

Algoritmos y Estructuras de Datos III - 2c/2022

Contenidos de las clases teóricas

Práctica 1: Técnicas de diseño de algoritmos

1. Repaso de complejidad computacional ([3], Secciones 1.2 y 1.3). Máquina RAM. Notación “O grande” ([2], Sección 3.1).
2. Algoritmos de fuerza bruta. Backtracking. Problema de las n damas. Resolución de sudokus.
3. Programación dinámica ([2], Capítulo 15). Algoritmos de programación dinámica para el cálculo de coeficientes binomiales, para el problema de la mochila, para multiplicación de n matrices minimizando las operaciones en punto flotante, y para el problema de subsecuencia común más larga.
4. Heurísticas. Algoritmos golosos ([2], Secciones 16.1 y 16.2). Algoritmos golosos para el problema de la mochila, para el problema de distribución de cambio, y para el problema de tiempo promedio de espera en un sistema con un único servidor.
5. Algoritmos probabilísticos.

Práctica 2: Introducción a la teoría de grafos

1. Grafos. Definiciones básicas. Representación de grafos ([2], Sección 22.1).
2. Grafos bipartitos.
3. Conexión y recorrido de grafos. Algoritmos BFS y DFS recursivos e iterativos. Propiedades de estos algoritmos ([2], Secciones 22.2 y 22.3; [2], Capítulo 15 por referencias de programación dinámica).
4. Orden topológico ([1], Sección 3.4).
5. Algoritmo de Kosaraju.

Práctica 3: Problema de árbol generador mínimo

1. Algoritmos de Prim y Kruskal ([2], Capítulo 23 y [1], Secciones 13.1 a 13.5). Colas de prioridad ([2], Capítulo 19) y estructura de datos union-find ([2], Capítulo 21; [2], Capítulo 16 por referencias a algoritmos golosos).
2. Camino minimax uno a todos.

Práctica 4: Problemas de camino mínimo ([2], Capítulos 24 y 25)

1. Clasificación de problemas de camino mínimo. Ejemplos de modelado ([1], Secciones 4.1 y 4.2).
2. Existencia de caminos mínimos en un grafo.
3. Algoritmo de Dijkstra ([1], Secciones 4.3 y 4.5).

4. Algoritmo de Bellman-Ford.
5. Resolución de sistemas de diferencias.
6. Algoritmo de Floyd-Warshall ([1], Sección 5.6).
7. Camino mínimo sobre grafos acíclicos ([1], Sección 4.3).
8. Algoritmo de Johnson.

Práctica 5: Problemas de flujo en redes

1. Problema de flujo máximo. Certificado de optimalidad y concepto de red residual. Algoritmo de Ford-Fulkerson y sus propiedades ([1], Secciones 6.1 a 6.6). Algoritmo de Edmonds-Karp ([1], Secciones 7.1 y 7.4). Matching máximo en grafos bipartitos como problema de flujo máximo ([1], Sección 12.3).
2. Problema de flujo de costo mínimo. Certificado de optimalidad y concepto de red residual. Algoritmo de Klein ([1], Secciones 3.5, 9.1, 9.3 y 9.6).
3. Introducción a la programación lineal. Algoritmos para su resolución. Expresión de los problemas polinomiales vistos en la materia como modelos de programación lineal. Dualidad ([1], Anexos C.1, C.2 y C.6).

Práctica 6: Introducción a la teoría de NP-completitud

1. Problemas de optimización, evaluación, localización y decisión. Problema del viajante de comercio.
2. Máquinas de Turing determinísticas y no determinísticas. Complejidad y corrección en ambos modelos de cómputo ([3], Secciones 2.1 a 2.3).
3. Clases P y NP ([3], Sección 2.4). Ejemplos de problemas en NP.
4. Transformaciones polinomiales. Clases NP-completo y NP-hard ([3], Sección 2.5). Consecuencias teóricas y prácticas.
5. Teorema de Cook-Levin ([3], Sección 2.6).
6. NP-completitud de 3-SAT, de CLIQUE y de problemas relacionados ([3], Sección 3.1). Otros problemas NP-completos.
7. Restricción y extensión de un problema ([3], Sección 3.2.1).
8. Clase co-NP ([3], Sección 7.1).

References

- [1] R. Ahuja, T. Magnanti y J. Orlin, *Network flows: theory, algorithms, and applications*. Prentice-Hall, 1993.
- [2] T. Cormen, C. Leiserson, R. Rivest y C. Stein, *Introduction to algorithms*. The MIT Press, 2009. Third edition.
- [3] M. Garey y D. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, 1979.