IFPI CAMPUS PICOS
CURSO: ANÁLISE E DESENVOLVIMENTO DE SISTEMAS
DISCIPLINA: ESTRUTURA DE DADOS I
TURMA: ADS-M2
TURNO: TARDE

ALUNOS : GABRIEL DE SOUSA FILHO, DÁRIO YGOR DE SOUSA LEAL, RAFAEL PEREIRA BORGES, JOÃO PEDRO RUIDIVALLE MEDEIROS DE AMORIM

Título do Trabalho

Comparação entre os algoritmos de ordenação

Resumo

Este artigo analisa os principais algoritmos de ordenação utilizados em computação, destacando suas diferenças, eficiência e aplicações. São abordados algoritmos como Bubble Sort, Insertion Sort, Selection Sort, Merge Sort e Quick Sort, comparando suas complexidades e performance em diferentes cenários. O objetivo é fornecer uma compreensão clara sobre como esses algoritmos funcionam e qual a melhor escolha dependendo do contexto.

1. Introdução

A ordenação de dados é uma das operações fundamentais em ciência da computação, influenciando diretamente o desempenho de sistemas e aplicações. Diferentes algoritmos de ordenação possuem características distintas em termos de complexidade computacional e eficiência. Este trabalho apresenta um estudo comparativo entre seis métodos de ordenação: Bubble Sort, Selection Sort, Insertion Sort, Merge Sort, Quick Sort e Counting Sort, analisando o tempo de execução para diferentes tamanhos de base de dados.

2. Metodologia

Para avaliar a eficiência dos algoritmos, foram utilizadas bases de dados de diferentes

tamanhos (100, 500, 1000, 5000, 30000, 80000, 100000, 150000 e 200000 elementos). Os testes foram realizados em um ambiente controlado, medindo o tempo de execução de cada algoritmo. O tempo foi registrado em segundos para possibilitar a comparação entre os métodos.

3. Resultados e Discussão

3.1 Bubble Sort

O Bubble Sort apresentou os piores tempos de execução, tornando-se inviável para bases de dados grandes. Para 200.000 elementos, levou cerca de 174,79 segundos, evidenciando sua complexidade quadrática $O(n^2)$.

Bubble Sort

Base de Dados	Tempo (s)
100	0.000154
500	0.003049
1000	0.011554
5000	0.148254
30000	4.009516
80000	27.513993
100000	42.886987
150000	96.714914
200000	174.790515

3.2 Selection Sort

O Selection Sort também apresentou um desempenho insatisfatório para bases maiores. Com 68,49 segundos para 200.000 elementos, foi mais eficiente que o Bubble Sort, mas ainda impraticável para grandes volumes de dados devido à complexidade $O(n^2)$.

Selection Sort

Base de Dados	Tempo (s)
100	0.000029
500	0.000502
1000	0.001812
5000	0.049034
30000	1.573784
80000	11.120934
100000	17.155942
150000	38.573908
200000	68.496389

3.3 Insertion Sort

O Insertion Sort superou o Bubble e o Selection Sort, especialmente para bases menores. Para 200.000 elementos, executou em 38,11 segundos, evidenciando que, apesar de ser $O(n^2)$ no pior caso, pode ser mais eficiente que os anteriores em alguns cenários.

Insertion Sort

Base de Dados	Tempo (s)
100	0.000019
500	0.000293
1000	0.001037
5000	0.031017
30000	0.891573
80000	6.123621
100000	9.563467
150000	21.470546
200000	38.111177

3.4 Merge Sort

O Merge Sort, com complexidade O(n log n), demonstrou excelente desempenho. Mesmo para 200.000 elementos, levou apenas 0,078 segundos, destacando-se como uma das opções mais eficientes para ordenação em grandes volumes de dados.

Merge Sort

Base de Dados	Tempo (s)
100	0.000031
500	0.000175
1000	0.000337
5000	0.001795
30000	0.010494
80000	0.030625
100000	0.036167
150000	0.058085
200000	0.078517

3.5 Quick Sort

O Quick Sort apresentou tempos competitivos, sendo um dos melhores métodos testados.

Ordenou 200.000 elementos em 0,72 segundos, mostrando-se eficiente na prática, apesar de sua complexidade O(n log n) no caso médio e $O(n^2)$ no pior caso.

Quick Sort

Base de Dados	Tempo (s)
100	0.000013
500	0.000067
1000	0.000144
5000	0.001112
30000	0.019450
80000	0.126312
100000	0.189212
150000	0.410025
200000	0.720172

3.6 Counting Sort

O Counting Sort foi o mais rápido entre todos os algoritmos testados, com apenas 0,0048 segundos para 200.000 elementos. Como possui complexidade O(n + k), sua eficiência depende do intervalo dos valores a serem ordenados.

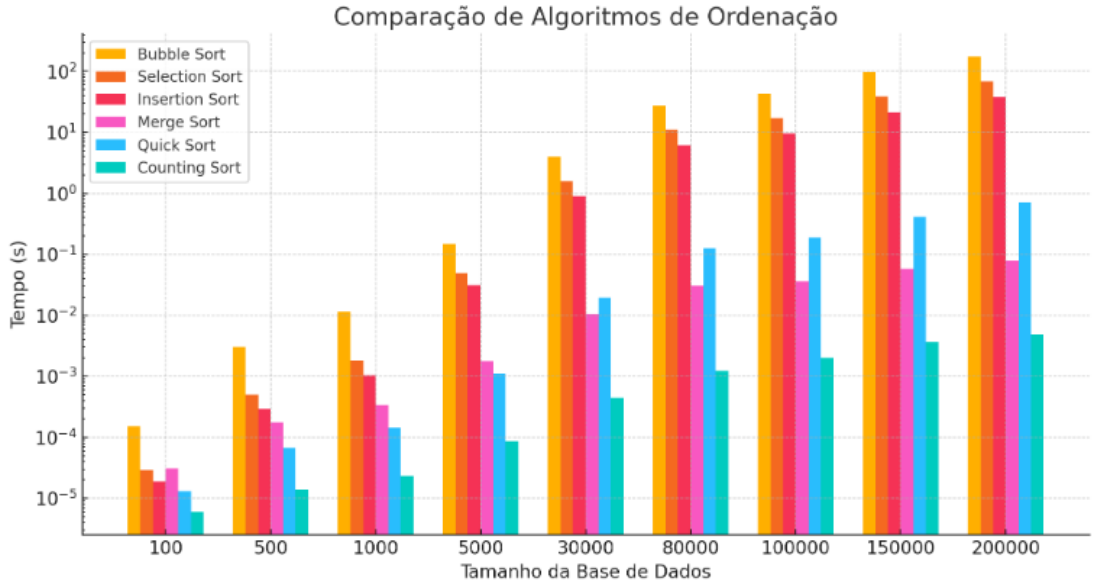
Counting Sort

Base de Dados	Tempo (s)
100	0.000006
500	0.000014
1000	0.000023
5000	0.000086
30000	0.000441
80000	0.001233
100000	0.002024
150000	0.003643
200000	0.004876

3.7 Gráfico Geral

O gráfico representa a comparação de tempos de execução de seis algoritmos de ordenação (Bubble Sort, Selection Sort, Insertion Sort, Merge Sort, Quick Sort e Counting Sort) para diferentes tamanhos de entrada. O eixo X mostra o tamanho da base

de dados, variando de 100 a 200.000 elementos, enquanto o eixo Y exibe o tempo de execução em segundos, plotado em escala logarítmica para melhor visualização das diferenças entre os algoritmos.



4. Conclusão

Os experimentos confirmam que algoritmos com complexidade O(n log n) (Merge Sort e Quick Sort) são significativamente mais rápidos que os de complexidade $O(n^2)$ (Bubble Sort, Selection Sort e Insertion Sort). O Counting Sort se destacou como o mais eficiente, mas sua aplicação é restrita a cenários onde o intervalo dos dados permite seu uso.

Assim, para aplicações gerais, o Merge Sort e o Quick Sort são as melhores opções, garantindo desempenho otimizado mesmo para grandes volumes de dados.