UNIVERSIDADE DE SÃO PAULO

INSTITUTO DE CIÊNCIAS MATEMÁTICAS E DE COMPUTAÇÃO

DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

SCC0505 – Introdução à Teoria da Comutação – 1o. semestre de 2011
Profa. Graça Nunes

Lista de Exercícios
Análise de Algoritmos
Notação Assintótica

1. Considere a ordenação de n números armazenados no arranjo A, localizando primeiro o menor elemento de A e permutando esse elemento contido em A[1]. Em seguida, encontre o segundo menor elemento de A e o troque pelo elemento A[2]. Continue dessa dessa maneira para os primeiros n-1 elementos de A. Escreva o pseudocódigo para esse algoritmo conheicido como ordenação por seleção. Faça a análise do algoritmo. Por que ele só precisa ser executado para os primeiros n-1 elementos, e não para todos os elementos? Fornaça os tempos de execução do melhor caso e do pior caso da ordenação por seleção.

2. Vamos supor que estamos comparando implementações de ordenação por inserção e ordenação por intercalação na mesma máquina. Para entradas de tamanho *n,* a ordenação por inserção é executada em $8n^2$ etapas, enquanto a ordenação por intercalação é executada em *64n lg n* etapas. Para que valores de *n* a ordenação por inserção supera a ordenação por intercalação?

3. Expresse a função $n^3/1000 - 100n^2 - 100n + 3$ em termos da notação *O.*

4. Sejam *f(n)* e *g(n)* funções assintoticamente não negativas. Usando a definição básica da notação O, prove que *max(f(n), g(n)) = O(f(n)+g(n))*.

5. É verdade que $2^{n+1} = O(2^n)$? É verdade que $2^{2n} = O(2^n)$?

6. Demostre que para duas funções quaisquer *f(n)* e *g(n),* temos $f(n) = \Theta(g(n))$ se e somente se $f(n) = O(g(n))$ e $f(n) = \Omega(g(n))$.

7. No problema de pesquisa de um valor v em uma seqüência *A={a1, a2, ... ,an},* observe que, se a seqüência A estiver ordenada, poderemos comparar o ponto médio da seqüência com o valor v e eliminar metade da seqüência de consideração posterior. A **pesquisa binária** é um algoritmo que repete esse procedimento, dividindo ao meio o tamanho da porção restante da seqüência a cada vez. Escreva o pseudocódigo para pesquisa binária. Demonstre que o tempo de execução do pior caso da pesquisa binária é *O(lg n).*

8. Descreva um algoritmo de tempo *O(n lg n)* que, dado um conjunto *S* de *n* elementos e outro inteiro *x*, determine se existem ou não dois elementos em *S* cuja soma seja exatamente *x*.

9. Comportamento assintóticos de polinômios:

   Seja $p(n) = \sum_{i=0}^{d} a_i n^i$, onde $a_d > 0$, um polinômio de grau *d* em *n,* e seja *k* uma constante. Use as definições das notações assintóticas para provar as propriedades a seguir:

   - Se $k >= d$, então $p(n) = O(n^k)$.
   - Se $k <= d$, então $p(n) = \Omega(n^k)$.
   - Se $k = d$, então $p(n) = \Theta(n^k)$.
   - Se $k > d$, então $p(n) = o(n^k)$.
   - Se $k < d$, então $p(n) = \omega(n^k)$.

10. Mostre que:

    - $1/2n(n+1)$ é $O(n^2)$;
    - $n + \sqrt{n}$ é $O(n)$;
    - $n/1000$ não é $O(1)$;
    - $1/2n^2$ não é $O(n)$;
    - $1/2n^2 - 3n$ é $O(n^2)$.