

## Algoritmos para Automação e Sistemas

### Primeira Lista de Exercícios

1) Implemente os algoritmos da ordenação por inserção e intercalação (vistos na sala de aula) usando a linguagem de programação C ou C++. Mensure e compare o tempo de execução de cada algoritmo considerando um vetor com 1K, 10K, 100K e 1M para os seguintes casos:

- Os elementos do vetor já estão ordenados.
- Os elementos do vetor estão em ordem inversa.
- Alguns elementos do vetor não estão ordenados

2) Determine o tempo de execução  $T(n)$  do algoritmo do *bubble sort* usando as técnicas de resolver somatórios:

```
1 para i = 1 até n
2 faça para j = n até i+1
3 faça se A[j] < A[j-1]
4 então trocar A[j] ↔ A[j-1]
```

- Qual é o tempo de execução do melhor e pior caso deste algoritmo?
- Como ele se compara ao tempo de execução da ordenação por inserção?
- Enuncie um loop invariante para o loop das linhas 1 a 4.
- Mensure o tempo de execução do algoritmo considerando um vetor com 1K, 10K, 100K e 1M para os casos a), b) e c) descritos na questão 1).

3) Considere o seguinte pseudocódigo para pesquisa linear (visto na sala de aula). Usando um loop invariante, prove que este algoritmo é correto. Certifique-se que o seu loop invariante satisfaz às três propriedades necessárias (isto é, inicialização, manutenção e término).

```
j=1
while j<=length(A) and A[j]!=q
do j++
if j<=length(A) then return j
else return NIL
```

4) Considere o problema de somar, subtrair, multiplicar e dividir dois inteiros binários de  $n$  bits, armazenado em dois arranjos de  $n$  elementos A e B. A soma, subtração, multiplicação e divisão dos dois inteiros deve ser armazenada em forma binária em um arranjo de  $(n+1)$  elementos C. Escreva um pseudocódigo para somar, subtrair, multiplicar e dividir os dois inteiros. Qual é o tempo de execução assintótico do seu código para cada uma das operações aritméticas?

5) Para cada função  $f(n)$  e cada tempo  $t$  na tabela a seguir, determine o maior tamanho  $n$  de um problema que pode ser resolvido no tempo  $t$ , supondo-se que o algoritmo para resolver o problema demore  $f(n)$  microssegundos.

	1 segundo	1 minuto	1 hora	1 dia	1 mês	1 ano	1 século
$\lg n$							
$n^{1/2}$							
$n$							
$n \lg n$							
$n^2$							
$n^3$							
$2^n$							
$n!$							
$n^n$							

6) Verifique se as seguintes afirmativas estão corretas mostrando as respectivas provas matemáticas (Dica: use a definição da notação):

- No pior caso, o algoritmo do *bubblesort* (desenvolvido no exercício 2) é  $\Theta(n^2)$
- $\Theta(n^2) + \Theta(n) + \Theta(1) = \Theta(n^2)$
- $O(n^3) + O(n \lg n) + O(n^2) = O(n^3)$
- $n^3/100 - 100n^2 - 100n + 3 = \Theta(n^3)$
- $O(n) \times O(n) = O(n^2)$
- $\Theta(n) + \Theta(m) = \Theta(n+m)$
- Quaisquer constantes reais  $a$  e  $b$ , onde  $b > 0$ ,  $(n+a)^b = \Theta(n^b)$

- 7) Dado um vetor ordenado, como você encontra a localização de inteiro  $x$ ? Qual é a complexidade da sua solução?
- 8) Descreva um algoritmo de tempo  $\Theta(n \lg n)$  que, dado um conjunto  $S$  de  $n$  inteiros e outro inteiro  $x$ , determine se existem ou não dois elementos em  $S$  cuja soma seja exatamente  $x$ .
- 9) Sem utilizar uma calculadora, quantos zeros estão no final do “100!”? Lembre que  $100! = 100 * 99 * 98 * \dots * 3 * 2 * 1$ .
- 10) Escreva uma função para inverter a ordem das palavras em uma *string*.

**Data de entrega: 05 de maio de 2015 (terça-feira).**

**Após esta data será descontado 2 pontos por dia de atraso.**

**A lista de exercícios deve ser resolvida e entregue individualmente.**

24/04/2015