

## SSC0503 - Introdução à Ciência de Computação II

### 1ª Lista

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

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

---

1. O que significa dizer que uma função  $f(n)$  é  $O(g(n))$ ?
2. Expresse a função  $2n^3/100 - 5n^2 + 100n + 3$  em termos da notação  $\Theta$ .
3. Por muitas vezes damos atenção apenas ao pior caso dos algoritmos. Explique o porquê.
4. É verdade que  $2^{n+1} = O(2^n)$ ? É verdade que  $2^{2n} = O(2^n)$ ?
5. Dois algoritmos A e B possuem complexidade  $n^5$  e  $2^n$  respectivamente. Qual algoritmo você considera "melhor"? Qual algoritmo é melhor assintoticamente? Você utilizaria o algoritmo B ao invés do A, em algum caso? Exemplifique.
6. Suponha que estamos comparando implementações da ordenação por inserção (*insertion sort*) e *merge sort* na mesma máquina. Para entradas de tamanho  $n$ , o *insertion sort* executa em  $8n^2$  passos, enquanto o *merge sort* é executado em  $64n \cdot \lg n$  passos. Para que valores de  $n$  o *insertion sort* é melhor que o método *merge sort*?
7. Qual é o menor valor de  $n$  tal que um algoritmo que executa num tempo de  $100n^2$  roda mais rápido do que um algoritmo cujo tempo de execução é  $2^n$  na mesma máquina?
8. Mostre que:
  - $2n + 10$  é  $O(n)$
  - $\frac{1}{2}n(n + 1)$  é  $O(n^2)$
  - $n + \sqrt{n}$  é  $O(n)$
  - $n/1000$  não é  $O(1)$
  - $\frac{1}{2}n^2$  não é  $O(n)$
  - $\frac{1}{2}n^2 - 3n$  é  $O(n^2)$
9. Seja a seguinte definição: "Dadas duas funções,  $f(n)$  e  $g(n)$ , diz-se que  $f(n)$  é da ordem de  $g(n)$  ou que  $f(n)$  é  $O(g(n))$ , se existirem inteiros positivos  $a$  e  $b$  tais que  $f(n) \leq a \cdot g(n)$  para todo  $n \geq b$ ." Verifique se as seguintes proposições estão corretas:
  1.  $7 \in O(n)$
  2.  $n \in O(1)$
  3.  $n + 7 \in O(n)$
  4.  $n + 7 \in O(1)$
  5.  $n^2 + 2 \in O(n)$
  6.  $n + 2 \in O(n^2)$

7.  $3n^3 + n \in O(n^3)$
8.  $2n^4 \in O(n^4)$
9.  $n^4 \in O(2n^4)$
10.  $3n^4 + 2n^3 \in O(2n^4)$
11.  $2n^4 \in O(3n^4 + 2n^3)$
12.  $\log n \in O(1)$
13.  $\log n + 1 \in O(\log n)$
14.  $\log n + 1 \in O(n)$
15.  $\log n + 1 \in O(n^2)$
16.  $\log n + 1 \in O(n^3)$
17.  $n \cdot \log n \in O(1)$
18.  $\log n + 1 \in O(\log n)$
19.  $\log n + 1 \in O(n)$
20.  $\log n + 1 \in O(n^2)$
21.  $\log n + 1 \in O(n^3)$
22.  $n \cdot \log n \in O(1)$
23.  $n \cdot \log n + 1 \in O(\log n)$
24.  $n \cdot \log n + 1 \in O(n)$
25.  $n \cdot \log n + 1 \in O(n^2)$
26.  $n \cdot \log n + 1 \in O(n^3)$
27.  $2\log n \in O(n \cdot \log n)$
28.  $3n \cdot \log n \in O(\log n)$
29.  $2n + n \in O(2^3)$
30.  $n^2 \in O(2^n)$
31.  $100n^4 \in O(2^n)$
32.  $100n^4 \in O(n^n)$
33.  $2^n \in O(100n^4)$
34.  $2^n \in O(n^n)$
35.  $n^n \in O(2^n)$
36.  $n^{100} \in O(n^n)$
37.  $n(n+1)/2 \in O(n^3)$
38.  $n(n+1)/2 \in O(n^2)$
39.  $n(n+1)/2 \in \Theta(n^3)$
40.  $n(n+1)/2 \in \Omega(n)$

10. Construa e analise gráficos em uma planilha eletrônica (excel ou calc) das seguintes funções  $1$ ,  $\log(\log(n))$ ,  $\log(n)$ ,  $\sqrt{(n)}$ ,  $n$ ,  $n \cdot \log(n)$ ,  $n^2$ ,  $n^3$ ,  $2^n$ ,  $3^n$ ,  $n!$ ,  $n^n$ . Para a construção dessa planilha, coloque cada função em uma coluna e nas linhas coloque a entrada  $n$  variando de 1 a 100. Faça a plotagem de vários gráficos diferentes variando os valores limite do conjunto imagem (eixo y).
11. Ordene as seguintes funções por suas taxas de crescimento:  $n$ ,  $\sqrt{n}$ ,  $n^1$ ,  $5$ ,  $n^2$ ,  $n \cdot \log n$ ,  $n \cdot \log(\log n)$ ,  $n \cdot (\log n)^2$ ,  $n \cdot \log n^2$ ,  $2/n$ ,  $2^n$ ,  $2^n/2$ ,  $37$ ,  $n^2 \cdot \log n$ ,  $n^3$ ,  $(n-2)!$ ,  $5 \cdot \log(n+100)^{10}$ ,  $0,001n^4 + 3n^3 + 1$ ,  $\sqrt[3]{n}$ ,  $3^n$ .
12. Compare as duas funções  $n^2$  e  $\frac{2^n}{4}$  para vários valores de  $n$ . Determine quando a segunda se torna maior que a primeira.
13. Para cada uma das seguintes funções, determine a classe  $\Theta(g(n))$  a qual a função pertence. (Use  $g(n)$  mais simples possível).
1.  $(n^2 + 1)^{10}$
  2.  $\sqrt{10n^2 + 7n + 3}$
  3.  $2n \cdot \log(n + 2)^2 + (n + 2)^2 \cdot \log(\frac{n}{2})$
  4.  $2^{n+1} + 3^{n-1}$
  5.  $\log(n + 37)$
14. Considere um computador com *clock* de 2GHz, que realiza cada operação relevante em 1 ciclo. Estime, apenas com esses dados, o tempo necessário para que ele execute um algoritmo que realiza  $(n^2 - n)/2$  operações relevantes, considerando que há 4M dados de entrada.
15. Idem, usando um algoritmo que realiza  $n^3$  operações relevantes.
16. Idem, usando um algoritmo que realiza  $2^n$  operações relevantes.
17. Idem, usando um algoritmo que realiza  $n^n$  operações relevantes.
18. Idem, para um computador com *clock* de 100MHz, ordenando a mesma sequência, usando um algoritmo que realiza  $4/3 \cdot n \cdot \log n$  operações relevantes. Analise os resultados.
19. Considere o seguinte algoritmo, sendo  $n \geq 0$  e inteiro:
- (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) Qual a resposta dada ao executar este algoritmo quando  $n = 10$ ?

- (ii) Qual é a operação básica?
- (iii) Quantas vezes essa operação é executada?
- (iv) Qual é a classe de eficiência desse algoritmo?
- (v) Existe um algoritmo melhor que responda ao mesmo problema? Descreva-o ou mostre que tal algoritmo não existe.

20. Considere o seguinte algoritmo, sendo  $n \geq 0$  e inteiro:

- (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) Qual o valor das variáveis  $\min$  e  $\max$  ao executar este algoritmo quando o vetor  $v = \{4, 6, 1, 5, 2, 9, 3, 0, 8, 7\}$ ?
- (ii) Qual é a operação básica?
- (iii) Quantas vezes essa operação é executada?
- (iv) Qual é a classe de eficiência desse algoritmo?
- (v) Existe um algoritmo melhor que responda ao mesmo problema? Descreva-o ou mostre que tal algoritmo não existe.

21. Baseado em seus conhecimentos sobre programação. Diga as ordens de complexidade dos algoritmos abaixo. Descubra em cada algoritmo quais são as entradas.

- Algoritmo para converter um número de Celsius para Kelvin.
- Algoritmo para converter um número de km/h para m/s.
- Algoritmo para converter um número de graus para radianos.
- Algoritmo para calcular o IMC de uma pessoa.
- Algoritmo para fazer a troca de dois números.
- Algoritmo para calcular a média aritmética de dois números.
- Algoritmo para calcular a média aritmética de 10 números.
- Algoritmo para calcular a média aritmética de  $n$  números.
- Algoritmo para calcular a raiz de uma equação do segundo grau.

- Algoritmo para calcular se um ano é bissexto.
- Algoritmo para calcular se um número é primo.
- Algoritmo para somar dois vetores de tamanho  $n$ .
- Algoritmo para subtrair dois vetores de tamanho  $n$ .
- Algoritmo para calcular o fatorial de  $n$ .
- Algoritmo para calcular o fibonacci de  $n$ .
- Algoritmo para verificar se uma palavra é palíndromo ou não.
- Algoritmo para verificar se um vetor de tamanho  $n$  está ordenado ou não.
- Algoritmo para criptografar uma frase pela cifra de César.
- Algoritmo para soma de duas matrizes.
- Algoritmo para subtração de duas matrizes.
- Algoritmo para multiplicação de duas matrizes.
- Algoritmo para multiplicação de uma matriz por um escalar.
- Algoritmo para transposta de uma matriz.

**Observação:** em alguns dos exemplos abaixo mais de uma estratégia de algoritmo poderá existir, sendo assim, tente pensar nos algoritmos vistos no semestre passado (ICCI) ou na forma mais simples possível desses algoritmos.