

## SSC0503 - Introdução à Ciência de Computação II

### Respostas da 1ª Lista

**Professor:** Claudio Fabiano Motta Toledo (claudio@icmc.usp.br)

**Estagiário PAE:** Jesimar da Silva Arantes (jesimar.arantes@usp.br)

---

**Resposta da pergunta 2:**  $\Theta(n^3)$ .

**Resposta da pergunta 8:**

1.  $7 \in O(n)$ . Sim.
2.  $n \in O(1)$ . Não.
3.  $n + 7 \in O(n)$ . Sim.
4.  $n + 7 \in O(1)$ . Não.
5.  $n^2 + 2 \in O(n)$ . Não.
6.  $n + 2 \in O(n^2)$ . Sim.
7.  $3n^3 + n \in O(n^3)$ . Sim.
8.  $2n^4 \in O(n^4)$ . Sim.
9.  $n^4 \in O(2n^4)$ . Sim.
10.  $3n^4 + 2n^3 \in O(2n^4)$ . Não.
11.  $2n^4 \in O(3n^4 + 2n^3)$ . Sim.
12.  $\log n \in O(1)$ . Não.
13.  $\log n + 1 \in O(\log n)$ . Sim.
14.  $\log n + 1 \in O(n)$ . Sim.
15.  $\log n + 1 \in O(n^2)$ . Sim.
16.  $\log n + 1 \in O(n^3)$ . Sim.
17.  $n \cdot \log n \in O(1)$ . Não.
18.  $\log n + 1 \in O(\log n)$ . Sim.
19.  $\log n + 1 \in O(n)$ . Sim.
20.  $\log n + 1 \in O(n^2)$ . Sim.
21.  $\log n + 1 \in O(n^3)$ . Sim.
22.  $n \cdot \log n \in O(1)$ . Não.

23.  $n \cdot \log n + 1 \in O(\log n)$ . Não.
24.  $n \cdot \log n + 1 \in O(n)$ . Não.
25.  $n \cdot \log n + 1 \in O(n^2)$ . Sim.
26.  $n \cdot \log n + 1 \in O(n^3)$ . Sim.
27.  $2\log n \in O(n \cdot \log n)$ . Sim.
28.  $3n \cdot \log n \in O(\log n)$ . Não.
29.  $2n + n \in O(2^3)$ . Não.
30.  $n^2 \in O(2^n)$ . Sim.
31.  $100n^4 \in O(2^n)$ . Sim.
32.  $100n^4 \in O(n^n)$ . Sim.
33.  $2^n \in O(100n^4)$ . Não.
34.  $2^n \in O(n^n)$ . Sim.
35.  $n^n \in O(2^n)$ . Não.
36.  $n^{100} \in O(n^n)$ . Sim.
37.  $n(n+1)/2 \in O(n^3)$ . Sim.
38.  $n(n+1)/2 \in O(n^2)$ . Sim.
39.  $n(n+1)/2 \in \Theta(n^3)$ . Não.
40.  $n(n+1)/2 \in \Omega(n)$ . Sim.

**Resposta da pergunta 12:**

1.  $(n^2 + 1)^{10} = \Theta(n^{20})$
2.  $\sqrt{10n^2 + 7n + 3} = \Theta(n)$
3.  $2n \cdot \log(n+2)^2 + (n+2)^2 \cdot \log\left(\frac{n}{2}\right) = \Theta(n^2 \cdot \log n)$
4.  $2^{n+1} + 3^{n-1} = \Theta(3^n)$
5.  $\log(n+37) = \Theta(\log n)$

**Resposta da pergunta 18:**

- (i) Soma()
- (ii)  $s = 0;$
- (iii) **para**  $i = 1$  **até**  $n$  **faça**
- (iv)  $s = s + i;$
- (v) **fim-para**
- (vi) **retorne**  $s;$
- (vii) **fim.**

- (i) 55
- (ii) a soma.
- (iii)  $n$  vezes.
- (iv)  $O(n)$ .
- (v) Sim. Existe uma formula fechada que faz esse calculo  $s = n \cdot (n + 1)/2$ . Dessa forma, basta implementar um algoritmo direto baseado nessa equação.

**Resposta da pergunta 19:**

- (i) MaxMin(vetor  $v$ )
- (ii)  $\max = v[1];$
- (iii)  $\min = v[1];$
- (iv) **para**  $i = 2$  **até**  $n$  **faça**
- (v) **se**  $v[i] > \max$  **então**
- (vi)  $\max = v[i];$
- (vii) **fim-se**
- (viii) **se**  $v[i] < \min$  **então**
- (ix)  $\min = v[i];$
- (x) **fim-se**
- (xi) **fim-para**
- (xii) **fim.**

- (i)  $\min = 0$ .  $\max = 9$ .
- (ii) a comparação.
- (iii)  $2 \cdot (n - 1)$  vezes.
- (iv)  $O(n)$ .
- (v) Não existe algoritmo mais eficiente.

**Resposta da pergunta 20:**

- Algoritmo para converter um número de Celsius para Kelvin. Complexidade:  $O(1)$ . Entrada: um número. Saída: um número.
- Algoritmo para converter um número de km/h para m/s. Complexidade:  $O(1)$ . Entrada: um número. Saída: um número.
- Algoritmo para converter um número de graus para radianos. Complexidade:  $O(1)$ . Entrada: um número. Saída: um número.
- Algoritmo para calcular o IMC de uma pessoa. Complexidade:  $O(1)$ . Entrada: dois números. Saída: um número.
- Algoritmo para fazer a troca de dois números. Complexidade:  $O(1)$ . Entrada: dois números. Saída: dois números.
- Algoritmo para calcular a média aritmética de dois números. Complexidade:  $O(2) = O(1)$ . Entrada: dois números. Saída: um número.
- Algoritmo para calcular a média aritmética de 10 números. Complexidade:  $O(10) = O(1)$ . Entrada: 10 números. Saída: um número.
- Algoritmo para calcular a média aritmética de  $n$  números. Complexidade:  $O(n)$ . Entrada:  $n$  número. Saída: um número.
- Algoritmo para calcular a raiz de uma equação do segundo grau. Complexidade:  $O(1)$ . Entrada: três números. Saída: dois números.
- Algoritmo para calcular se um ano é bissexto. Complexidade:  $O(1)$ . Entrada: um número. Saída: um booleano.
- Algoritmo para calcular se um número é primo. Complexidade: Algumas implementações tem complexidade  $O(n)$ , outras  $O(n/2) = O(n)$  e outras  $O(\sqrt{n})$ . Entrada: um número. Saída: um booleano.
- Algoritmo para somar dois vetores de tamanho  $n$ . Complexidade:  $O(n)$ . Entrada:  $2 \cdot n$  números. Saída:  $n$  números.
- Algoritmo para subtrair dois vetores de tamanho  $n$ . Complexidade:  $O(n)$ . Entrada:  $2 \cdot n$  números. Saída:  $n$  números.
- Algoritmo para calcular o fatorial de  $n$ . Complexidade:  $O(n)$ . Entrada: um número. Saída: um número.
- Algoritmo para calcular o fibonacci de  $n$ . Complexidade: Abordagem iterativa  $O(n)$ , abordagem recursiva  $O(\phi^n)$ . Entrada: um número. Saída: um número.
- Algoritmo para verificar se uma palavra é palíndromo ou não. Complexidade:  $O(n/2) = O(n)$ . Entrada: uma palavras de tamanho  $n$ . Saída: um booleano.

- Algoritmo para verificar se um vetor de tamanho  $n$  está ordenado ou não. Complexidade:  $O(n)$ . Entrada:  $n$  números. Saída: um booleano.
- Algoritmo para criptografar uma frase pela cifra de César. Complexidade:  $O(n)$ . Entrada: uma frase de tamanho  $n$ . Saída: uma frase de tamanho  $n$ .
- Algoritmo para soma de duas matrizes. Complexidade:  $O(n^2)$ . Entrada: duas matrizes  $n \times n$ . Saída: uma matriz  $n \times n$ .
- Algoritmo para subtração de duas matrizes. Complexidade:  $O(n^2)$ . Entrada: duas matrizes  $n \times n$ . Saída: uma matriz  $n \times n$ .
- Algoritmo para multiplicação de duas matrizes. Complexidade:  $O(n^3)$ . Entrada: duas matrizes  $n \times n$ . Saída: uma matriz  $n \times n$ .
- Algoritmo para multiplicação de uma matriz por um escalar. Complexidade:  $O(n^2)$ . Entrada: uma matriz  $n \times n$ , um escalar. Saída: uma matriz  $n \times n$ .
- Algoritmo para transposta de uma matriz. Complexidade:  $O(n^2)$ . Entrada: uma matriz  $n \times n$ . Saída: uma matriz  $n \times n$ .