



FACULDADES INTEGRADAS LIBERTAS

Curso: Sistemas de Informação

Disciplina: Projeto e Análise de Algoritmos

Data de Entrega: 22/09/2025

Professor: Lucas Bruzzone

Aluno(a): _____

Lista de Exercícios: Análise de Complexidade (40 pontos)

1. Dado o algoritmo com complexidade $T(n) = 3n^2 + 5n \log n + 2 \log n + 7$, qual é o termo de maior ordem (mais complexo)?
 - a) 7
 - b) $2 \log n$
 - c) $5n \log n$
 - d) $3n^2$
2. Para a função $T(n) = 2^n + n^3 + 4n \log n + 100 + \log n$, qual é o termo de menor ordem (menos complexo)?
 - a) 100
 - b) $\log n$
 - c) $4n \log n$
 - d) n^3
3. Dado $T(n) = 2n^2 + n + 1$, **prove matematicamente** que $T(n) = O(n^2)$.

Demonstração requerida:

- Encontre constantes $c > 0$ e $n_0 \geq 0$ tais que $T(n) \leq c \cdot n^2$ para todo $n \geq n_0$
- Mostre os cálculos passo a passo
- Justifique a escolha das constantes

4. Para $T(n) = 7n \log n + 5n + 12$, a complexidade em Big O é:

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

5. Considerando $T(n) = 3n^3 + n^2 + n$, **prove matematicamente** que $T(n) = \Omega(n^3)$.

Demonstração requerida:

- Encontre constantes $c > 0$ e $n_0 \geq 0$ tais que $T(n) \geq c \cdot n^3$ para todo $n \geq n_0$
- Mostre os cálculos passo a passo
- Explique o significado da notação Ω

6. Para $T(n) = n^2 + 100n + 50$, a notação Omega (Ω) é:

- a) $\Omega(1)$
- b) $\Omega(n)$
- c) $\Omega(n^2)$
- d) $\Omega(n^3)$

7. Dado $T(n) = 2n^2 + n + 1$, **prove matematicamente** que $T(n) = \Theta(n^2)$.

Demonstração requerida:

- Prove que $T(n) = O(n^2)$ (limite superior)
- Prove que $T(n) = \Omega(n^2)$ (limite inferior)
- Encontre as constantes apropriadas para ambos os casos
- Explique por que isso implica $T(n) = \Theta(n^2)$

8. Analise o pseudocódigo da busca binária:

```
1 algoritmo BuscaBinaria(vetor, inicio, fim, x):
2     se inicio > fim:
3         retorna -1
4     meio = (inicio + fim) / 2
5     se vetor[meio] = x:
6         retorna meio
7     senao se vetor[meio] > x:
8         retorna BuscaBinaria(vetor, inicio, meio-1, x)
9     senao:
10        retorna BuscaBinaria(vetor, meio+1, fim, x)
```

Qual é a análise correta das complexidades deste algoritmo?

- a) $T(n) = O(\log n)$, $\Omega(\log n)$, $\Theta(\log n)$
- b) $T(n) = O(n)$, $\Omega(1)$, $\Theta(\log n)$
- c) $T(n) = O(\log n)$, $\Omega(1)$, $\Theta(\log n)$
- d) $T(n) = O(n)$, $\Omega(\log n)$, $\Theta(n)$

9. Considere o algoritmo Merge Sort:

```
1 algoritmo MergeSort(vetor, inicio, fim):
2     se inicio < fim:
3         meio = (inicio + fim) / 2
4         MergeSort(vetor, inicio, meio)
5         MergeSort(vetor, meio+1, fim)
6         Merge(vetor, inicio, meio, fim)
7
8 algoritmo Merge(vetor, inicio, meio, fim):
9     criar vetorTemp[fim - inicio + 1] // array auxiliar
10    // copia elementos para vetorTemp
11    // mescla ordenadamente de volta para vetor
```

Analise e determine:

- Explique claramente a diferença conceitual entre complexidade temporal e espacial
- A complexidade temporal $T(n)$ e sua justificativa
- A complexidade espacial $S(n)$ e sua justificativa