

---

## Lista de Exercícios: Noções de Complexidade

---

① Questão 6 página 24 [EDD]

Considere dois algoritmos  $A$  e  $B$  com funções de complexidade de tempo  $\alpha(n) = n^2 - n + 500$  e  $\beta(n) = 47n + 47$ , respectivamente. Para quais valores de  $n$  o algoritmo  $A$  leva menos tempo para executar do que  $B$ ?

② Questão 8 página 25 [EDD]

Calcule a complexidade, no pior caso, do seguinte fragmento de código:

```
1 int i,j,k;
2
3 for (i=0; i<N; i++){
4     for (j=0; j<N; j++){
5         R[i][j] = 0;
6         for(k=0; k<N; k++)
7             R[i][j] += A[i][k] * B[k][j]
8     }
9 }
```

③ Questão 9 página 25 [EDD]

Calcule a complexidade, no pior caso, do seguinte fragmento de código:

```
1 int i,j,k,s;
2
3 for (i=0; i<N-1; i++){
4     for (j=i+1; j<N; j++){
5         R[i][j] = 0;
6         for(k=1; k<j; k++)
7             s = 1;
8     }
9 }
```

④ Questão 10 página 25 [EDD]

Calcule a complexidade, no pior caso, do seguinte fragmento de código:

```
1 int i,j,s;
2 s = 0;
3 for (i=1; i<N-1; i++){
4     for (j=1; j<2*N; j++){
5         s = s + 1;
6     }
7 }
```

⑤ Considere o fragmento de código abaixo. Suponha que as linhas 3 e 4 consomem  $O(1)$  unidades de tempo (ou seja, consomem tempo constante, independente de  $n$ ). Suponha que cada chamada  $Aux(n)$  consome  $O(n^3)$  unidades de tempo. Quanto tempo consome o algoritmo todo?

```

1 a := 0
2 para k := 1 até n^2
3     a := a + 1
4     se k <= n
5         b := Aux(k)

```

⑥ Questão 3 página 24 [EDD]

O que significa dizer que a função  $g(n) \in O(f(n))$  (ou  $g(n)$  é  $O(f(n))$ )?

⑦ Questão 1.14 [EDA] Escrever as seguintes funções em notação  $O$ :

$$n^3 - 1; n^2 + 2 \log n; 3n^n + 5 \cdot 2^n; (n-1)^n + n^{n-1}; 302.$$

⑧ Existem números  $c$  e  $n_0$ , positivos, tais que  $n^3 \leq c \cdot n^2$  para todo  $n$  maior que  $n_0$ ?

⑨ (Transitividade) Suponha que  $f \in O(g)$  e  $g \in O(h)$ . Prove que  $f \in O(h)$ .

⑩ Questão 1.20 [EDA]

A *sequência de Fibonacci* é uma sequência de elementos  $f_1, \dots, f_n$ , definida do seguinte modo:

$$\begin{aligned} f_1 &= 0, \\ f_2 &= 1, \\ f_j &= f_{j-1} + f_{j-2}, \quad j > 2. \end{aligned}$$

Elaborar um algoritmo, não recursivo, para determinar o elemento  $f_n$  da sequência, cuja complexidade seja linear em  $n$ .

⑪ Questão 1.22 [EDA]

Determine o número de chamadas recursivas e a complexidade do seguinte algoritmo, para determinar o elemento  $f_n$  da sequência de Fibonacci.

```

1 funcao f(n)
2 f := se n=1 entao 0
3     senao se n=2 entao 1
4         senao f(n-1) + f(n-2)

```

Algorithm 0.1: Fibonacci

⑫ (**Prática**) Escreva as duas versões anteriores (iterativa e recursiva) para o cálculo do  $n$ -ésimo número de Fibonacci. Você pode considerar a implementação de duas funções distintas, uma iterativa e outra recursiva. Para calcular o tempo de máquina, você pode usar a biblioteca TIME.H e determinar o tempo em segundos que leva a chamada de cada uma das funções. O que acontece com o tempo de execução para valores crescentes do  $n$ ?

⑬ Questão 1.23 [EDA]

Considere a seguinte sequência de elementos  $g_1, \dots, g_n$  para um dado valor de  $k$ .

$$\begin{aligned} g_j &= j - 1, \quad 1 \leq j \leq k; \\ g_j &= g_{j-1} + g_{j-2}, \quad j > k; \end{aligned}$$

14 Resolva a seguinte equação de recorrência:

$$T(n) = \begin{cases} 2T\left(\frac{n}{2}\right) + O(n) & n > 1 \\ O(1) & n = 1 \end{cases}$$

[EDA] J.L. SZWARCFITER and L. MARKEZON, “Estruturas de Dados e seus Algoritmos,” *LTC*, 2010, 3<sup>a</sup> ed.

[EDD] A. BACKES, “Estruturas de Dados Descomplicada - Em Linguagem C,” *Elsevier*, 2016, 1<sup>a</sup> ed.