

**Lista de Exercícios - Unidade I**

**Complexidade:**

1) Da menor para a maior, determine a ordem das complexidades a seguir:  $O(n^2)$ ,  $O(4n)$ ,  $O(2^n)$ ,  $O((n^2) \lg n)$ ,  $O(1)$ ,  $O(n \lg n)$ ,  $O(n^4)$ ,  $O(n!)$ ,  $O(\lg n)$ ,  $O(n)$ ,  $O(n^*m)$ .

2) Qual a complexidade de  $T_1(n) = (5 \cdot n) \lg n + \lg n$ ;  $T_2(n) = 3^n + n^3 + 30$ ; e  $T_3(n, k) = k + n$ , onde  $k \leq n$ ? Da menor, para a maior, determine a ordem das complexidades.

3) Suponha que você calculou a complexidade de dois algoritmos diferentes, o primeiro algoritmo tem a complexidade descrita por  $f(n)$  e o outro por  $g(n,k)$ . Sendo  $f(n)=3n+n^2$  e  $g(n,k)=(n/(k+1))(3n + n^2)$  com  $k \geq n$  para todo  $k$  e  $n$  naturais. Os dois algoritmos têm complexidades iguais ou diferentes? Justifique sua resposta.

4) Descreva o teorema mestre e aplique-o para mostrar a solução para a recorrência  $T(n)=T(n/2) + \theta(1)$ .

5) Encontre o limite superior para as seguintes funções:

- a)  $f(n) = 5n^3 + 8$
- b)  $f(n) = n^3 + 1$
- c)  $f(n) = n^4 + 50n^3 + 50$
- d)  $f(n) = 2n^2/1000 - 3n^2$
- e)  $f(n) = n * m$
- f)  $f(n) = 230$

6) Use o teorema mestre para derivar um limite assintótico  $\Theta$  para as seguintes recorrências:

- a)  $T(n) = 2T(n/2) + n - 1$
- b)  $T(n) = 5T(n/2) + n$
- c)  $T(n) = 9T(n/3) + n^2$
- d)  $T(n) = 16T(n/4) + n^3$

7) Apresente a ordem de crescimento da execução dos algoritmos a seguir. Descreva a solução.

```

a)
int sum = 0
for(int i = 1; i < n; i*=3)
    for(j = 0; j < i; j++)
        sum++;

b)
int sum = 0
for(int i = 1; i < n; i*=3)
    for(j = 0; j < n^2; j++)
        sum++;

```

```

C) int count = 0;
   for (int i = 1; i < N; i *= 2)
       for (int j = 0; j < N; j++)
           count++;

D) int count = 0;
   for (int i = 1; i < N; i *= 2)
       for (int j = 0; j < N; j++)
           for (int k = 0; k < N; k++)
               count++;

```

### Ordenação:

- 8) Ilustre a operação do Selection sort no vetor E D I T O O E A S Y..
- 9) Descreva o passo a passo da execução do algoritmo de QuickSort no Vetor  $V = [9, 4, 7, 2, 10, 23, -4]$ , o elemento mais à direita como pivô.
- 10) Ilustre a operação do Insertion sort no vetor Q U E S T A O L E G A L.
- 11) Aplique o algoritmo do ShellSort para ordenar o seguinte vetor:  $V = [24, 7, 32, 13, 76, 45, 19, 36, 20, 65, 5]$ . Destaque o estado de  $V$  após todas as trocas, iniciando com o valor de  $h=4$ , depois com  $h=2$  e por fim com  $h=1$ .
- 12) Dado o vetor  $V = [37, 92, 45, 20, 32, 11, 53]$ , ordene em ordem crescente utilizando o algoritmo de ordenação MergeSort apresentando o passo a passo, a quantidade de comparações realizadas para ordenar completamente o vetor.
- 13) Qual o método mais rápido para ordenar um vetor no qual todos os elementos são idênticos, Insertion sort ou QuickSort?
- 14) Um vetor ordenado em ordem crescente é um min-heap? Demonstre sua solução
- 15) Você precisa armazenar  $n$  números em um max-heap ou em um vetor ordenado. Para cada caso abaixo, explique qual das estruturas de dados é a melhor, ou se não faz diferença.
  - a) É preciso achar o maior elemento rapidamente.
  - b) É preciso deletar um elemento rapidamente.
  - c) É preciso construir a estrutura rapidamente.
  - d) É preciso achar o menor elemento rapidamente.

### Referências:

SEDGEWICK, Robert; WAYNE, Kevin. Algorithms (4th edn). **Google Scholar**  
**Google Scholar Digital Library Digital Library**, 2011.  
 SKIENA, Steven S. The Algorithm Design Manual. 2008.  
 LOUDON, Kyle. **Mastering algorithms with C**. " O'Reilly Media, Inc.", 1999.  
 CORMEN, Thomas; LEISERSON, Charles; RIVEST, Ronald. **Algoritmos**.  
 Elsevier Brasil, 2017.