

CONTENIDO TEMÁTICO

Ordenar temas utilizando codificación decimal

1. Complejidad
 - 1.1. Orden de una sucesión.
 - 1.2. Notación $O()$. Definición. Propiedades.
 - 1.3. Aplicación al análisis de algoritmos. Ejemplos.
 - 1.4. Teorema maestro para el cálculo de orden en recurrencias. Aplicación del teorema.
2. Estructuras de Datos Avanzadas
 - 2.1. Árboles binarios de búsqueda. Operaciones de búsqueda, inserción y borrado. Ventajas y limitaciones.
 - 2.2. Árboles balanceados. Árboles rojo-negro. Invariante roja-negra. Propiedades.
 - 2.3. Heaps Binomiales. Invariante binomial. Operaciones sobre heaps. Propiedades.
 - 2.4. Heaps Fibonacci. Ventajas con respecto a otras implementaciones.
 - 2.5. Colas Brodal e implementaciones cache-oblivious. Problemática y consideraciones prácticas.
3. Estrategias de Diseño de Algoritmos
 - 3.1. Algoritmos con backtracking. Búsquedas en profundidad y en amplitud.
 - 3.2. Programación Dinámica. Caracterización de la subestructura óptima.
 - 3.3. Algoritmos Greedy. Definición de matroide. Modelado del problema utilizando matroides.
 - 3.4. Algoritmos sobre grafos. Definición de semianillos. Resolución usando semianillos. Semianillos para camino más corto, existencia de camino, cantidad de caminos.
4. Análisis de Costo amortizado
 - 4.1. Motivación.
 - 4.2. Método básico.
 - 4.3. Método del banquero.
 - 4.4. Método del físico.
 - 4.5. Aplicación: tablas dinámicas.
5. Representación de Conjuntos Disjuntos
 - 5.1. Representación con listas. Limitaciones de la representación.
 - 5.2. Representación con árboles. Optimización usando unión según rango y compresión de caminos.
 - 5.3. Análisis de complejidad de las diferentes variantes.

TRABAJOS PRÁCTICOS

a) Enumeración:

Ejercicios de práctica sobre los siguientes temas:

1. Cálculo de orden de una sucesión. Sucesión correspondiente a una recurrencia. Aplicación del teorema Maestro.
2. Implementación y cálculo de complejidad sobre estructuras arbóreas.
3. Diseño de soluciones a problemas utilizando heaps.
4. Solución de problemas utilizando programación dinámica, algoritmos greedy. Solución de problemas en grafos usando semianillos.
5. Cálculo de costo amortizado de algoritmos.
6. Resolución de problemas usando Set-Union-Find.

Trabajos prácticos:

1. Trabajo práctico nº 1

Programación gráfica de transformaciones usando matrices.

2. Trabajo práctico nº 2

Implementaciones eficientes de heaps.

b) Guías de trabajos prácticos publicadas: (con su código de publicación)

BIBLIOGRAFÍA

a) Adecuada al programa. Ordenada por temas y con su codificación de biblioteca, incluidas las publicaciones de la Cátedra con su código de publicación.

1. *Introduction to Algorithms*. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein. The MIT Press 2009.
2. *Algorithms and Data Structures: The Basic Toolbox*. Kurt Mehlhorn, Peter Sanders. Springer 2008.
3. *Data Structures and Algorithms*. Alfred V. Aho, Jeffrey D. Ullman, John E. Hopcroft. Addison Wesley 1983.
4. *Purely Functional Data Structures*. Chris Okasaki. Cambridge UP 1999

b) Complementaria para profundización o extensión de temas.