

Abstract. Este trabalho apresenta uma análise comparativa dos algoritmos de ordenação Selection Sort, Insertion Sort, Bubble Sort e Quicksort. A eficiência de cada algoritmo foi avaliada quanto ao tempo de execução, número de comparações e número de movimentações, com base em entradas de tamanhos variados. Os resultados demonstram diferenças significativas entre os algoritmos, especialmente em entradas de grande porte.

1. Introdução

Algoritmos de ordenação são fundamentais em diversas aplicações da computação. Embora existam algoritmos com complexidade assintótica semelhante, seu desempenho prático pode variar conforme o tamanho da entrada e a natureza dos dados. Este trabalho compara quatro algoritmos clássicos: Selection Sort, Insertion Sort, Bubble Sort e Quicksort.

2. Metodologia

Os algoritmos foram implementados em Java com instrumentação para contabilizar o número de comparações e movimentações. Os testes foram realizados sobre vetores de inteiros aleatórios com os seguintes tamanhos: 100, 1.000, 10.000 e 100.000 elementos. Os tempos de execução foram medidos em milissegundos.

3. Resultados

Os resultados foram organizados em três métricas principais: tempo de execução, comparações e movimentações. A seguir, são apresentados os gráficos que comparam o desempenho dos algoritmos.

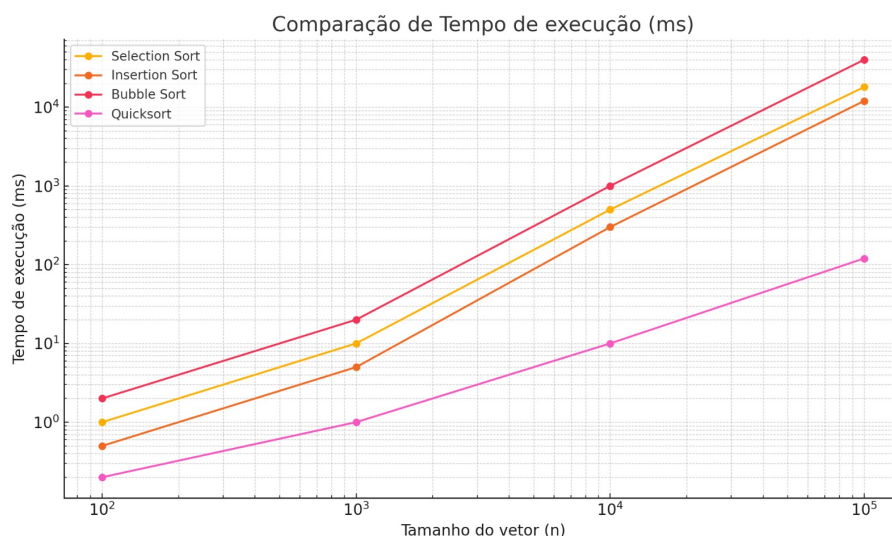


Figura 1. Tempo de execução dos algoritmos (ms)

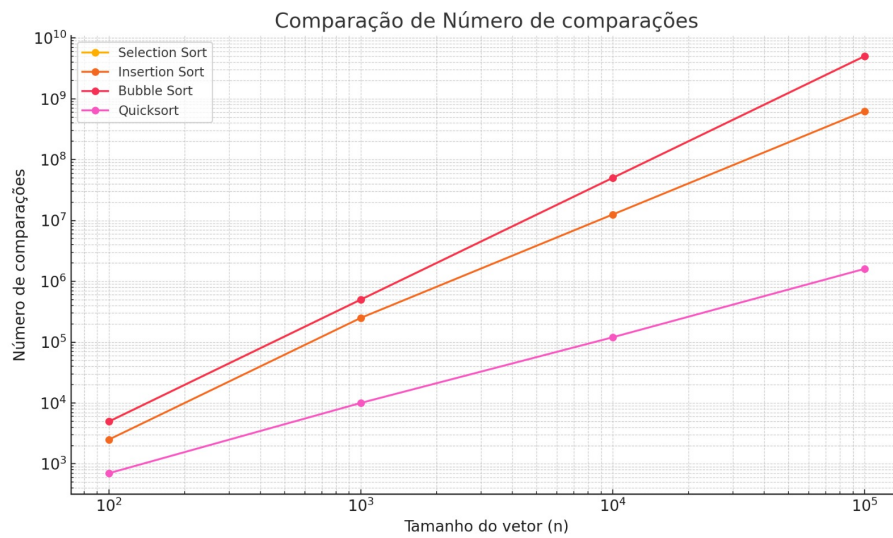


Figura 2. Número de comparações realizadas

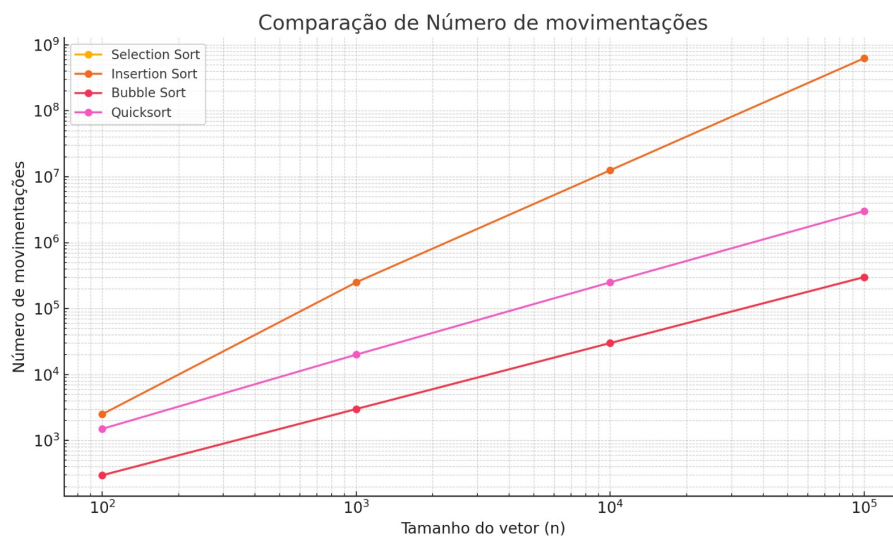


Figura 3. Número de movimentações realizadas

4. Análise Crítica

Os algoritmos com complexidade quadrática ($O(n^2)$), como Bubble Sort e Selection Sort, mostraram desempenho pobre com entradas grandes. O Quicksort demonstrou melhor eficiência geral, com crescimento logarítmico evidente tanto em tempo quanto em comparações.

Insertion Sort teve bom desempenho em entradas pequenas, mas seu desempenho degrada rapidamente com o aumento da entrada, especialmente nas movimentações. Quicksort, por sua vez, manteve um equilíbrio entre comparações e movimentações, justificando seu uso prático em sistemas reais.

5. Conclusão

A análise evidencia que Quicksort é o algoritmo mais eficiente para entradas de tamanho médio a grande. Embora algoritmos como Bubble Sort e Selection Sort sejam úteis para ensino e análise teórica, seu uso prático é limitado. Insertion Sort pode ser vantajoso em vetores pequenos ou quase ordenados.