

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO PARANÁ
Resolução de Problemas Estruturados em Computação –
CIÊNCIA DA COMPUTAÇÃO

Professor: Andrey Cabral Meira

Estudante: Nicolas Andreas Jackel



Relatório de Implementação e Análise dos Algoritmos Counting Sort e Quick Sort

Repositório GitHub: <https://github.com/concertmaster40/TDE3-EstruturaDados>

Objetivo:

Neste relatório está contida a análise dos algoritmos Counting Sort e Quick Sort, implementados em JAVA. Foram analisadas as métricas de desempenho a partir de execuções em arrays com cinco tamanhos variados e elementos aleatórios, gerados com uma seed fixa.

Algoritmos e Métricas Avaliadas:

As métricas coletadas durante os testes incluem:

- Tempo de execução
- Total de trocas (apenas no Quick Sort, pois o Counting Sort não efetua trocas, apenas gera array)
- Total de iterações

Os algoritmos foram testados em arrays com os seguintes tamanhos: 1.000, 10.000, 100.000, 500.000 e 1.000.000 de elementos. Cada tamanho foi executado cinco vezes.

Resultados Obtidos:

Counting Sort:

Tamanho do Array	Tempo Médio de Execução (ms)	Iterações Médias
1.000	7.949	1002000.00
10.000	10.623	1020000.00
100.000	5.329	1200000.00
500.000	13.247	2000000.00
1.000.000	18.585	3000000.00

Quick Sort:

Tamanho do Array	Tempo Médio de	Trocas Médias	Iterações Médias
------------------	----------------	---------------	------------------

	Execução (ms)		
1.000	0.292	6051.60	11997.80
10.000	1.901	83087.40	165156.60
100.000	11.225	1057579.00	2129476.20
500.000	71.372	6400647.00	12209345.80
1.000.000	128.456	13342420.00	25669866.60

Análise dos Resultados:

Desempenho:

O Counting Sort apresentou um desempenho estável e rápido, principalmente para arrays grandes, já que sua complexidade é $O(n+k)$. Já o Quick Sort mostrou um aumento considerável no número de trocas e iterações conforme o tamanho do array crescia, o que está alinhado com sua complexidade média de $O(n \log n)$.

Comparação e Conclusões:

O Counting Sort apresentou uma média de tempo muito mais curta do que o Quick Sort para organizar os arrays, outro ponto é que o Quick Sort faz muito mais iterações para retornar o array final, no Counting Sort é muito menos, como podemos ver no gráfico. Contudo, o Quick Sort se mostra muito mais vantajoso se pensarmos em flexibilidade, pois não necessita saber o valor máximo de elementos no array previamente

Gráficos:

Gráficos de Comparação de Desempenho de Tempo e Iterações:

