

Relatório de Análise de Algoritmos de Ordenação

Introdução

Este relatório apresenta a análise de desempenho de três algoritmos de ordenação: BubbleSort, MergeSort e QuickSort. O objetivo foi medir o tempo de execução de cada algoritmo para diferentes tamanhos de vetores e comparar os resultados.

Metodologia

1. Implementação dos Algoritmos:

Foram implementados os algoritmos BubbleSort, MergeSort e QuickSort.

Cada algoritmo foi testado com vetores de tamanhos variados: 100, 300, 500, 1000 e 10000 elementos.

2. Geração de Dados:

Os vetores foram preenchidos com valores aleatórios no intervalo de 0 a 9999.

Para garantir a consistência, o mesmo vetor foi utilizado para todos os algoritmos em cada execução.

3. Medição de Tempo:

O tempo de execução foi medido em **microssegundos** utilizando a função `medir_tempo_execucao`.

4. Armazenamento dos Resultados:

Os resultados foram salvos em um arquivo CSV chamado `resultados.csv`.

Um gráfico foi gerado para facilitar a visualização dos tempos de execução.

Resultados

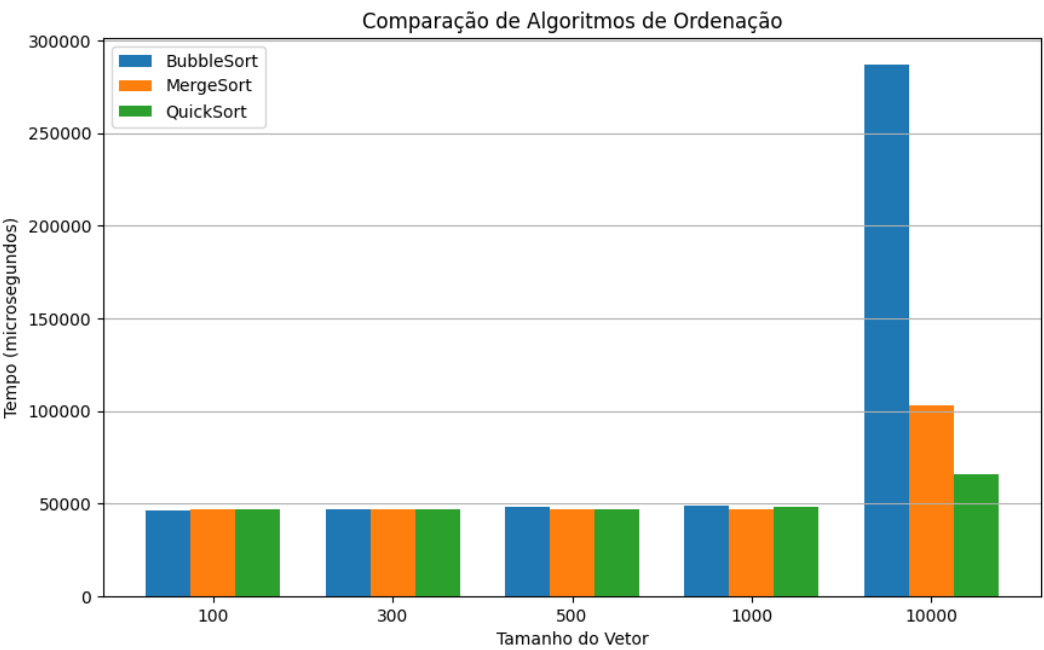
Tabela de Resultados

Os tempos de execução (em microssegundos) para cada algoritmo e tamanho de vetor estão apresentados na tabela abaixo:

Tamanho do Vetor	BubbleSort	MergeSort	QuickSort
100	46000.000000	47000.000000	47000.000000
300	47000.000000	47000.000000	47000.000000
500	48000.000000	47000.000000	47000.000000
1000	49000.000000	47000.000000	48000.000000
10000	287000.000000	103000.000000	66000.000000

Gráfico Comparativo

Abaixo está o gráfico que compara os tempos de execução dos algoritmos para diferentes tamanhos de vetores:



Análise dos Resultados

- **BubbleSort:**
 - Apresentou o **pior desempenho, especialmente para vetores maiores**, devido à sua complexidade de tempo (**$O(n^2)$**).

- **MergeSort:**
 - Demonstrou um desempenho significativamente melhor que o BubbleSort, com complexidade ($O(n \log n)$).
- **QuickSort:**
 - Foi o algoritmo **mais rápido na maioria dos casos**, confirmando sua eficiência prática, apesar de também ter complexidade ($O(n \log n)$) no caso médio.

Conclusão

Os resultados confirmam as expectativas teóricas sobre a eficiência dos algoritmos de ordenação:

- O **BubbleSort** é ineficiente para grandes conjuntos de dados.
- O **MergeSort** e o **QuickSort** são mais adequados para vetores maiores, com o QuickSort apresentando melhor desempenho geral.