

UNIVERSIDADE FEDERAL DO PIAUÍ  
CENTRO DE CIÊNCIAS NATURAIS  
DEPARTAMENTO DE COMPUTAÇÃO  
ESTRUTURA DE DADOS AVANÇADO

COMPARAÇÃO DO DESEMPENHO ENTRE OS MÉTODOS DE  
ORDENAÇÃO BUBBLESORT, INSERTIONSORT, MERGESORT,  
QUICKSORT, SELECTIONSORT E SHELLSORT

Wesley de Aquino Ferreira

Teresina – PI

2019

## 1. INTRODUÇÃO

Armazenar dados é de fundamental importância para as atividades computacionais, porém muitas vezes é preciso ordenar esses dados segundo algum critério para ganho de eficiência ou tempo.

Diversos métodos de ordenação de dados já foram criados para realizar este procedimento. Cada um deles possui características e finalidades específicas, e não existe o método ideal para todos os tipos de situação. É dever do programador estudar o problema e decidir qual a melhor forma de ordenar os dados trabalhados.

## 2. METODOLOGIA

Para esse trabalho foram analisados 6 métodos de ordenação de dados: *Bubble Sort*, *Insertion Sort*, *Merge Sort* e *Quick Sort*, *Selection Sort* e *Shell Sort*.

*Bubble Sort* é provavelmente o método mais conhecido por ser de fácil entendimento e implementação. *Merge Sort* e *Quick Sort* são métodos de ordenação por particionamento muito difundidos, sendo que o segundo é geralmente o mais utilizado em grandes projetos, por ser muito eficiente, *Insertion Sort* é eficiente quando aplicado a um pequeno número de elementos, já o é um algoritmo de ordenação baseado em se passar sempre o menor valor para a primeira posição (ou o maior dependendo da ordem requerida), depois o de segundo menor valor para a segunda posição, e assim é feito sucessivamente com os  $n-1$  elementos restantes, até os últimos dois elementos.

A implementação desses métodos não será analisada nesse trabalho, pois é tópico amplamente abordado em qualquer livro ou site sobre programação e estrutura de dados.

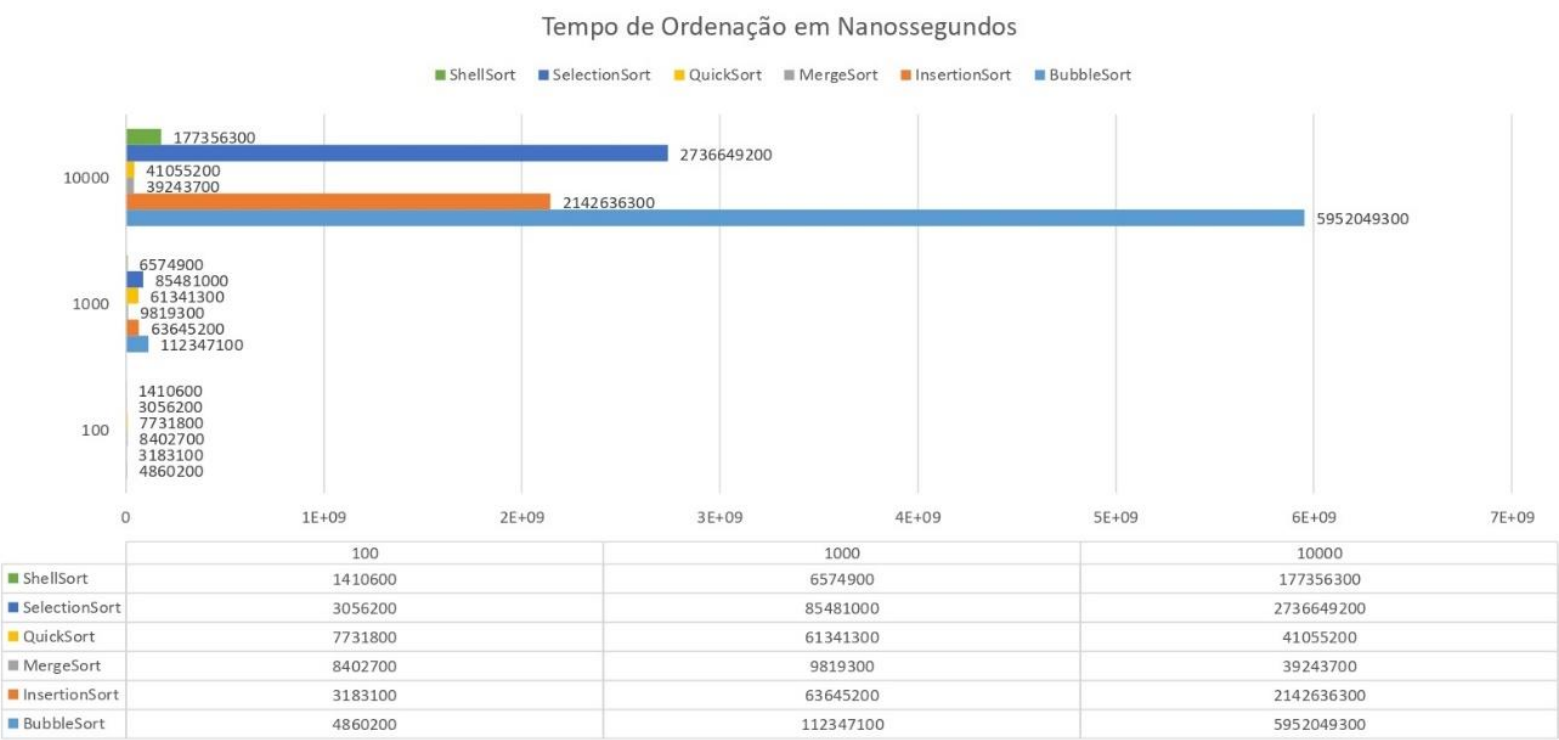
Os dados utilizados para essa análise foram nomes (Strings) aleatórios lidos de 3 arquivos diferentes, o primeiro com 100 nomes, o segundo com 1000 e o terceiro com 10000. Os armazenamentos desses nomes foram feitos em listas estáticas.

Foi utilizada a linguagem de programação Java, versão 8. O tempo de ordenação foi medido em nanossegundos com o método `nanoTime()` da classe padrão `System`, contado a partir do início do processo de ordenação, até sua conclusão.

Para auxiliar na análise desses algoritmos também foi medido a quantidade de comparações feitas entre os “nós” das estruturas de dados durante o processo de ordenação.

3. RESULTADOS

3.1. TEMPO DE ORDENAÇÃO (em nanossegundos)



3.2. NÚMERO DE COMPARAÇÕES



#### 4. ANÁLISE DOS RESULTADOS

Observando os gráficos verificamos que nas ordenações com 100 dados, todos apresentaram valores aproximados de tempo, mas na comparação o Bubble Sort ficou bastante na frente, com 1000 dados o Bubble Sort realizou a ordenação em um tempo um pouco maior que os outros, mas ficou na média em relação as comparações. Porém para 10000 nomes levou quase o dobro do tempo que os outros métodos, tendo também um número muito alto de comparações, o Insertion Sort e o Selection Sort ficaram logo atrás do Bubble Sort em relação ao tempo, entretanto o número de comparações de ambos foi baixo. Para 1000 dados apresentou desempenho semelhante aos outros.

O método Merge Sort apresentou o melhor desempenho para ordenar 1000 dados, e 10000 dados tendo feito também uma das menores quantidades de comparações para essas quantidades de dados. Com 100 dados fez uma quantidade mínima de comparações, porém apresentou desempenho similar ao método Quick Sort.