MAC 338 - Análise de Algoritmos

Departamento de Ciência da Computação

Primeiro semestre de 2005

Lista 2

---

1. Resolva as recorrências abaixo. Para todas elas, suponha que $T(1) = \Theta(1)$.

   (a) $T(n) = 8T(n/2) + \Theta(n^2)$

   (b) $T(n) = 2T(n/2) + \Theta(n^3)$

   (c) $T(n) = 7T(n/3) + \Theta(n^2)$

   (d) $T(n) = T(\lfloor 9n/10 \rfloor) + \Theta(n)$

   (e) $T(n) = 2T(n/4) + \sqrt{n}$

2. Suponha que desejamos determinar um segmento não-vazio com soma o mais próxima de zero possível em vez de um com soma máxima. Projete um algoritmo para esta tarefa. Qual é o consumo de tempo do seu algoritmo? E se estivéssemos interessados em um algoritmo para encontrar um segmento de soma mais próxima de um dado valor?

3. No problema do retângulo de soma máxima, é dada uma matriz $A_{m \times n}$ de números inteiros e deseja-se determinar um retângulo de soma máxima. Projete um algoritmo para este problema. Qual o consumo de tempo do seu algoritmo?

4. Escreva um algoritmo que ordena uma lista de n itens dividindo-a em três sublistas de aproximadamente $n/3$ itens, ordenando cada sublista recursivamente e intercalando as três sublistas ordenadas. Analise seu algoritmo concluindo qual é o seu consumo de tempo.

5. Escreva um algoritmo eficiente que busca um valor x em uma matriz $A_{m \times n}$ cujas linhas e colunas estão ordenadas em ordem não-decrescente.

6. Escreva um algoritmo de divisão e conquista que determina o maior valor que aparece em um vetor. Escreva uma recorrência que descreve o número de comparações envolvendo elementos do vetor que tal algoritmo faz. Resolva a recorrência e compare o resultado com o número de comparações efetuada pelo algoritmo óbvio de cálculo do máximo de um vetor.

7. Considere o seguinte problema: dados n, uma seqüência de números reais $a_n, \ldots, a_0$ e um número real x, determinar o valor do polinômio $p_n(x) = a_n x^n + \cdots + a_1 x + a_0$. Pode-se resolver esse problema calculando o valor de $q_{n-1}(x) = a_n x^{n-1} + \cdots + a_2 x + a_1$ e, posteriormente, $p_n(x) = x\, q(x) + a_0$. Esse método é chamado de *regra de Horner*. Projete um algoritmo de divisão e conquista para resolver o problema e compare-o ao método derivado da regra de Horner. Quantas adições e quantas multiplicações faz o seu algoritmo?