

### **Lista de Exercícios 2 – 5 pontos**

Disciplina	Entrega 1	Entrega 2
Projeto e Análise de Algoritmos	Q1, Q2, Q3	Q4, Q5, Q6
Laboratório de Computação 3	Q7	Q8, Q9

1- (0,5 pontos) O que significa dizer que uma função  $g(n)$  é  $O(f(n))$ ?

2- (1 ponto) Indique se as alternativas abaixo são verdadeiras ou falsa justificando suas respostas

- $3n^3 + 2n^2 + n + 1 = O(n^3)$
- $7n^2 = O(n)$
- $2^{n+2} = O(2^n)$
- $2^{2n} = O(2^n)$
- $5n^2 + 7n = \Theta(n^2)$
- $6n^3 + 5n^2 \neq \Theta(n^2)$
- $9n^3 + 3n = \Omega(n)$

3) (1 ponto) Determine a complexidade dos seguintes algoritmos:

a)

```
ALG1 ()  
for i ← 1 to n do  
    for j ← 1 to n do  
        k ← k + 1
```

b)

```

ALG2 ()
i ← 1
soma ← 0
while (i ≤ n) do
    j ← 1
    while (i ≤ n) do
        soma ← soma + 1
        j ← j + 1
    i ← i + 1

```

c)

```

ALG3 ()
for i ← 1 to 2 do
    for j ← i to n do
        for k ← i to j do
            temp ← temp + i + j + k

```

d)

```

INSERTION-SORT(A)
for j ← 2 to comprimento[A] do
    chave ← A[j]
    i ← j - 1
    A[0] ← chave //sentinela
    while A[i] > chave do
        A[i+1] ← A[i]
        i ← i-1
    A[i+1] ← chave

```

e)

```
BUBBLE-SORT (A)
for i ← 1 to comprimento[A] do
    for j ← comprimento[A] downto i+1 do
        if A[j] < A[j-1] then
            A[j] ↔ A[j-1]
```

f)

```
SELECTION-SORT (A)
for i ← 1 to comprimento[A]-1 do
    Min ← i
    for j ← i+1 to comprimento[A] do
        if A[j] < A[Min] then
            Min ← j
    A[Min] ↔ A[i]
```

4) (1 ponto) Resolva os seguintes exercícios do livro do Cormen:

1.1-2) Other than speed, what other measures of efficiency might one use in a real-world setting?

1.1-3) Select a data structure that you have seen previously, and discuss its strengths and limitations.

1.1-4) How are the shortest-path and traveling-salesman problems given above similar? How are they different?

1.1-5) Come up with a real-world problem in which only the best solution will do. Then come up with one in which a solution that is “approximately” the best is good enough

1.2-2) Suppose we are comparing implementations of insertion sort and merge sort on the same machine. For inputs of size  $n$ , insertion sort runs in  $8n^2$  steps, while merge

sort runs in  $64n \lg n$  steps. For which values of  $n$  does insertion sort beat merge sort?

1.2-3) What is the smallest value of  $n$  such that an algorithm whose running time is  $100n^2$  runs faster than an algorithm whose running time is  $2n$  on the same machine?

1.2-3) Express the function  $n^3 = 1000 - 100n^2 - 100n + 3$  in terms of  $\Omega$  Omega-notation.

### 3-2 Relative asymptotic growths

Indicate, for each pair of expressions  $(A, B)$  in the table below, whether  $A$  is  $O$ ,  $o$ ,  $\Omega$ ,  $\omega$ , or  $\Theta$  of  $B$ . Assume that  $k \geq 1$ ,  $\epsilon > 0$ , and  $c > 1$  are constants. Your answer should be in the form of the table with “yes” or “no” written in each box.

	$A$	$B$	$O$	$o$	$\Omega$	$\omega$	$\Theta$
a.	$\lg^k n$	$n^\epsilon$					
b.	$n^k$	$c^n$					
c.	$\sqrt{n}$	$n^{\sin n}$					
d.	$2^n$	$2^{n/2}$					
e.	$n^{\lg c}$	$c^{\lg n}$					
f.	$\lg(n!)$	$\lg(n^n)$					

5) (0,75 ponto) Determine a complexidade dos seguintes procedimentos recursivos a partir de suas equações de recorrência. Utilize o método da expansão telescópica

a)  $T(n) = 3.T(n/3) + n$

b)  $T(n) = 2.T(n/2) + n^2$

c) Série de Fibonacci

d) Função *misterio*

```

inteiro misterio (conjunto A com n elementos)
    se o número de elementos de A for igual a um
    então retorne o elemento de A
    senão se o número de elementos de A for zero
    então retorne menos infinito
    senão
    - Divida o conjunto A em três conjuntos (B, C, D) de tamanhos iguais
    - b = misterio (B)
    - c = misterio (C)
    - d = misterio (D)
    - maior = max {b,c,d}
    retorne maior
    
```

6) (0,75 ponto) utilizando o teorema mestre, determine a complexidade dos seguintes procedimentos recursivos.

a)  $T(n) = T(n/2) + 2^n$

b)  $T(n) = 4T(n/2) + n^2$

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

### **Questões Práticas / Implementação**

7) (3 ponto) Análise experimental de Algoritmos

Escolha dois algoritmos de ordenação interna e compare-os quanto ao tempo médio de execução no caso médio. A média deverá ser calculada em 10 amostras aleatórias variando 10 tamanhos de  $n$ . Ou seja, você deverá realizar 10x10 experimentos para cada algoritmo. A comparação deverá ser realizada por meio de gráficos e discussões. Nos gráficos aponte a curva de tempo médio para cada valor de  $n$  e seu desvio padrão (erro).

8) (1 pontos) Seja  $L$  uma sequência de número inteiros. Projete e implemente em Java um algoritmo que encontra uma subsequência de soma máxima com custo  $O(n)$ .

9) (1 ponto) Dado duas sequências ordenadas com  $m$  e  $n$  elementos, respectivamente. Projete e implemente em Java um algoritmo com custo computacional  $O(\log(\max(m, n)))$  para encontrar o  $k$ -ésimo menor elemento quando juntarmos as duas sequências.