

Aluno: Sizenando Souza França

RGM: 50575

Disciplina: Algoritmo e Estruturas de Dados II

## Introdução e configuração

O trabalho realizado implementa os seguintes algoritmos de ordenação:

1. Bubble-sort original;
2. Bubble-sort melhorado;
3. Insertion-sort;
4. Mergesort;
5. Quicksort com pivô sendo o último elemento;
6. Quicksort com pivô sendo um elemento aleatório;
7. Quicksort com pivô sendo a mediana de três;
8. Heapsort;

Cada algoritmo de ordenação é executado em três categorias a quais serão comentadas ao longo desse relatório, sendo elas:

1. Crescente;
2. Decrescente;
3. Aleatório;

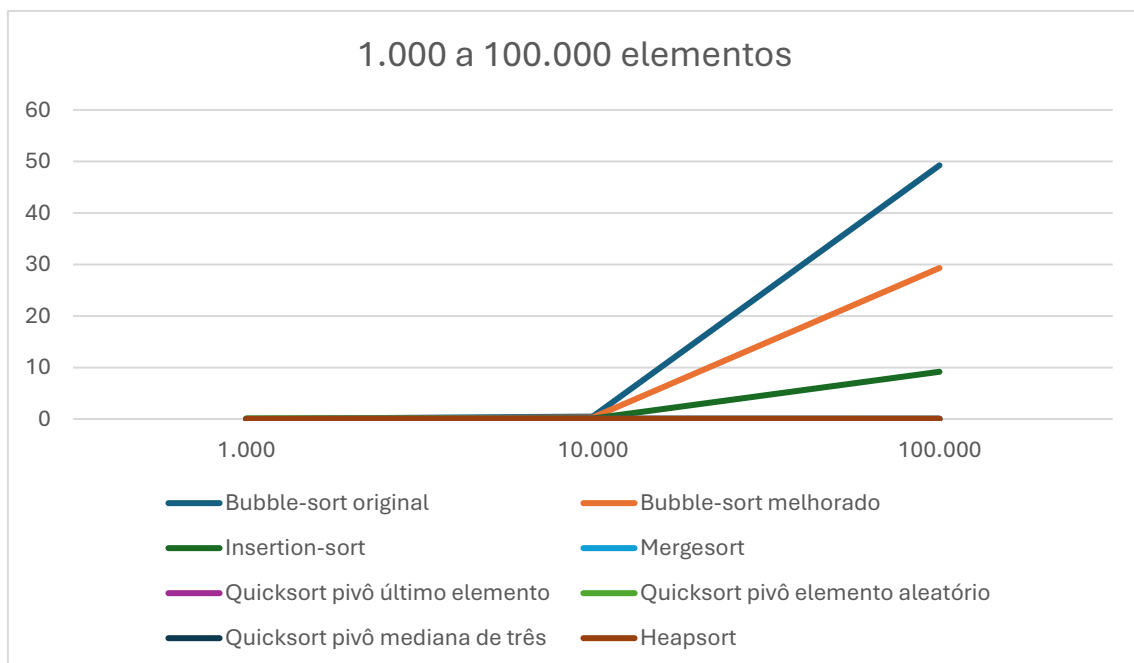
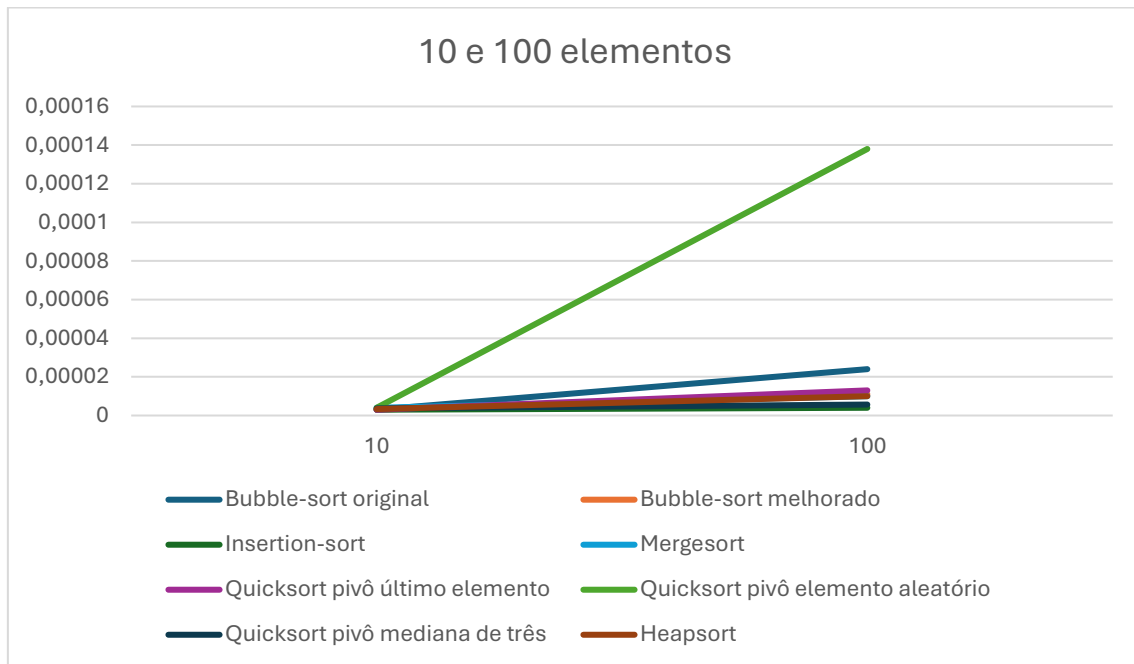
Onde em cada categoria citada, existirá de 10 até 50.000.000 elementos. E para cada grupo de elementos o tempo de execução de cada algoritmo é guardado, foram feitas três execuções para cada algoritmo e então, o cálculo da média aritmética entre os três tempos com a precisão de 9 casas decimais pois foi notado que em alguns casos três casas decimais não foram suficientes para visualização do resultado.

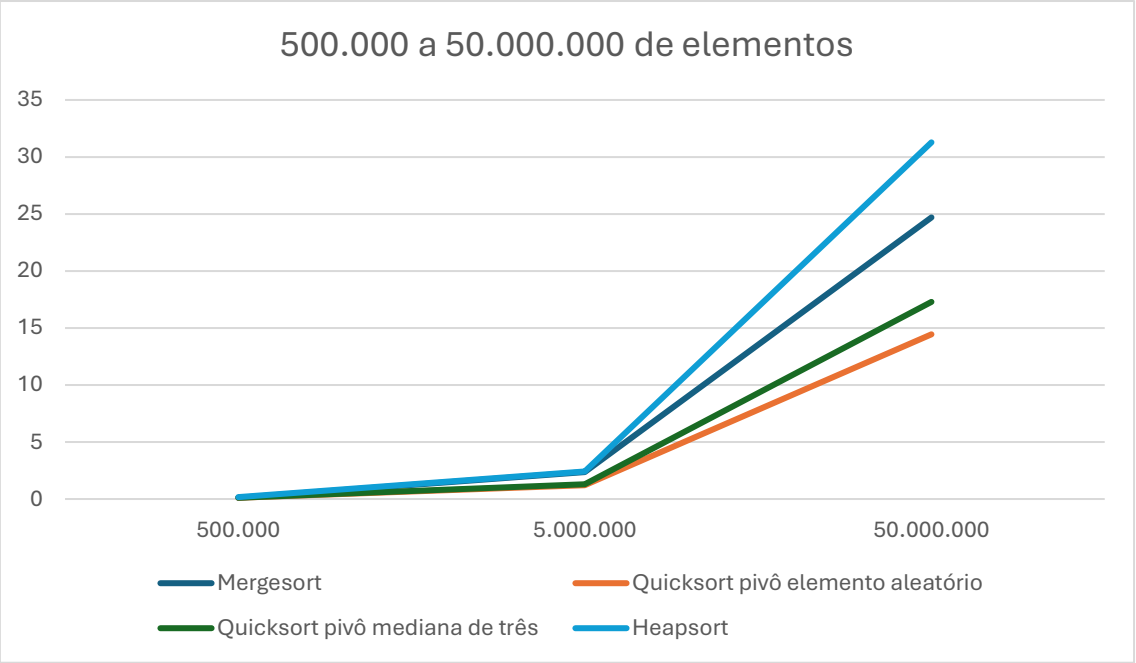
As informações referentes às configurações da máquina onde os algoritmos foram executados são:

- Sistema Operacional: Linux/Ubuntu;
- Arquitetura: 64 bits;
- Processador: Intel Core i3-7100 CPU @ 3.90GHz com 2 núcleos e cache de 64KB;
- Memória RAM: 8GB;

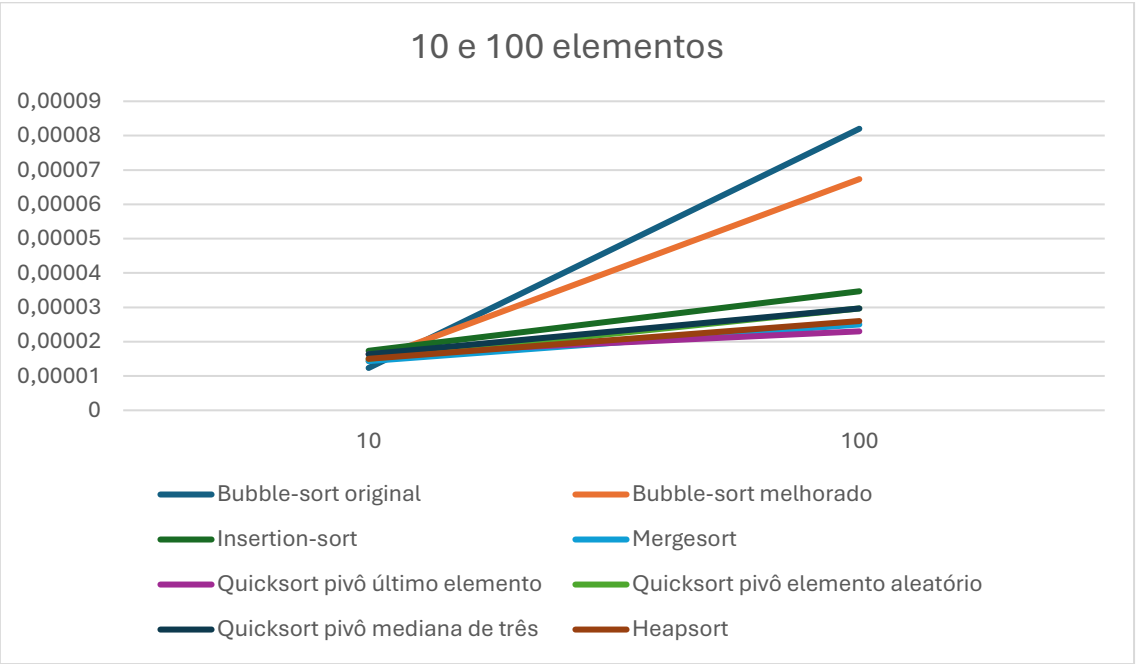
# Testes

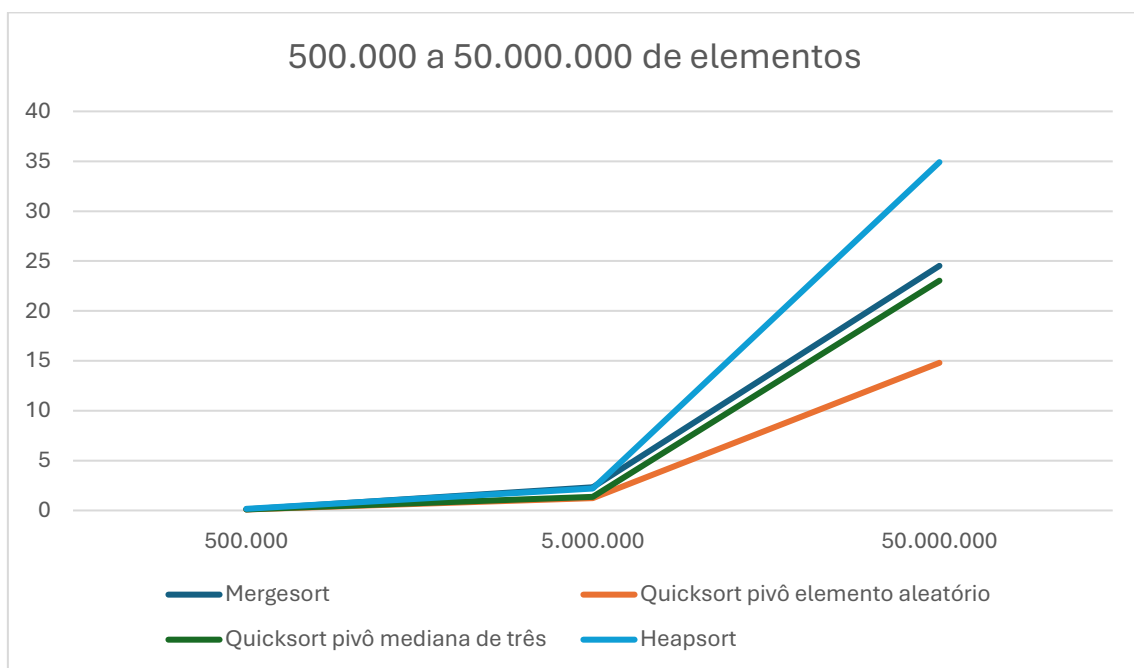
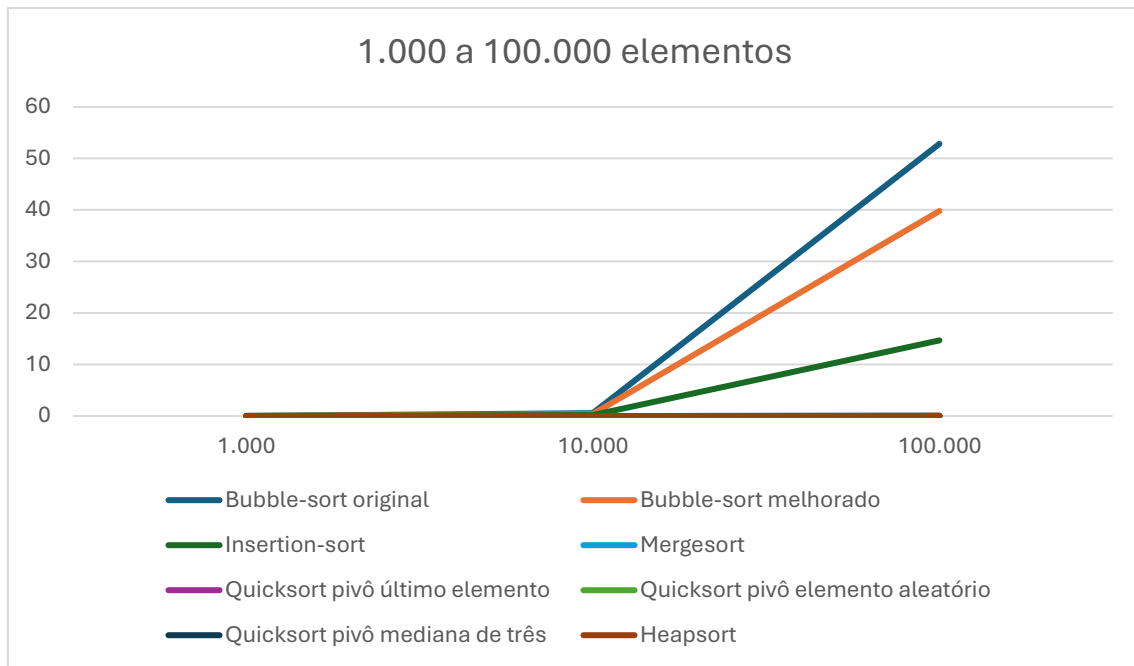
## Crescente



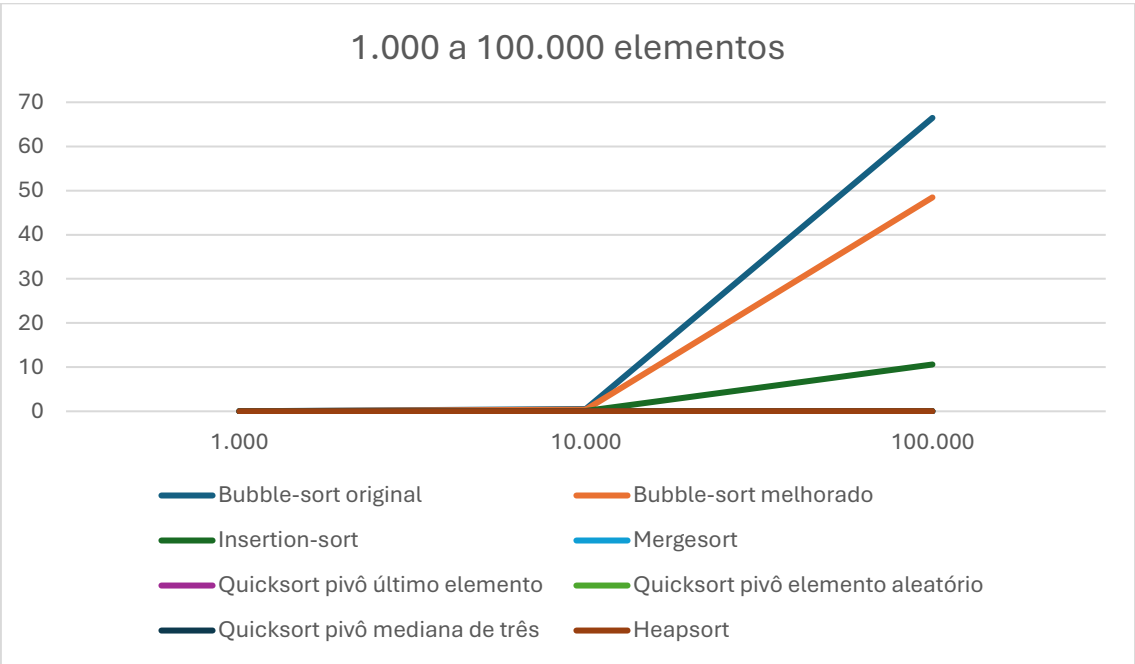
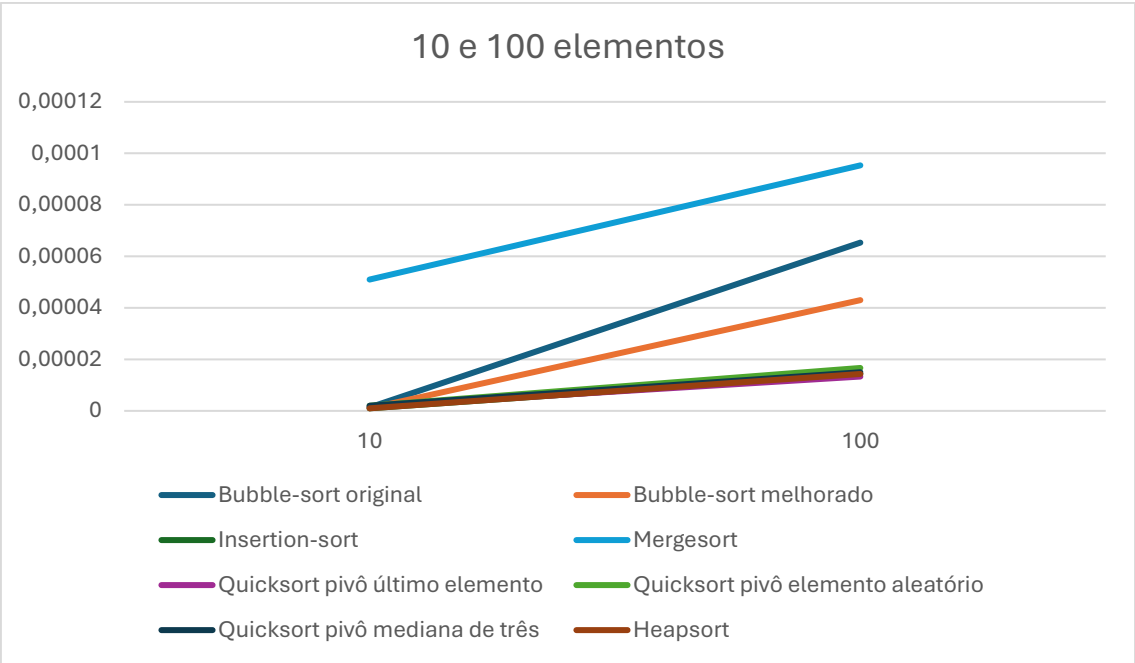


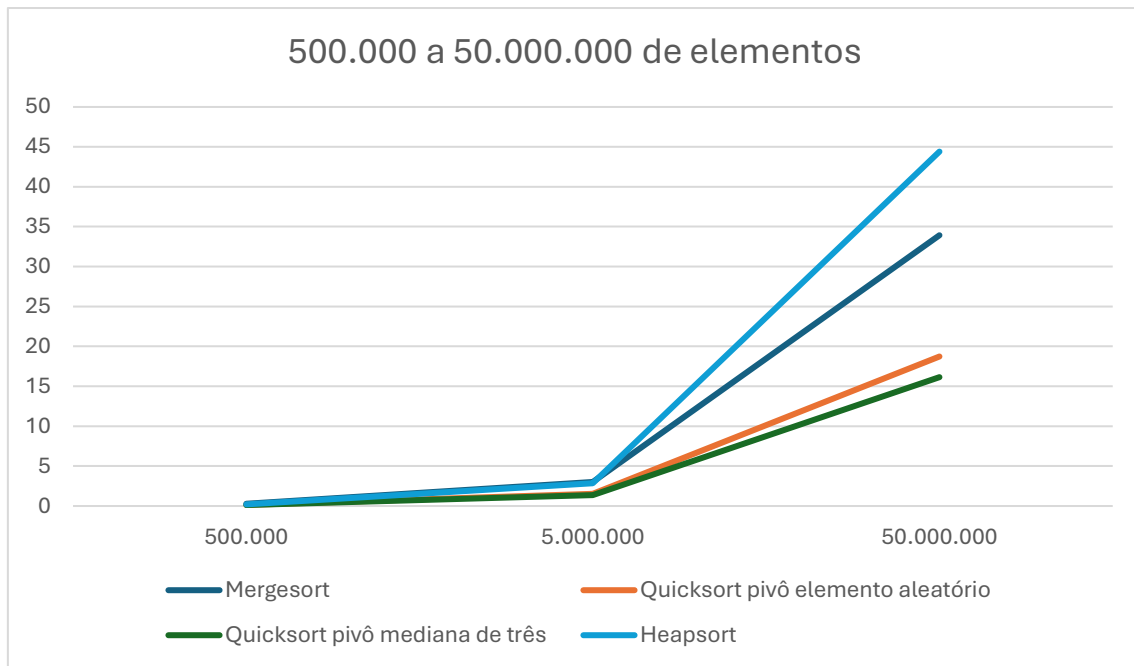
## Decrescente





# Aleatório





## Conclusão

Foi possível notar que em todos os testes os algoritmos de Quicksort se destacaram de maneira especial, mas um comportamento curioso foi identificado. Nos elementos crescentes e decrescentes, é visível que o algoritmo de Quicksort com o pivô aleatório foi mais eficaz que o Quicksort com a mediana de três, quando deveria ser ao contrário já que a mediana de três pega exatamente o elemento do meio, que é a melhor escolha de um elemento para ser pivô. Porém, esse resultado foi obtido porque no algoritmo Quicksort com a mediana de três é necessário descobrir qual é a mediana, e esse cálculo foi o que, possivelmente, atrasou a ordenação em relação ao Quicksort de elemento aleatório.

Apesar do Quicksort de elemento aleatório “brincar” com a sorte ao escolher um pivô aleatoriamente, foi o mais rápido entre todos os casos, onde é necessário um verdadeiro “azar” para conseguir o pior caso.

Sem dúvidas, o pior algoritmo de ordenação observado foi o Bubblesort e mesmo que sua versão melhorada seja mais rápida que a original, ainda perde drasticamente para todos os outros algoritmos de ordenação testados.