

MAC 338 - Análise de Algoritmos

Departamento de Ciência da Computação

Primeiro semestre de 2013

Lista 1

1. Usando a definição de notação O , prove que

(a) $n^2 + 10n + 20 = O(n^2)$

(b) $\lceil n/3 \rceil = O(n)$

(c) $\lg n = O(\log_{10} n)$

(d) $n = O(2^n)$

(e) $n/1000$ não é $O(1)$

(f) $n^2/2$ não é $O(n)$

(Lembre-se que $\lg n$ denota o logaritmo na base 2 de n .)

2. Prove ou dê um contra-exemplo para as afirmações abaixo:

(a) $\lg \sqrt{n} = O(\lg n)$

(b) Se $f(n) = O(g(n))$ e $g(n) = O(h(n))$ então $f(n) = O(h(n))$.

(c) Se $f(n) = O(g(n))$ e $g(n) = \Theta(h(n))$ então $f(n) = \Theta(h(n))$.

(d) Suponha que $\lg(g(n)) > 0$ e que $f(n) \geq 1$ para todo n suficientemente grande. Neste caso, se $f(n) = O(g(n))$ então $\lg(f(n)) = O(\lg(g(n)))$.

(e) Se $f(n) = O(g(n))$ então $2^{f(n)} = O(2^{g(n)})$.

3. Prove que

(a) $\sum_{i=1}^n i^k$ é $\Theta(n^{k+1})$

(b) $\sum_{i=1}^n \frac{1}{2^i} \leq 2$.

4. Resolva as recorrências abaixo.

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

(b) $T(n) = 8T(\lfloor n/2 \rfloor) + \Theta(n^2)$

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

(d) $T(n) = 7T(\lfloor n/3 \rfloor) + \Theta(n^2)$

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

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 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.

7. Sejam $X[1..n]$ e $Y[1..n]$ dois vetores, cada um contendo n números ordenados. Escreva um algoritmo $O(\lg n)$ para encontrar uma das medianas de todos os $2n$ elementos nos vetores X e Y .

8. Seja $X[1..n]$ um vetor de inteiros e i e j dois índices distintos de X , ou seja, i e j são inteiros entre 1 e n . Dizemos que o par (i, j) é uma *inversão* de X se $i < j$ e $X[i] > X[j]$. Escreva um algoritmo $O(n \lg n)$ que devolva o número de inversões em um vetor X , onde n é o número de elementos em X .