada implementación y su implementación eficiente.

Unidad 4: Algoritmos de ordenamiento.

Ordenamientos Comparativos. Ordenamientos no comparativos: Counting Sort, Radix Sort, Bucket Sort. Características de ordenamientos: estabilidad, in place.

Unidad 5: TDA Conjunto y Diccionario, Hashing y Hashes.

TDA Conjunto y Diccionario. Tablas de Hash. Resolución de colisiones con Hashing Abierto y Cerrado (probing lineal, cuadrático, multiple hashing, etc). Cuckoo Hashing. Hopscotch Hashing. Hashing Perfecto.

Unidad 6: Árboles e Implementación de diccionarios con Árboles.

Árboles Binarios. Recorridos sobre árboles. Algoritmos sobre árboles. Árboles Binarios de Búsqueda, como implementación de diccionarios. Árboles auto balanceados (AVL). Árboles B y B+.

Unidad 7: TDA Cola de Prioridad.

Implementación con heaps binarios. Operaciones de heaps. Representación en árbol y en arreglo. Construcción de Heap a partir de un arreglo (heapify). Heapsort.

Unidad 8: TDA Grafo.

Aplicaciones. Operaciones y Representaciones (matriz de incidencia, matriz de adyacencia, listas de adyacencias, y combinaciones). Recorridos sobre Grafos. BFS. DFS. Ejemplos de uso de BFS y DFS. Orden topológico. Grafos bipartitos. Componentes fuertemente conexas (Algoritmo de Tarjan). Puntos de Articulación, otros.

Unidad 9: Algoritmos sobre Grafos.

Caminos mínimos: BFS, Algoritmo de Dijkstra y Algoritmo de Bellman-Ford. Árboles de tendido mínimo: Algoritmos de Prim y Kruskal (TDA UnionFind). Estrategia Backtracking, y aplicaciones sobre grafos (Coloreo, Camino Hamiltoniano, otros). Flujo sobre una red: Algoritmo de Ford-Fulkerson.

BIBLIOGRAFÍA

- Weiss, Mark Allen: "Estructuras de datos y algoritmos", Addison-Wesley Iberoamericana, 1995.
- Cormen, Thomas H.; Leiserson Charles E.; Rivest Ronald L; Stein Clifford: "Introduction to Algorithms", The MIT Press, 2009.
- Kleinberg, Jon; Tardos, Éva: "Algorithm Design", Addison-Wesley, 2005.
- Kruse, Robert Leroy; Tondon, Clovis L.; Leung, Bruce P: "Data structures and program design in C", Prentice-Hall, 1997.
- Brassard, Gilles; Bratley, Paul: "Algorithmics: theory and practice", Prentice-Hall, 1988.
- Sedgewick, Robert: "Algorithms in C++", Addison-Wesley, 1992.
- Tenenbaum, Aaron M.; Langsam, Yedidyah; Augenstein, Moshe J.: "Estructuras de datos en C", Prentice-Hall Hispanoamericana, 1993.
- Kernighan, Brian W.; Ritchie, Dennis M.: "El lenguaje de programación C", Prentice-Hall Hispanoamericana, 1991.

RÉGIMEN DE CURSADA

Metodología de enseñanza

Clases teórico-prácticas con trabajos prácticos a realizar por los alumnos. Resolución de problemas por parte de los alumnos controlada por los docentes. Elaboración de la mayoría de las estructuras de datos explicadas en clase, en conjunto con trabajos prácticos que evalúen el entendimiento tanto del lado del implementador como del usuario de un TDA.

Modalidad de Evaluación Parcial

Durante el curso deberán rendirse 3 (tres) evaluaciones parciales (denominadas, parcialitos). En la suma de los 3, se evalúan todos los contenidos de la cursada, y se dividen en esta cantidad para aligerar la carga sobre los alumnos. Los parcialitos pueden recuperarse incluso habiendo aprobado, sin riesgo a empeorar la nota (queda la mejor nota entre todas las veces que se hubo rendido).

Modalidad de Evaluación Integradora

Los alumnos que hayan aprobado la cursada (TPs y parcialitos) deberán rendir una evaluación escrita integradora en donde se pedirá que manejen conceptos, aplicación de conocimientos y dominio de técnicas, mediante preguntas y resolución de problemas.

Aquellos alumnos que hayan alcanzado un desempeño excepcional no deberán rendir el examen integrador.

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	Presentación de la materia, régimen de cursada, bibliografía. Conceptos de memoria estática y dinámica. Repaso de Recursividad. División y conquista. Merge sort.	Introducción a la materia. Introducción a la programación en C. Tipos, funciones. Introducción a las estructuras de control de flujo de C. Manejo de vectores y punteros.				Weiss 10.2. Cormen 4. Weiss 7.6.
<2> 16/03 al 21/03	Cálculo de eficiencia, notación O. Ejemplos iterativos y recursivos. Resolución analítica. Quick sort. Teorema Maestro. Ejemplos	Memoria dinámica en C: declaración de estructuras y manejo con memoria dinámica. Cadenas. Debugging. Tipos abstractos de datos. Archivos .h Vector dinámico.			Entrega TP0	Weiss 2. Cormen 2 y 3. Weiss 7.7. Cormen 4.5
<3> 23/03 al 28/03	Feriado Abstracción. Tipos abstractos de datos. Vector Dinámico, Pilas y Colas. Aplicaciones de Pilas y Colas.	Feriado Uso de void**. TDA Pila.			Entrega VD	Weiss 3.6 y 3.7. Cormen 10.1.
<4> 30/03 al 04/04	Implementaciones sobre Arreglos y Estructuras Enlazadas. Tipos de Estructuras enlazadas. Distintos tipos de listas. Sort por Comparación son $O(n \log n)$. Sorts con más información: ordenamientos no comparativos. Counting Sort. Radix Sort	Implementación de cola sobre nodos enlazados. Punteros a función. Ejercitación de TDAs Pila y Cola. Análisis de eficiencia. Teorema del maestro. Ejercitación. Caso de estudio: padrón electoral.			Entrega Pila dinámica	Weiss 3. Cormen 10. Weiss 7.8, 7.9 y 7.11. Cormen 8.
<5> 06/04 al 11/04	Ordenamientos no comparativos, continuación: BucketSort. TDA Conjunto y Diccionario. Implementaci	Listas enlazadas. Iteradores externos. Ejercitación sobre listas enlazadas. Feriado			Entrega Cola Enlazada	Weiss 7.11. Cormen 8. Weiss 5. Cormen 11.

Semana	Temas de teoría	Resolución de problemas	Laboratorio	Otro tipo	Fecha entrega Informe TP	Bibliografía básica
	ones: Mapa de bits, tablas de hashing. Funciones de Hashing. Resoluciones de colisiones (abierto y cerrado). Feriado					
<6> 13/04 al 18/04	Otros métodos de resolución de colisiones: Coochoo hashing. Hopscotch hashing. Hashing Perfecto 1er Parcialito: Complejidad, División y Conquista, TDAs, Pilas, Colas y Listas	Implementación de lista enlazada. Iteradores internos. Ordenamientos no comparativos.			Entrega Lista Enlazada	Cormen 11.5
<7> 20/04 al 25/04	Árboles. Árboles binarios. Recorridos sobre árboles. ABB como implementación de diccionarios. Propiedad de ABB. Borrados.	Archivos en C, entrada y salida. Diccionarios: hashing cerrado y hashing abierto. Funciones de hashing. Flujos en C. Detalles de implementación de hash: Eliminación, Iteradores.				Weiss 4.1 y 4.2 Weiss 4.3. Cormen 12.
<8> 27/04 al 02/05	Árboles autobalanciados. AVL. Feriado	Árboles como estructuras enlazadas. Árboles binarios de búsqueda. Feriado			Entrega Hash	Weiss 4.4.
<9> 04/05 al 09/05	Otros tipos de árboles: árboles B, B+. Colas de prioridad. Heaps.	Árboles binarios de búsqueda: iteradores internos y externos. Iteración por rangos. Árboles AVL. Primitivas del TAD ABB.				Weiss 4.7. Cormen 18. Weiss 6. Cormen 6.5, 6.1 y 6.2.
<10> 11/05 al 16/05	Heapify. Heapsort. Comparación de ordenamientos. TDA Grafo. Características. Ejemplos de uso. Operaciones sobre grafo mínimo. Representaciones.	Repaso de heaps. Downheap, upheap. Primitivas del heap. Detalles de implementación. Algoritmo Top-K. Ejercitación de Ordenamientos y Hashes			Entrega TP1 y ABB	Weiss 7.5. Cormen 6.3 y 6.4. Weiss 9.1. Cormen 22.1.

Semana	Temas de teoría	Resolución de problemas	Laboratorio	Otro tipo	Fecha entrega Informe TP	Bibliografía básica
<11> 18/05 al 23/05	Recorridos sobre Grafos. BFS, DFS. Ejemplos de uso de BFS y DFS. 2do Parcialito (Ordenamientos, Hashes, Árboles y Heaps)	Ejercitación de Árboles y Heaps Grafos: Terminología. Representaciones. Órdenes de cada operación				Cormen 22.2 y 22.3.
<12> 25/05 al 30/05	Feriado Ejemplo de recorridos DFS y BFS. Orden topológico.	Feriado Recorrido de grafos en anchura (BFS) y profundidad (DFS). Grafos bipartitos.			Entrega Heap	Weiss 9.2. Cormen 22.4.
<13> 01/06 al 06/06	Caminos mínimos: BFS. Algoritmo de Dijkstra. Algoritmo de Bellman-Ford. MST: Algoritmo de Prim. Algoritmo de Kruskal y Conjuntos disjuntos.	Componentes Fuertemente Conexas: Algoritmo de Tarjan. Puntos de Articulación. Caminos mínimos: algoritmo de Dijkstra. Ejercitación de Camino Mínimo			Entrega TP2	Weiss 9.3. Cormen 24. Weiss 9.5. Cormen 23.
<14> 08/06 al 13/06	Backtracking. Aplicación sobre Grafos. Ciclo de largo N. Camino Hamiltoniano. Flujo en una red. Algoritmo/Método de Ford-Fulkerson.	Ejercitación sobre los algoritmos de Prim y Kruskal. Ejercitación de conceptos de Árboles de tendido mínimo. Ejercitación de Backtracking. - Coloreo - TSP - Laberinto				Weiss 10.5. Weiss 9.4. Cormen 26.1 y 26.2.
<15> 15/06 al 20/06	Feriado Continuación de Flujo en una red. Ejercitación. Matching en Grafos Bipartitos. Técnicas de Diseño de Algoritmos: Introducción a Algoritmos Greedy y Programación Dinámica. Backtracking en caso general.	Feriado Ejercitación sobre Maximización de Flujo				Weiss 9.4. Cormen 26.2 y 26.3. Cormen 15 y 16.
<16> 22/06 al 27/06	Problemas avanzados con grafos 3er Parcialito: Grafos	Ejercitación 4to Parcialito. Cierre de la materia.			Entrega TP3	

CALENDARIO DE EVALUACIONES

Evaluación Parcial

Oportunidad	Semana	Fecha	Hora	Aula
1º	6	17/04	17:30	
2º	11	22/05	17:30	
3º	16	26/06	17:30	
4º				
Otras observaciones				
Las fechas corresponden a las fechas de los 3 parcialitos				