

Relatório de Análise de Algoritmos de Ordenação

1. Visão Geral da Configuração de Benchmarking

- Bubble Sort: Complexidade de tempo $O(n^2)$.
- Insertion Sort: Complexidade de tempo $O(n^2)$, mas tem melhor desempenho em dados quase ordenados.
- Quick Sort: Complexidade de tempo média $O(n \log n)$, com otimizações para evitar o pior caso em dados ordenados ou ordenados de forma reversa.

Esses algoritmos foram testados em nove conjuntos de dados, categorizados da seguinte forma:

- Dados Aleatórios (aleatorio_): Não ordenados e em ordem aleatória.
- Dados Ordenados (crescente_): Já ordenados em ordem crescente.
- Dados Ordenados de Forma Reversa (decrecente_): Ordenados em ordem decrescente.

Cada categoria possui três tamanhos: 100, 1.000 e 10.000 elementos.

2. Análise dos Resultados de Benchmark

Dados Aleatórios (aleatorio_)

Dataset	Bubble Sort (ms)	Insertion Sort (ms)	Quick Sort (ms)
aleatorio_100.csv (100 elementos)	0,024 ms	0,010 ms	0,012 ms
aleatorio_1000.csv (1.000 elementos)	0,550 ms	0,087 ms	0,047 ms
aleatorio_10000.csv (10.000 elementos)	73,030 ms	3,653 ms	0,604 ms

Dados Ordenados (crescente_)

Dataset	Bubble Sort (ms)	Insertion Sort (ms)	Quick Sort (ms)
---------	------------------	---------------------	-----------------

crescente_100.csv (100 elementos)	0,00008 ms	0,0002 ms	0,003 ms
crescente_1000.csv (1.000 elementos)	0,0002 ms	0,0011 ms	0,027 ms
crescente_10000.csv (10.000 elementos)	0,002 ms	0,008 ms	0,300 ms

Dados Ordenados de Forma Reversa (decrecente_)

Dataset	Bubble Sort (ms)	Insertion Sort (ms)	Quick Sort (ms)
decrecente_100.csv (100 elementos)	0,013 ms	0,001 ms	0,003 ms
decrecente_1000.csv (1.000 elementos)	0,520 ms	0,071 ms	0,029 ms
decrecente_10000.csv (10.000 elementos)	52,121 ms	7,328 ms	0,341 ms

Interpretação e Observações

- * Bubble Sort apresenta escalabilidade quadrática, com tempos de execução que aumentam significativamente com o tamanho do conjunto de dados.
- * Insertion Sort também exibe comportamento quadrático, mas com desempenho melhor que o Bubble Sort.
- * Quick Sort mantém eficiência em escalabilidade, com seleção aleatória do pivô evitando cenários de pior caso.

Os tempos de execução escalam conforme esperado de acordo com a complexidade dos algoritmos, especialmente em dados já ordenados, onde Bubble e Insertion Sort se beneficiam da otimização. O Quick Sort mantém um desempenho consistente em diferentes distribuições.