

Disciplina: Estruturas de Dados 1

Aluno(a): _____

Prof. Tiago Pessoa Ferreira de Lima

1. Um algoritmo tem seu tempo de execução descrito pela função $T(n)=5n^2+100n+500$. Qual é a sua complexidade de tempo na notação Big O?

- a) $O(n)$
- b) $O(n^2)$
- c) $O(1)$
- d) $O(\log n)$

2. Qual das seguintes classes de complexidade representa o algoritmo mais eficiente para um conjunto de dados muito grande?

- a) $O(n)$
- b) $O(n^2)$
- c) $O(n \log n)$
- d) $O(\log n)$

3. Analise o código Python a seguir. Qual é a sua complexidade de tempo?

```
def verificar_duplicatas(lista):  
    n = len(lista)  
    for i in range(n):  
        for j in range(i + 1, n):  
            if lista[i] == lista[j]:  
                return True  
    return False
```

- a) $O(n)$
- b) $O(n^2)$
- c) $O(1)$
- d) $O(\log n)$

4. Se um algoritmo tem complexidade $O(n)$ e leva 2 segundos para processar 10.000 itens, qual seria o tempo de execução esperado para processar 20.000 itens?

- a) 2 segundos
- b) 4 segundos
- c) 8 segundos
- d) 16 segundos

5. Qual é a complexidade do seguinte trecho de código?

```
def soma_e_produto(lista):  
    soma = 0  
    for x in lista:  
        soma += x  
    produto = 1  
    for y in lista:  
        produto *= y  
    return soma, produto
```

- a) $O(n)$
- b) $O(n^2)$
- c) $O(1)$
- d) $O(\log_n)$

6. Por que a notação Big O ignora constantes?

- a) Porque as constantes são sempre pequenas e não impactam o resultado.
- b) Porque a notação Big O foca na taxa de crescimento assintótica, e as constantes não afetam como a complexidade escala com entradas grandes.
- c) Porque é uma regra matemática sem uma razão prática.
- d) Porque os computadores são tão rápidos que as constantes não fazem diferença.

7. Qual dos seguintes algoritmos possui uma complexidade de tempo $O(1)$?

- a) Encontrar um item em uma lista não ordenada de tamanho n .
- b) Acessar o terceiro elemento de um array.
- c) Imprimir todos os valores de uma lista de tamanho n .
- d) Porque os computadores são tão rápidos que as constantes não fazem diferença.
- e) Contar o número de elementos em uma lista encadeada.

8. Um algoritmo que, para uma entrada de tamanho n , executa uma operação que divide o conjunto de dados pela metade repetidamente até que reste apenas um elemento, tem qual complexidade?

- a) $O(n)$
- b) $O(n^2)$
- c) $O(1)$
- d) $O(\log_n)$

Rascunho