Relatório de Desempenho de Algoritmos de Ordenação

Alejandro e Eduardo

1 Introdução

Este relatório analisa o desempenho de diferentes algoritmos de ordenação: Insertion Sort, Quick Sort, Shell Sort e Counting Sort. O desempenho é avaliado em termos de tempo de execução (em nanosegundos) e número de trocas realizadas para diferentes tamanhos de entrada.

2 Resultados

2.1 Desempenho dos Algoritmos

Tamanho	Algoritmo	Tempo (ns)	Trocas
1000	Insertion Sort	2,389,800	243,584
	Quick Sort	418,800	0
	Shell Sort	353,200	7,261
	Counting Sort	4,527,000	0
10,000	Insertion Sort	6,499,200	24,800,299
	Quick Sort	519,200	0
	Shell Sort	1,973,700	143,021
	Counting Sort	2,160,200	0
100,000	Insertion Sort	685,541,500	2,496,191,152
	Quick Sort	6,453,500	0
	Shell Sort	10,919,500	2,795,133
	Counting Sort	2,928,500	0
500,000	Insertion Sort	17,857,821,800	62,556,855,694
	Quick Sort	35,919,600	0
	Shell Sort	68,413,500	20,711,885
	Counting Sort	6,503,300	0
1,000,000	Insertion Sort	73,053,899,800	250,193,923,181
	Quick Sort	75,831,200	0
	Shell Sort	154,214,200	50,648,787
	Counting Sort	8,995,500	0

Table 1: Desempenho dos algoritmos de ordenação para diferentes tamanhos de entrada.

2.2 Análise do Desempenho

- **Quick Sort**: Este algoritmo se destacou como o mais eficiente em todos os tamanhos de entrada, com tempos de execução baixos e sem trocas realizadas.
- **Insertion Sort**: Apresentou desempenho inferior, com tempos de execução crescentes drasticamente e um alto número de trocas, tornando-o inadequado para entradas maiores.

- **Shell Sort**: Ofereceu um desempenho intermediário, com tempos de execução razoáveis, mas ainda acima do Quick Sort.
- **Counting Sort**: Demonstrou alta eficiência em tempo, especialmente em conjuntos de dados maiores, e não realizou trocas, embora seja limitado a tipos de dados específicos.

3 Conclusão

Este estudo revelou diferenças significativas no desempenho dos algoritmos de ordenação analisados. O **Quick Sort** se destacou como a opção mais eficiente para entradas de grande dimensão, com tempos de execução muito inferiores e nenhuma troca, o que o torna ideal para aplicações práticas que exigem alta performance.