

Introdução

Foram utilizados cinco métodos de ordenação, onde foram passados diferentes tipos de combinações para que pudessem ser calculados os tempos de execução de cada um deles.

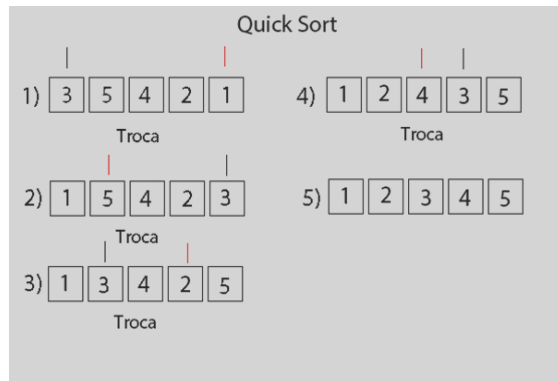
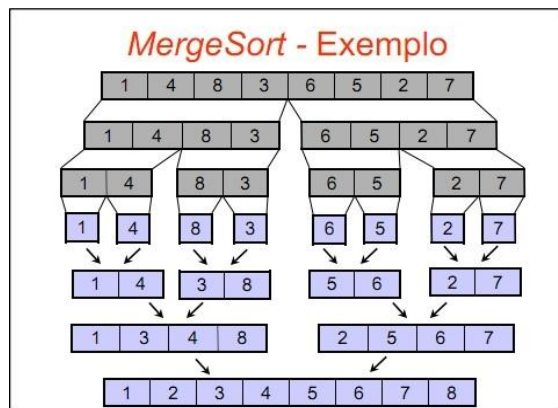
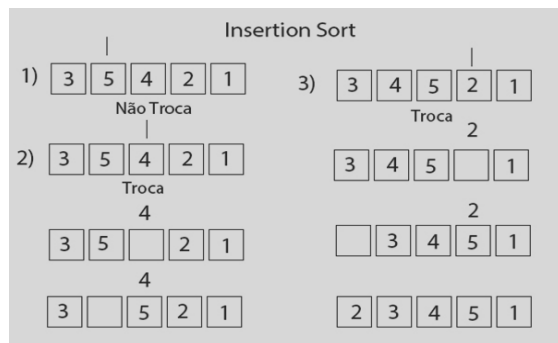
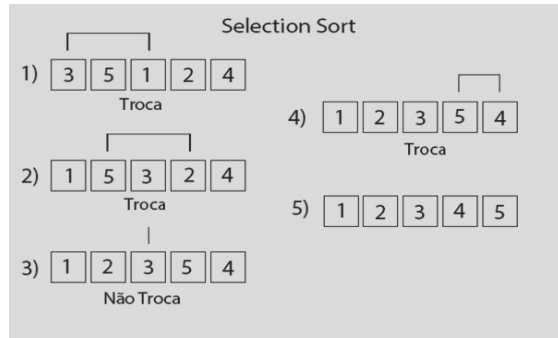
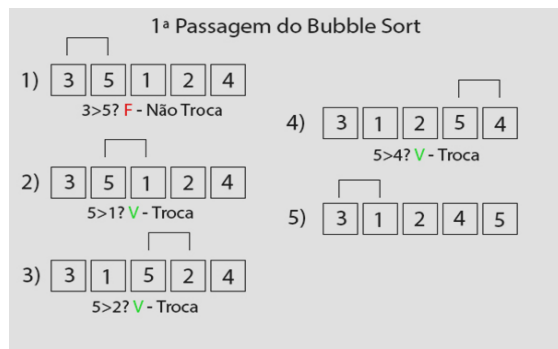
Do primeiro, mais simples e mais lento, foi apresentado o bubbleSort, aquele que no qual faz o caminho em um vetor, analisando cada posição, fazendo isso diversas vezes, ele combina cada elemento até que todos estejam organizados da maneira solicitada;

Em segundo foi feito selectionSort, basicamente é feito a seleção do menor ou maior valor dentro do vetor e vai o posicionando no início do vetor, ou seja, ele pega o valor passado e coloca no começo deste vetor, e vai buscando os menores ou maiores valores para assim ir organizando esta array.

Em terceiro foi utilizado o InsertionSort, onde o algoritmo faz a organização começando da esquerda para a direita, selecionando os valores e mudando as posições, de maneira que sempre a parte da esquerda vá ficando ordenada.

Em quarto foi utilizado o MergeSort que é um exemplo de algoritmo de ordenação que faz uso da estratégia “dividir para conquistar” para resolver problemas. Esse algoritmo divide o problema em pedaços menores, resolve cada pedaço e depois junta (merge) os resultados. O vetor será dividido em duas partes iguais, que serão cada uma divididas em duas partes, e assim até ficar um ou dois elementos cuja ordenação é trivial.

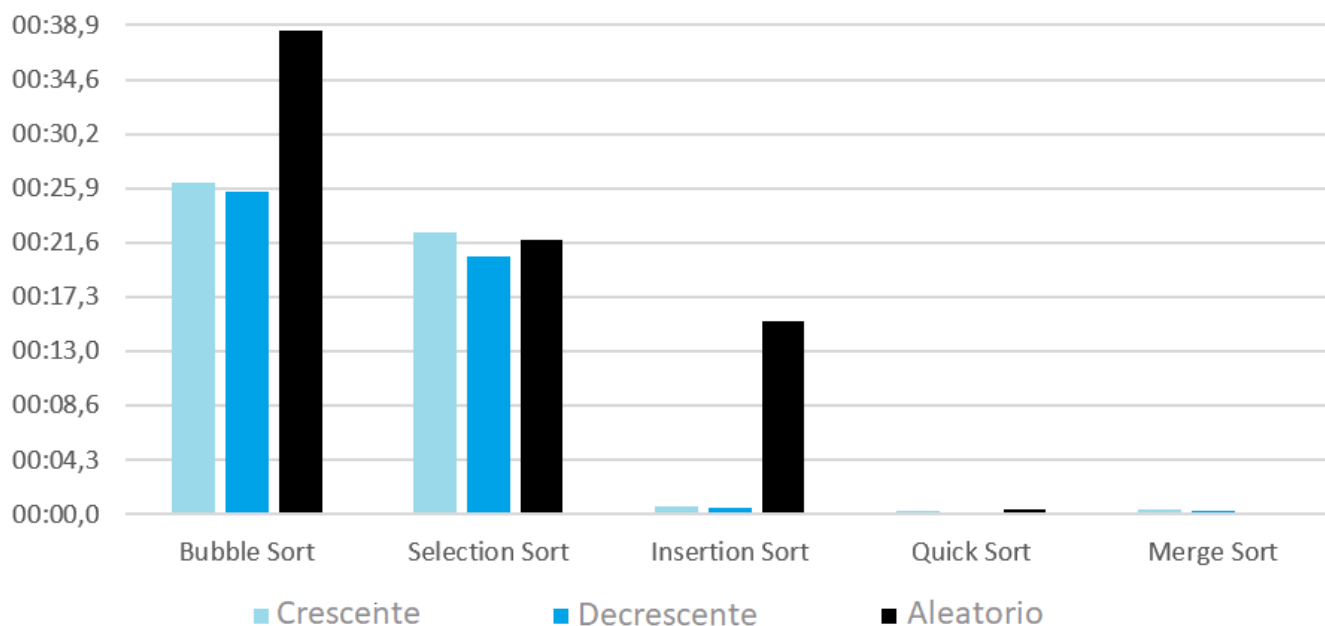
E por ultimo, O Quicksort que é o algoritmo mais eficiente na ordenação por comparação. Nele se escolhe um elemento chamado de pivô (ou como utilizado dentro do código como *“intermédio”*), a partir disto é organizada a lista para que todos os números anteriores a ele sejam menores que ele, e todos os números posteriores a ele sejam maiores que ele. Ao final desse processo o número pivô ou *“intermédio”* já está em sua posição final.



Implementação

Foram implementados três vetores, no qual cada um armazenaria 100 mil posições, e cada um deles organizariam de forma, crescente, decrescente e aleatoriamente utilizando cada um dos algoritmos de ordenação. Dentro de métodos, foram implementadas a estruturas dos algoritmos e a estrutura de cálculos para analisar o tempo de execução destes algoritmos.

Média de tempos para cada algoritmo



Conclusão

Concluiu-se que, existem diversas maneiras para se chegar a um resultado, foram apresentadas dentro deste mesmo trabalho, 5 maneiras para executar uma simples forma de organização de números, e ainda a mostra de comparações que existe entre as escolhas de algoritmos para determinadas execuções. O trabalho então consistiu em pesquisa, pratica e testes, para que se chegasse a um resultado final. A maior dificuldade encontrada foi na implementação, onde foi necessário buscar diversas ferramentas de pesquisas e vídeos para que pudesse chegar ao ponto atual. Foi de grande experiência claro, pois tive a oportunidade de conhecer um código “complexo” que é necessário ter bastante atenção para escreve-lo.