

MAC338 - Lista de Exercícios

02/06/2006

Para fazer estes exercícios, você deve ler o capítulo correspondente do CLRS.

1 Programação Dinâmica (CLRS, Capítulo 15)

1. (CLRS 15.2-1) Encontre a maneira ótima de fazer a multiplicação das matrizes cujas dimensões são (5, 10, 3, 12, 5, 50, 6).
2. (CLRS 15.2-5) Mostre que são necessários exatamente $n - 1$ pares de parênteses para especificar exatamente a ordem de multiplicação de um produto com n fatores.
3. (CLRS 15.3-2) Desenhe a árvore de recursão para o algoritmo MERGESORT aplicado a um vetor de 16 elementos. Por que a técnica de programação dinâmica não é capaz de acelerar o algoritmo?
4. (CLRS 15.4-1) Determine a subsequência comum máxima de

(1, 0, 0, 1, 0, 1, 0, 1)

e

(0, 1, 0, 1, 1, 0, 1, 1, 0).

5. (CLRS 15.4-5) Mostre como o algoritmo da subsequência comum máxima pode ser usado para resolver o problema da subsequência crescente máxima de uma sequência numérica. Dê uma delimitação justa, em notação Θ do consumo de tempo de sua solução.

2 Algoritmos Gulosos (CLRS, Capítulo 16)

Problema dos Intervalos disjuntos: Dado um conjunto de intervalos $[s_1, f_1), \dots, [s_n, f_n)$ encontrar uma coleção máxima de intervalos disjuntos dois a dois. (Obs: Uma definição mais detalhada do problema se encontra no CLRS sob o nome de *Activity-selection problem*).

1. (CLRS 16.1-1) Escreva e explique um algoritmo guloso para resolver o problema dos intervalos disjuntos.
2. (CLRS 16.1-1) Escreva e explique um algoritmo de programação dinâmica para resolver o problema dos intervalos disjuntos.
3. (CLRS 16.1-1) Compare o tempo de execução dos algoritmos dos exercícios anteriores.

3 Estruturas de Dados para Famílias de Conjuntos Disjuntos (CLRS, Capítulo 21)

1. (CLRS 21.3-3) Dê uma seqüência de operações MAKESET, UNION, e FINDSET com m operações no total, n das quais MAKESET, que consome tempo $\Omega(m \log n)$. Aqui, supomos que usamos a estrutura *disjoint-set forest*, com *union by rank*.
2. (CLRS 21.4-2) Suponha que usamos a estrutura *disjoint-set forest*, com *union by rank* mas sem compressão de caminhos, para implementar MAKESET, UNION, e FINDSET. Prove que para todo elemento x , sempre temos $rank[x] \leq \log_2 n$, onde n é o número de MAKESET executados.

4 Análise Amortizada (CLRS, Capítulo 17)

1. CLRS 17.1-3
2. CLRS 17.2-2
3. Faça o Problema CLRS 17-2.

5 Hashing (CLRS, Capítulo 11)

1. CLRS 11.2-1
2. CLRS 11.2-2
3. CLRS 11.3-1
4. CLRS 11.3-5
5. CLRS 11.3-6

6 Complexidade (CLRS, Capítulo 34)

1. CLRS 34.1-4
2. CLRS 34.2-7
3. CLRS 34.2-8
4. CLRS 34.2-10