Trabalho Prático de Algoritmos e Estruturas de Dados 1

IFMG Campus Ouro Branco

Rodrigo Augusto Correa Soares - 0039063

1. Introdução

A ordenação de dados é uma tarefa fundamental na computação e desempenha um papel vital na manipulação eficiente de informações em uma ampla gama de aplicações. A capacidade de organizar conjuntos de dados de forma ordenada é essencial em áreas que vão desde bancos de dados até processamento de imagens e algoritmos de busca.

Neste trabalho foram implementados quatro algoritmos de ordenação (Selection Sort, Insertion Sort, Merge Sort e Quick Sort). Foi feita a analise de complexidade de cada um deles e os testes para exemplificar essa análise.

2. Analises de complexidade

2.1 Selection Sort

A abordagem do Selection Sort consiste em iterar repetidamente por todo o conjunto de dados, encontrar o menor elemento e movê-lo para a primeira posição. Executando esses passos repetidas vezes o vetor é ordenado.

O Selection Sort não é um algoritmo estável, o que significa que ele não preserva a ordem relativa de elementos com valores iguais. A complexidade do Selection Sort é caracterizada por uma busca pelo menor elemento em cada passo, que é executada n-1 vezes, onde "n" é o tamanho do vetor. Portanto, a complexidade do Selection Sort é $O(n^2)$.

2.2 Insertion Sort

O Insertion Sort é um algoritmo de ordenação que se baseia em construir uma lista ordenada um elemento por vez. Ele começa com um elemento do conjunto e insere esse elemento na posição correta na lista ordenada, deslocando os

elementos maiores à direita. Esse processo é repetido para cada elemento do conjunto, resultando em uma lista completamente ordenada.

O Insertion Sort é um algoritmo estável, o que significa que ele preserva a ordem relativa de elementos com valores iguais. Sua complexidade é quadrática, com um desempenho médio de $O(n^2)$, tornando-o mais eficiente em conjuntos de dados pequenos ou quase ordenados.

2.3 Merge Sort

O Merge Sort é um algoritmo de ordenação dividir para conquistar. Ele divide o conjunto de dados em duas metades, ordena cada metade separadamente e, em seguida, mescla as duas metades ordenadas para produzir um único conjunto de dados ordenado. Esse processo é repetido recursivamente até que todo o conjunto de dados esteja ordenado.

O Merge Sort é um algoritmo estável e sua complexidade é $O(n \log n)$, o que o torna eficiente para grandes conjuntos de dados.

2.4 Quick Sort

O Quick Sort é outro algoritmo de ordenação dividir para conquistar que se baseia no conceito de pivô. Ele escolhe um elemento como pivô e reorganiza os elementos de forma que os elementos menores que o pivô estejam à esquerda e os elementos maiores à direita. Em seguida, aplica o mesmo processo nas duas metades, à esquerda e à direita do pivô.

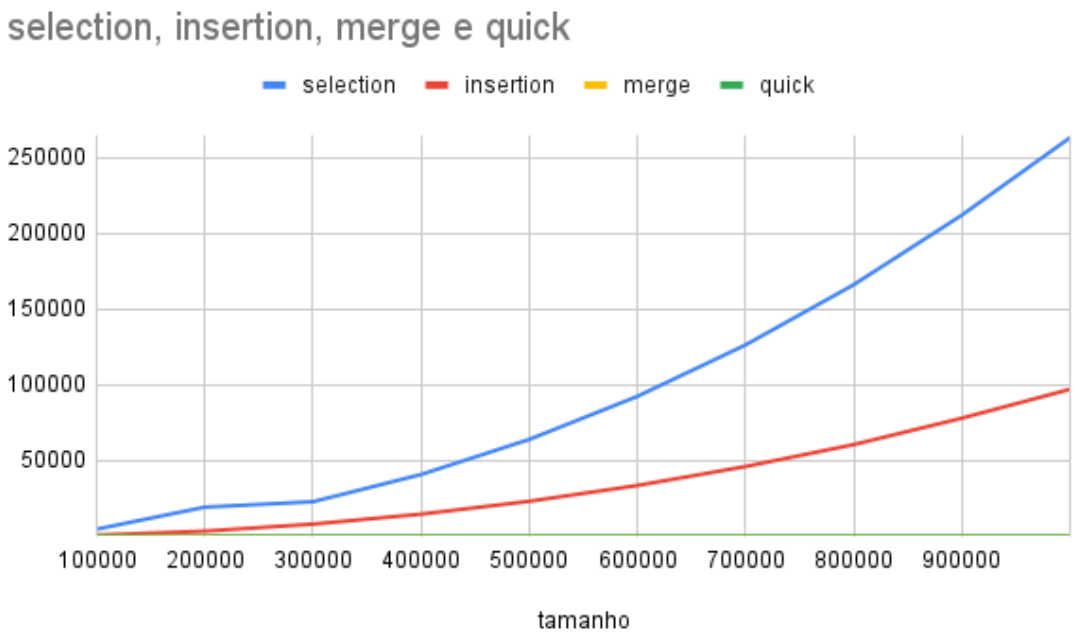
O Quick Sort não é estável e sua complexidade média é $O(n \log n)$. No entanto, em casos raros, o pior cenário pode levar à complexidade quadrática.

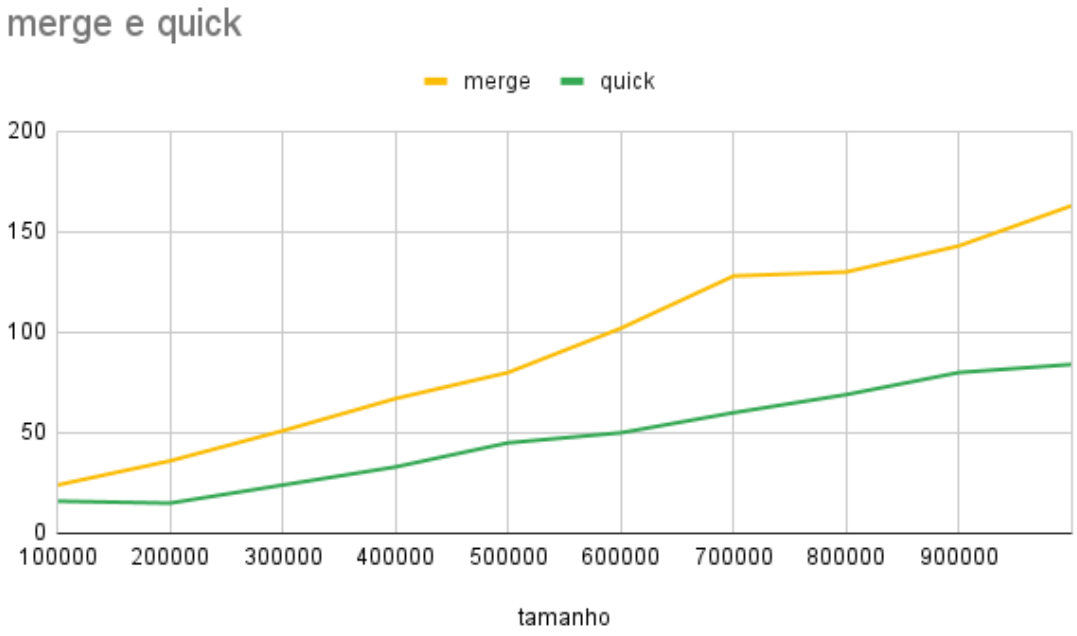
3. Testes e resultados

Os testes de desempenho foram conduzidos em diferentes volumes de dados, variando de 100.000 a 1 milhão, com incrementos de 100.000 a cada iteração. O primeiro conjunto de testes foi executado em dados completamente aleatórios.

O segundo e o terceiro com dados ordenados em ordem crescente e decrescente.

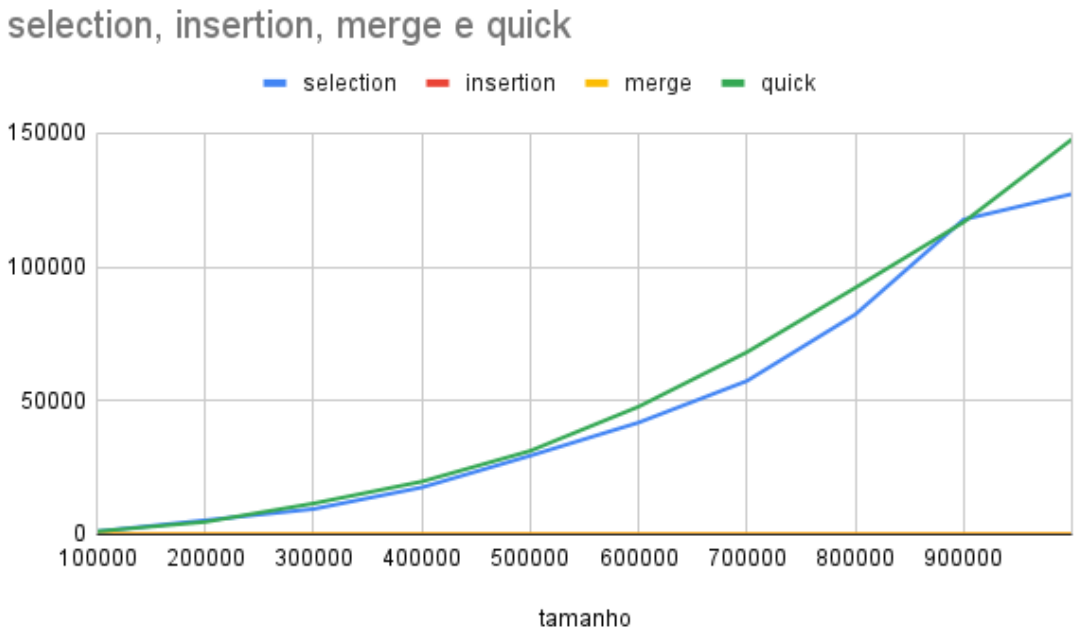
3.1 Valores aleatórios:

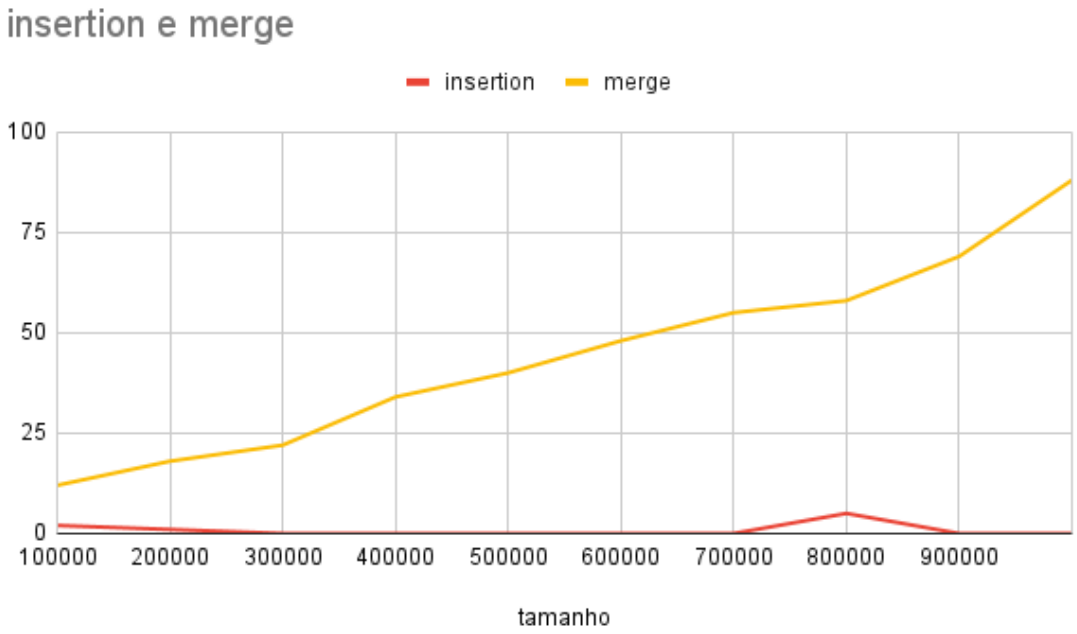




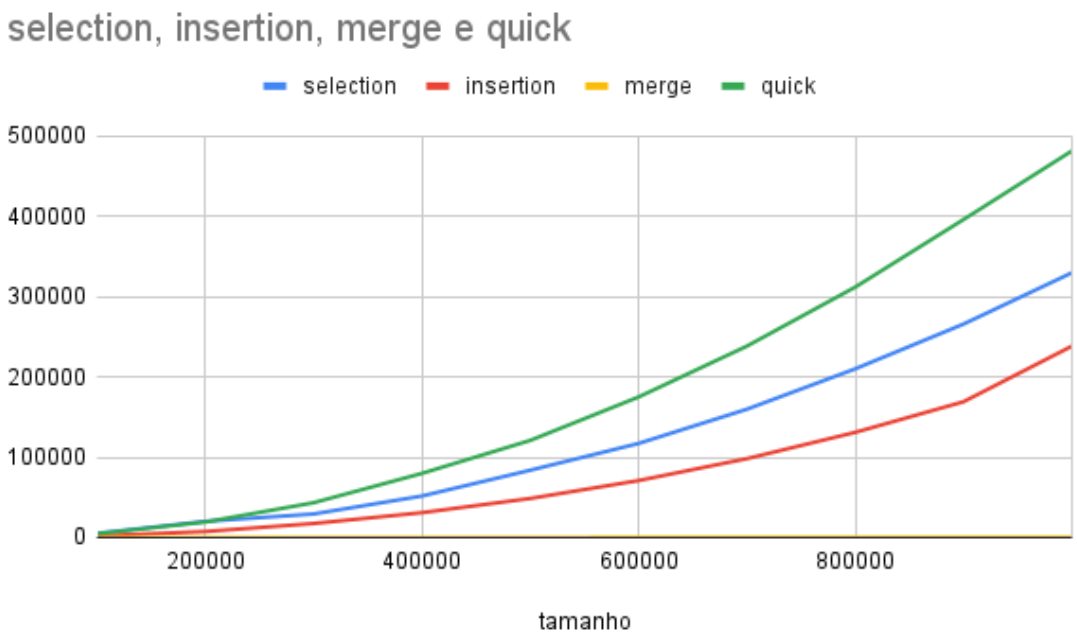
3.2 Valores ordenados em ordem crescente:

Aqui fica evidente o Quick Sort operando no pior caso e o Insertion Sort no melhor caso.





3.3 Valores ordenados em ordem decrescente:



4. Conclusão

Em resumo, o Insertion Sort é eficaz para conjuntos pequenos, o Merge Sort é eficiente para conjuntos grandes, e o Quick Sort combina eficiência e desempenho aceitável na maioria dos casos. Cada um desses algoritmos tem suas características e é escolhido com base no tamanho do conjunto de dados e em outras considerações específicas do problema a ser resolvido.