



Curso de Bacharelado em Ciência da Computação

Estruturas de Dados

Prof. Otávio Alcântara

Nome: \_\_\_\_\_

Data: \_\_\_\_\_

### Lista de exercícios

1. Para cada uma das operações abaixo, forneça o tempo de execução usando a notação O grande.
  - a. Busca de um dado em array desordenado com n posições.
  - b. Busca binária de um dado em um array ordenado com n posições.
  - c. Busca de um dado em uma matriz quadrada com n linhas e n colunas.
  - d. Cálculo de  $c^n$  usando apenas multiplicações.
  - e. Cálculo de  $c^n$  usando apenas adições.
2. Coloque as funções abaixo em ordem crescente de complexidade assintótica.
  - a.  $10n(\log n)^2$ ,  $n^3$ ,  $100(\log n)^3$ ,  $2^n$
  - b.  $10^4 n^2$ ,  $10^{-2} \times 2^n$ ,  $10n^3$ ,  $300n \log n$
  - c.  $100n^3$ ,  $10^3 n^2 \cdot 2n$ ,  $10^2 n \log n$
  - d.  $4n \log n + 2n$ ,  $2^{10}$ ,  $n^3$ ,  $n \log n$ ,  $n^2 + 10n$ ,  $4n$
3. Marque V ou F, considerando que:
  - a.  $f(n) = 5000n^2 + 800n \log n$
  - b.  $g(n) = 90n^3 + 900n^2 + 9000n + 90000$
  - c.  $h(n) = 20 \log n + 100n + n^2$
  - i. (    )  $f(n) = O(h(n))$
  - ii. (    )  $f(n) = \Omega(g(n))$
  - iii. (    )  $f(n) = \theta(g(n))$
  - iv. (    )  $g(n) = O(f(n))$
  - v. (    )  $g(n) = \Omega(h(n))$
  - vi. (    )  $g(n) = \theta(f(n))$
4. O número de operações executada pelas algoritmos A e B é  $40n^2$  e  $2n^3$ , respectivamente. Determine  $n_0$  tal que A é melhor do que B para  $n \geq n_0$ .
5. Descreva um algoritmo eficiente para encontrar os dez maiores elementos em uma sequência de inteiros de tamanho n. Qual é a complexidade do seu algoritmo?

6. Determine a complexidade dos trechos de código abaixo:

```
def example1(s):  
    n = len(s)  
    total = 0  
    for j in range(n):  
        total += s[j]  
    return total
```

a.

```
def example2(s):  
    n = len(s)  
    total = 0  
    for j in range(n):  
        for k in range(1+j):  
            total += s[k]  
    return total
```

b.

```
def example3(s):  
    n = len(s)  
    total = 0  
  
    for j in range(n):  
        k = 1  
        while k < n//2:  
            total += s[k]  
            k += j  
    return total
```

c.

1. Um rei CRUEL possui  $n$  garrafas de vinho, um espião acaba de envenenar uma delas. O veneno é mortal, mata a pessoa que beber uma gota em um mês. Projete um algoritmo para determinar qual garrafa foi envenenada em um único mês, enquanto usa apenas  $O(\log n)$  provadores de vinho.