

# Resolução de Problemas Estruturados em Computação – PUCPR

TDE 03 - Ordenação

**Alunos:** Fernando Moro

## RELATÓRIO:

Para o trabalho de análise e implementação dos algoritmos de ordenação, foram escolhidos os algoritmos **QUICK SORT** e **GNOME SORT**. Ambos implementados em Java.

As métricas a serem analisadas foram:

- 1- Tempo de execução
- 2- Total de trocas (swaps)
- 3- Total de iterações

Para isso foram executados os algoritmos de ordenação em arrays aleatórios de tamanho 1000, 10.000, 100.000, 500.000 e 1.000.000. Cada um com 5 gerações com seeds diferentes. As métricas então foram calculadas a partir da média das cinco execuções de cada tamanho de array.

Os resultados (médias) obtidos podem ser vistos em formato textual a seguir:

## QUICK SORT:

Array de **1.000** elementos:

Tempo de execução: 0,4 milissegundos

Total de trocas: 2576,4

Total de iterações: 2858

Array de **10.000** elementos:

Tempo de execução: 0,8 milissegundos

Total de trocas: 33651,8

Total de iterações: 36545

Array de **100.000** elementos:

Tempo de execução 7,2 milissegundos

Total de trocas: 412461

Total de iterações: 441031,4

Array de **500.000** elementos:

Tempo de execução: 42,4 milissegundos

Total de trocas: 2333132,6

Total de iterações: 2476136,6

Array de **1.000.000** elementos:

Tempo de execução: 86,4 milissegundos

Total de trocas: 4872887

Total de iterações: 5158606,2

## **GNOME SORT:**

Array de **1.000** elementos:

Tempo de execução: 2 milissegundos

Total de trocas: 252044,6

Total de iterações: 505090,2

Array de **10.000** elementos:

Tempo de execução: 47,6 milissegundos

Total de trocas: 24910660,8

Total de iterações: 49831322,6

Array de **100.000** elementos:

Tempo de execução 5864,8 milissegundos

Total de trocas: 2098913456,76

Total de iterações: 4997148227,8

Array de **500.000** elementos:

Tempo de execução: 147344,6 milissegundos

Total de trocas: 62504741626,6

Total de iterações: 125009983254,2

Array de **1.000.000** elementos:

Tempo de execução: 588193,4 milissegundos

Total de trocas: 250029799060,8

Total de iterações: 500060598122,6

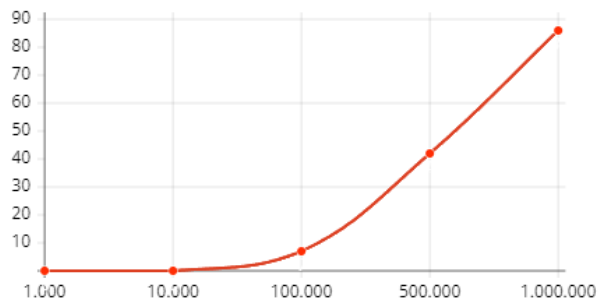
## GRÁFICOS:

Devido a grande diferença de performance, não fazia exibir os resultados dos 2 algoritmos de ordenação no mesmo gráfico, pois prejudicava a sua legibilidade. Então separei em gráficos para ver a progressão das métricas com o decorrer do crescimento no número de elementos no array.

Para visualizar a diferença gritante de performance, basta verificar a diferença da grandeza dos valores do eixo 'y' de ambos os gráficos. Que no **GNOME SORT** se dá principalmente a partir de arrays de tamanho 500.000.

## Quick Sort

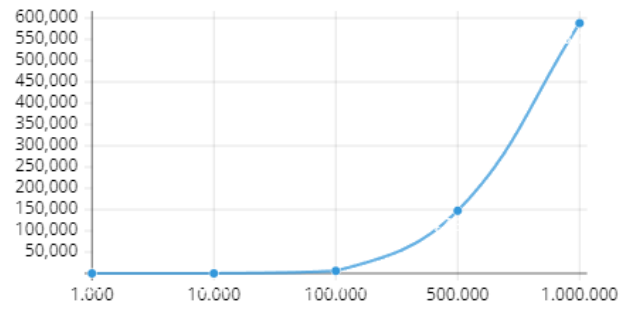
Tempo por tamanho do array



Made with Livegap Charts

## Gnome Sort

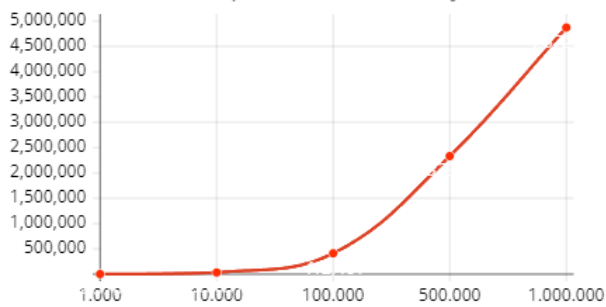
Tempo por tamanho do array



Made with Livegap Charts

## Quick Sort

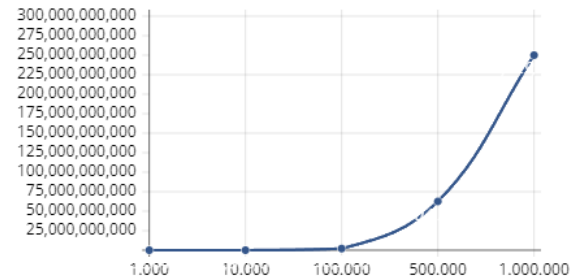
Trocas por tamanho do array



Made with Livegap Charts

## Gnome Sort

Trocas por tamanho do array



Made with Livegap Charts

## Quick Sort

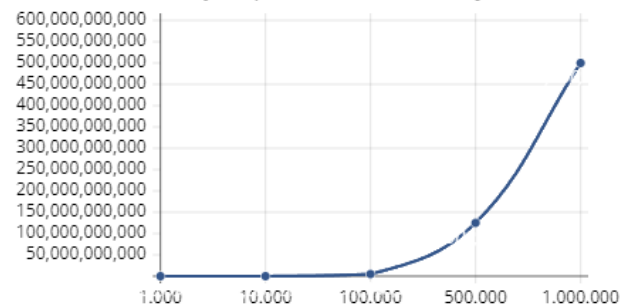
Iterações por tamanho do array



Made with Livegap Charts

## Gnome Sort

Iterações por tamanho do array



Made with Livegap Charts