

Análise de Desempenho de Algoritmos Clássicos de Ordenação

Daniel Matos Marques

¹Instituto de Ciências Exatas
Pontifícia Universidade Católica de Minas Gerais (PUC-MG)

Resumo. *Este trabalho apresenta uma análise comparativa entre quatro algoritmos clássicos de ordenação: Selection Sort, Insertion Sort, Bubble Sort e Quicksort. Os algoritmos foram implementados em Java e avaliados quanto ao tempo de execução, número de comparações e movimentações realizadas em vetores de inteiros aleatórios com tamanhos de 100, 1000, 10000 e 100000 elementos. Os resultados são apresentados graficamente e discutidos criticamente, com destaque para a eficiência de cada algoritmo conforme o aumento da entrada.*

1. Introdução

Algoritmos de ordenação são fundamentais na computação, sendo amplamente utilizados em diversas aplicações. Este trabalho visa comparar algoritmos clássicos quanto ao desempenho prático em três aspectos principais: tempo de execução, número de comparações e movimentações de elementos.

2. Metodologia

Foram implementados os algoritmos de ordenação Selection Sort, Insertion Sort, Bubble Sort e Quicksort em Java. Cada algoritmo foi executado sobre vetores gerados aleatoriamente com tamanhos de 100, 1000, 10000 e 100000 elementos. Para cada execução, registrou-se:

- Tempo de execução (em milissegundos)
- Número de comparações
- Número de movimentações (trocas)

3. Resultados

3.1. Tempo de Execução

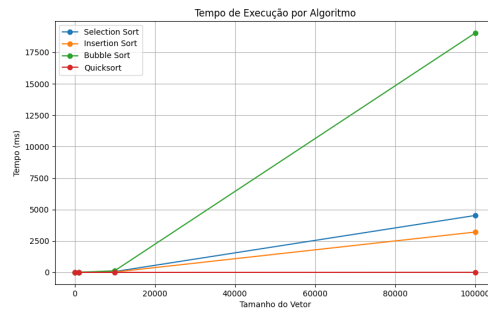


Figure 1. Tempo de execução por algoritmo e tamanho de entrada

3.2. Comparações

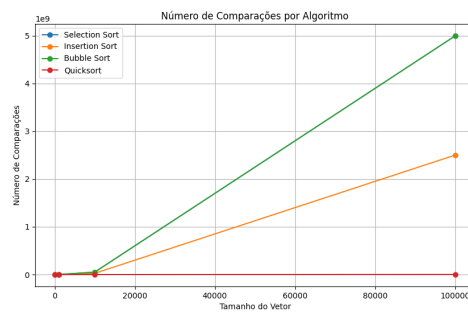


Figure 2. Número de comparações realizadas por cada algoritmo

3.3. Movimentações

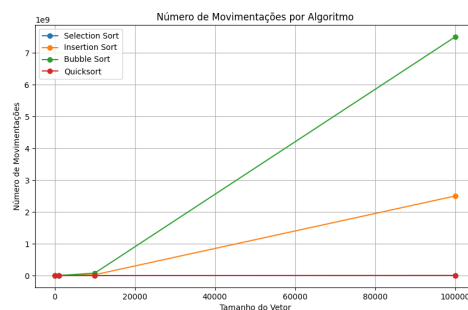


Figure 3. Número de movimentações realizadas por cada algoritmo

4. Análise Crítica

Os algoritmos de ordenação simples (Selection, Insertion e Bubble) demonstraram desempenho pobre para entradas grandes, com tempos de execução e número de operações crescendo rapidamente. O Quicksort, por outro lado, manteve desempenho superior mesmo em entradas com 100000 elementos.

Em termos de comparações, o Quicksort foi o mais eficiente, seguido pelo Insertion Sort para entradas menores. O Selection Sort, apesar de realizar menos movimentações, faz muitas comparações.

5. Conclusão

A análise evidencia que algoritmos de ordenação simples são adequados apenas para conjuntos pequenos. O Quicksort destacou-se como a melhor escolha para entradas maiores, sendo mais eficiente tanto em tempo quanto em operações. Esta comparação reforça a importância de escolher algoritmos adequados ao contexto e ao volume de dados.