

IBMEC 2025/01

Prova 1 - Estrutura de Dados

Turma 1

5 de setembro de 2025

Instruções:

- A prova é individual e com consulta a uma folha A4 preparada pelos alunos, condicionada a entrega de questões.
- Tempo de duração da prova é de duas horas.

Dica para o sucesso na prova:

- Estruturar a fase de estudo do problema ANTES de começar a fazer o algoritmo ou programa; Identificar entradas; identificar a saída esperada; RESOLVER o problema; identificar como validar a solução; codificar; testar.

1 Complexidade de Algoritmos

1. (2.5 Pontos) Determine se as afirmações abaixo são verdadeiras (V) ou falsas (F) em relação à análise de complexidade de algoritmos:

- (a) A notação $O(f(n))$ representa um limite superior assintótico para o tempo de execução de um algoritmo. ()
- (b) O tempo de execução de um algoritmo com complexidade $O(n^3)$ é sempre maior do que o de um algoritmo com complexidade $O(n)$, independentemente do tamanho de n . ()
- (c) O pior caso de um algoritmo representa o cenário em que o tempo de execução é maximizado para o tamanho da entrada considerada. ()
- (d) O tempo de execução de um algoritmo com complexidade $O(1)$ é constante, independentemente do tamanho da entrada. ()
- (e) Para dois algoritmos A e B , se A tem complexidade $O(n \log n)$ e B tem complexidade $O(n^2)$, então A será mais rápido que B para todos os valores de n . ()

2. (1 Ponto) **Algoritmos e Estrutura de Dados**

Considere um vetor ordenado em ordem crescente com tamanho teoricamente infinito, ou seja, não é possível determinar diretamente seu tamanho ou seu último índice válido. Seu objetivo é desenvolver um algoritmo eficiente para encontrar a posição (índice) de um determinado valor nesse vetor. Assuma que o acesso a qualquer posição do vetor retorna o valor naquela posição caso a posição exista, ou retorna um valor especial indicando que a posição é inválida (por exemplo, infinito).

- a) Descreva claramente a lógica e o funcionamento do algoritmo proposto.
- b) Implemente, utilizando pseudocódigo, o algoritmo descrito na letra (a).
- c) Determine a complexidade computacional do algoritmo proposto em termos de tempo, justificando sua resposta.

Observação: O algoritmo proposto deve ser mais eficiente do que simplesmente percorrer sequencialmente o vetor elemento por elemento.

3. (1 Ponto) O que significa dizer que um algoritmo tem complexidade de tempo $O(N^3)$. Dica: Demonstre o crescimento da função e compare com outras funções de complexidade de tempo
4. (1 Ponto) Considere o problema de calcular o fatorial de um número n , definido como:

$$n! = n \times (n - 1) \times (n - 2) \times \cdots \times 1$$

Responda às perguntas abaixo com base na análise de complexidade de algoritmos para o cálculo do fatorial:

- (a) Qual é a complexidade temporal do algoritmo iterativo que calcula o fatorial de n ?
- (b) Existe uma diferença de complexidade entre o algoritmo iterativo e o recursivo para calcular o fatorial? Explique.
- (c) Implemente os dois algoritmos para o cálculo do fatorial (iterativo e recursivo).
5. (1 Pontos) Determine a complexidade de tempo do código abaixo e explique.

```
1 void func(int n) {
2     for(int i = 1; i < n; i *= 2) {
3         std::cout << i << std::endl;
4     }
5 }
```

6. (1.5 Pontos) Considere o seguinte código em C++ que realiza a multiplicação de duas matrizes A e B , ambas de dimensão $n \times n$:

```
1 void multiplica(int A[][100], int B[][100], int C[][100], int n) {
2     for (int i = 0; i < n; i++) {
3         for (int j = 0; j < n; j++) {
4             C[i][j] = 0;
5             for (int k = 0; k < n; k++) {
6                 C[i][j] += A[i][k] * B[k][j];
7             }
8         }
9     }
10 }
```

- (a) Qual é a complexidade temporal do algoritmo de multiplicação de matrizes implementado no código acima?
- (b) Se as matrizes fossem de dimensões $n \times m$ e $m \times p$, como a complexidade seria afetada?
- (c) Explique o que aconteceria com se n dobrasse. Como isso impactaria o número total de operações?

2 Ordenação

- (a) (1 Ponto) Implemente um algoritmo de ordenação qualquer (a sua escolha) e ordene o seguinte vetor: 3, -1, 2, 5, 12, 7, 5, 4, 9, -3;. Demonstre passo a passo e explique a complexidade do algoritmo escolhido.
- (b) (1 Ponto) Descreva em que circunstâncias é mais vantajoso escolher o Selection Sort em vez do Bubble Sort ou do Insertion Sort. Comente sobre os motivos dessa escolha.