

# Lista de Exercícios de Análise de Algoritmos

Ásafe Duarte  
ESTRUTURAS-2024-2

January 4, 2025

**Questão 1.** Para cada uma das operações abaixo, forneça o tempo de execução usando a notação O grande.

- Busca de um dado em array desordenado com  $n$  posições:  $O(n)$
- Busca binária de um dado em um array ordenado com  $n$  posições:  $O(\log n)$
- Busca de um dado em uma matriz quadrada com  $n$  linhas e  $n$  colunas:  $O(n^2)$
- Cálculo de  $cn$  usando apenas multiplicações:  $O(\log c)$
- Cálculo de  $cn$  usando apenas adições:  $O(c)$

**Solução.** As soluções foram apresentadas diretamente ao lado das operações.

**Questão 2.** Coloque as funções abaixo em ordem crescente de complexidade assintótica:

- a.  $\log n, n, n^3, 2^n$
- b.  $n \log n, n^2, n^3, 2^n$
- c.  $n^3, n^2, n \log n$
- d.  $n^3, n, 1$

**Solução.** a)  $\log n, n, n^3, 2^n$   
b)  $n \log n, n^2, n^3, 2^n$   
c)  $n \log n, n^2, n^3$   
d)  $1, n, n^3$

**Questão 3.** Marque V ou F, considerando que:

- i.  $f(n) = O(h(n)) : V$
- ii.  $f(n) = \Omega(g(n)) : F$
- iii.  $f(n) = \theta(g(n)) : F$

iv.  $g(n) = O(f(n)) : F$

v.  $g(n) = \Omega(h(n)) : V$

vi.  $g(n) = \theta(f(n)) : F$

**Solução.** As soluções estão ao lado das expressões.

**Questão 4.** O número de operações executadas pelos algoritmos A e B é  $40n^2$  e  $2n^3$ , respectivamente. Determine  $n_0$  tal que A é melhor do que B para  $n \geq n_0$ .

**Solução.** Resolvemos a equação:

$$40n^2 = 2n^3$$

Dividindo ambos os lados por  $n^2$ :

$$40 = 2n$$

Dividindo ambos os lados por 2:

$$20 = n$$

Logo,  $n_0 = 21$ .

**Questão 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?

**Solução.** Github

**Questão 6.** Determine a complexidade dos trechos de código abaixo:

**Solução.** a)  $n$

b)  $n^2$

c)  $n \log n$

**Questão 7.** 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.

**Solução.** podemos converter a quantidade de garrafas para binário:

cada provador vai provar das seguintes garrafas:

provador 1: em que o bit 1 (da direita para a esquerda) (menos significativo) está em estado alto (1)

provador 2: em que o bit 2 está em estado alto

Assim, após um mês, podemos juntar os bits baseados nos provadores que morreram, descobrindo qual garrafa era a envenenada.