Análise de Algoritmos de Ordenação

1. Introdução

Este relatório apresenta a análise de desempenho de nove algoritmos de ordenação: Insert Sort, Selection Sort, Comb Sort, Merge Sort, Quick Sort, Shell Sort, Radix Sort, Timsort e Stooge Sort. A avaliação foi realizada considerando três métricas principais: tempo de execução (ms), número de trocas e número de iterações.

Foram utilizados vetores de tamanhos 1000, 10000, 100000, 500000 e 1000000, com 5 rodadas para cada tamanho, utilizando seeds para garantir replicabilidade.

2. Análise dos Resultados

Durante os testes, observamos que algoritmos como QuickSort, MergeSort e RadixSort apresentaram excelente desempenho para grandes volumes de dados. Esses algoritmos mantiveram baixos tempos de execução, número reduzido de trocas e quantidade eficiente de iterações.

Por outro lado, algoritmos como InsertSort, SelectionSort e StoogeSort se mostraram extremamente ineficientes para vetores grandes. O tempo de execução e o número de operações crescem exponencialmente, tornando-os inviáveis em contextos que demandam performance.

O Timsort, algoritmo utilizado internamente por linguagens modernas como Python e Java, demonstrou desempenho sólido e estável, combinando características de ordenações por inserção e fusão. ShellSort e CombSort apresentaram desempenho intermediário, sendo mais eficientes que as ordenações simples (como Insert e Selection), mas inferiores aos algoritmos mais sofisticados como Quick e Merge.

3. Conclusão

A análise demonstra que, para aplicações que exigem alta performance, os algoritmos QuickSort, MergeSort e RadixSort são os mais recomendados. Eles oferecem um ótimo equilíbrio entre velocidade e eficiência computacional.

Algoritmos como InsertSort, SelectionSort e especialmente StoogeSort devem ser evitados em cenários que lidam com grandes volumes de dados, dado seu crescimento exponencial de tempo e custo computacional.

Por fim, Timsort, ShellSort e CombSort são boas opções intermediárias, apresentando desempenho aceitável em diversos cenários práticos.

4. Links

-https://youtu.be/kc7ncdFZSFI?si=s75le_rXh2VN93ym

-https://github.com/C4rlosJeronimo/Ordenacao.git