

# Relatório de Desempenho: Análise e Comparação dos Algoritmos de Ordenação

**ALUNO:** Allan Pedro da Silva Costa

**DATA:** 12/03/2025

**CURSO:** ciência da computação

## 1. Introdução

Este relatório apresenta a análise do desempenho de três algoritmos de ordenação: Bubble Sort, Quick Sort e Merge Sort. Os testes foram realizados em diferentes cenários para avaliar a eficiência de cada algoritmo.

## 2. Metodologia

Os algoritmos foram implementados em Python e testados com diversos tipos de dados:

- Listas já ordenadas.
- Listas quase ordenadas (apenas o último elemento fora de ordem).
- Listas inversamente ordenadas.
- Listas vazias.
- Listas com elementos repetidos.
- Listas aleatórias de tamanhos 1000, 10000 e 100000 elementos.

O tempo de execução foi medido usando a biblioteca time do Python.

## 3. Resultados

### 3.1 Testes Unitários

Todos os algoritmos passaram nos testes unitários, demonstrando que estão corretamente implementados. As listas foram ordenadas conforme esperado em todos os casos.

### 3.2 Medição do Tempo de Execução

Os tempos de execução para as listas aleatórias de diferentes tamanhos são mostrados abaixo:

TAMANHO DA LISTA	BUBBLE SORT (S)	QUICK SORT (S)	MERGE SORT (S)
1000	0.0706	0.0000	0.0128
10000	4.8851	0.0371	0.0246
100000	512.8139	0.2394	0.3172

## **4. Análise e Comparação dos Resultados**

### **4.1 Análise Geral**

A partir dos resultados obtidos, é possível observar uma grande diferença no desempenho entre os algoritmos. O Bubble Sort apresentou tempos de execução significativamente maiores que os outros dois algoritmos, especialmente para listas maiores. Já o Quick Sort e o Merge Sort se mostraram muito mais eficientes.

### **4.2 Comparação Detalhada**

#### **4.2.1 Listas Pequenas (1000 elementos)**

Para listas pequenas, todos os algoritmos apresentaram tempos de execução relativamente baixos. No entanto, o Quick Sort e o Merge Sort foram ligeiramente mais rápidos que o Bubble Sort.

#### **4.2.2 Listas Médias (10000 elementos)**

Neste caso, a diferença de desempenho ficou mais evidente. O Bubble Sort levou cerca de 4.88 segundos para ordenar a lista, enquanto o Quick Sort e o Merge Sort levaram apenas 0.0371 e 0.0246 segundos, respectivamente.

#### **4.2.3 Listas Grandes (100000 elementos)**

Para listas grandes, o desempenho do Bubble Sort se tornou impraticável, com um tempo de execução de aproximadamente 512.81 segundos. Em contraste, o Quick Sort e o Merge Sort continuaram sendo eficientes, com tempos de 0.2349 e 0.3172 segundos, respectivamente.

## 5. Identificação dos Algoritmos Mais Eficientes

### 5.1 Para Listas Pequenas

Para listas pequenas, tanto o Quick Sort quanto o Merge Sort se mostraram eficientes, com tempos de execução similares. O Bubble Sort também é viável, mas menos eficiente.

### 5.2 Para Listas Médias e Grandes

Para listas médias e grandes, o Quick Sort e o Merge Sort são claramente os algoritmos mais eficientes. O Quick Sort tende a ser ligeiramente mais rápido que o Merge Sort, mas ambos são significativamente melhores que o Bubble Sort.

## 6. Conclusão

Em resumo, o Quick Sort e o Merge Sort são os algoritmos mais recomendados para ordenação de listas, especialmente para listas médias e grandes. O Bubble Sort, embora seja simples de implementar, não é adequado para aplicações que requerem desempenho eficiente.

## 7. Gráficos

Os gráficos abaixo visualizam os tempos de execução para cada algoritmo e tamanho de lista:



