

INSTITUTO FEDERAL DE MINAS GERAIS
CAMPUS OURO BRANCO

Rodolfo Oliveira Miranda

TESTES DE ALGORITMOS DE ORDENAÇÃO

Ouro Branco, MG

2023

Rodolfo Oliveira Miranda

TESTES DE ALGORITMOS DE ORDENAÇÃO

Trabalho referente aos testes e comparações
sobre algoritmos de ordenação em Java.

Área de concentração: AEDs

Orientador: Luciana Lourdes Silva

Ouro Branco, MG

2023

COMPLEXIDADE

Bubble Sort:

O pior caso, caso médio e melhor caso têm uma complexidade de $O(n^2)$, tornando-o ineficiente para arrays com grandes quantidades de dados.

Selection Sort:

Também tem complexidade de $O(n^2)$ no pior, médio e melhor caso.

Insertion Sort:

Possui igualmente complexidade $O(n^2)$ no pior caso, porém pode ser mais eficiente em conjuntos parcialmente ordenados.

Merge Sort:

Tem complexidade $O(n \log n)$ em todos os casos, tornando-o mais eficiente em arrays com grandes quantidades de dados.

Quick Sort:

Em média, tem complexidade $O(n \log n)$, tornando-o eficiente para a maioria dos casos. No entanto, no pior caso, pode ser $O(n^2)$.

ESTABILIDADE

O Bubble, Selection, Insertion e Merge Sorts são estáveis, isso significa que a ordem relativa dos elementos iguais é preservada.

Apenas o Quick Sort não é classificado como estável, pois não é garantido a preservação de elementos iguais ao pivô.

USO DE MEMÓRIA

Bubble, Selection, Insertion e Quick Sorts são classificados “in-place”, o que significa que eles ordenam os elementos no próprio array de entrada sem alocar memória adicional.

Já o Merge Sort necessita a alocação adicional de memória para realizar a mesclagem, o que pode ser um problema em conjuntos de dados muito grandes.

IMPLEMENTAÇÃO

Bubble, Selection e Insertion Sorts são algoritmos mais simples de implementar e são úteis em situações em que a simplicidade é mais importante do que a eficiência.

Enquanto o Merge Sort e Quick Sort são mais complexos de implementar, mas oferecem melhor desempenho em geral.

MELHOR USO DE CASO

Bubble, Selection e Insertion Sorts são mais adequados para arrays pequenos ou quase ordenados.

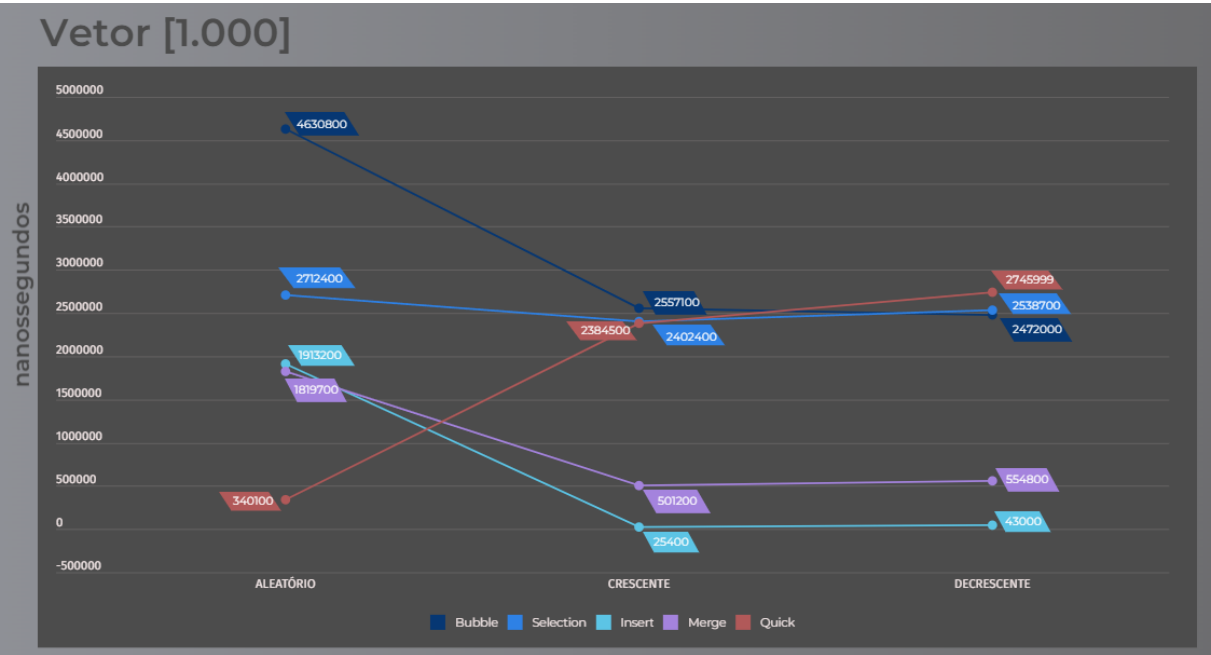
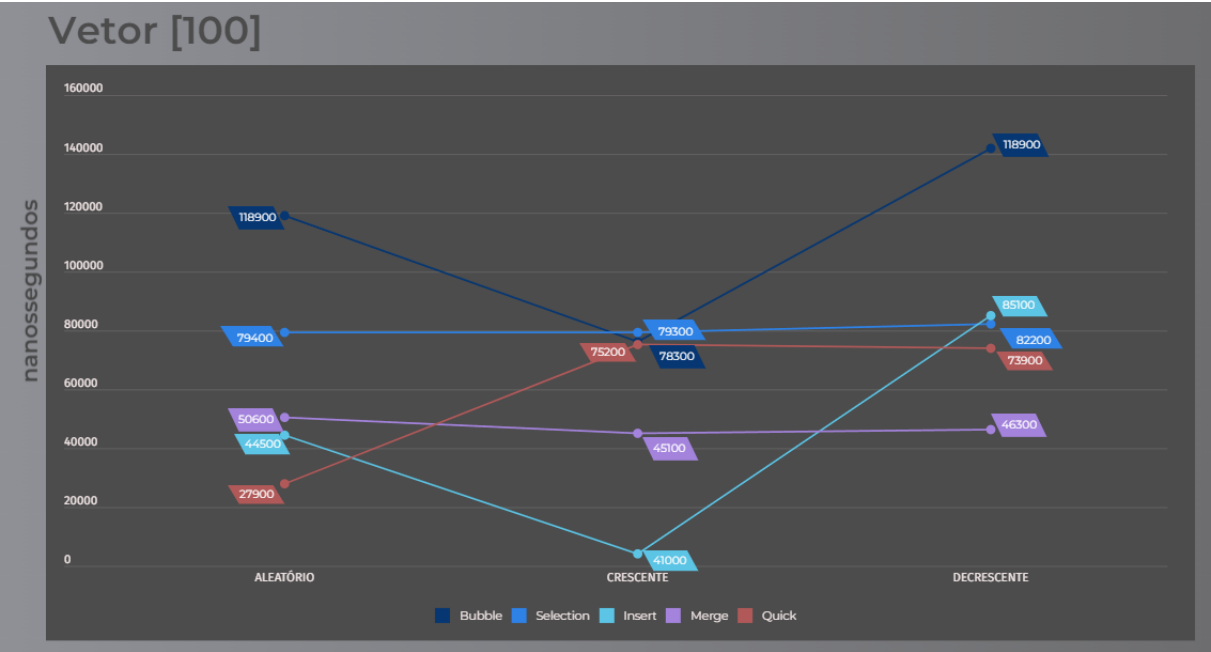
Merge Sort é mais eficaz em todos os tamanhos e é uma escolha sólida para ordenar grandes arrays.

Quick Sort é geralmente eficiente na média, mas pode degradar para $O(n^2)$ no pior caso.

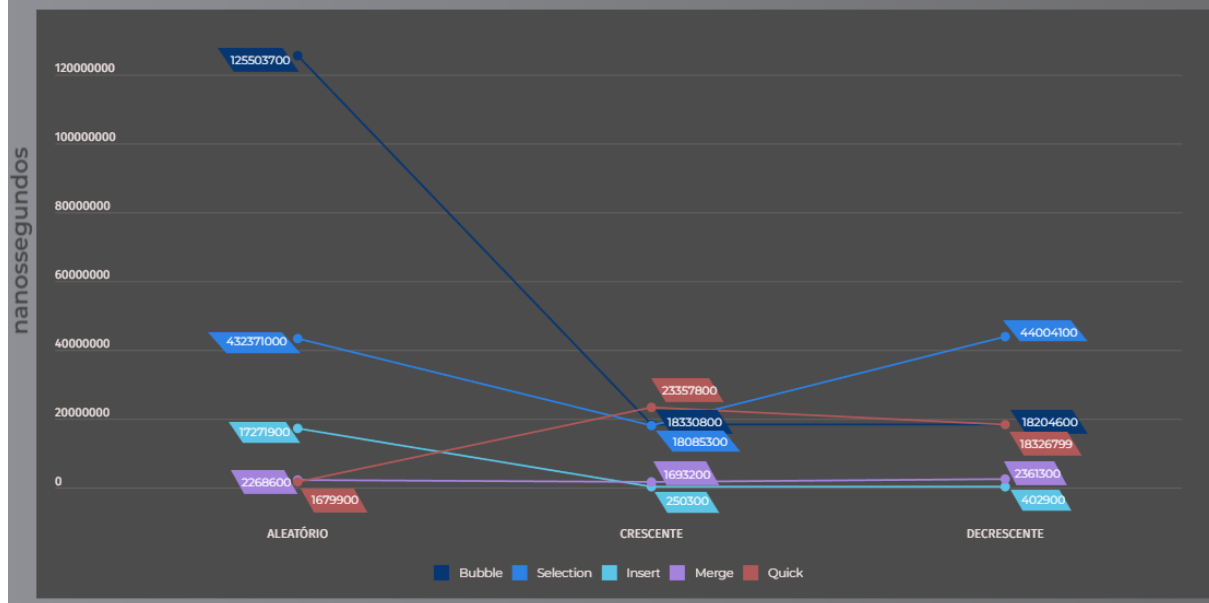
CONCLUSÃO

Após a análise dos dados obtidos e feita a comparação entre os algoritmos de ordenação, pode-se concluir que o Bubble Sort, Selection Sort e Insertion Sort são algoritmos simples de implementar, mas menos eficientes em termos de tempo em comparação ao Merge Sort e Quick Sort, que são mais eficientes de modo geral.

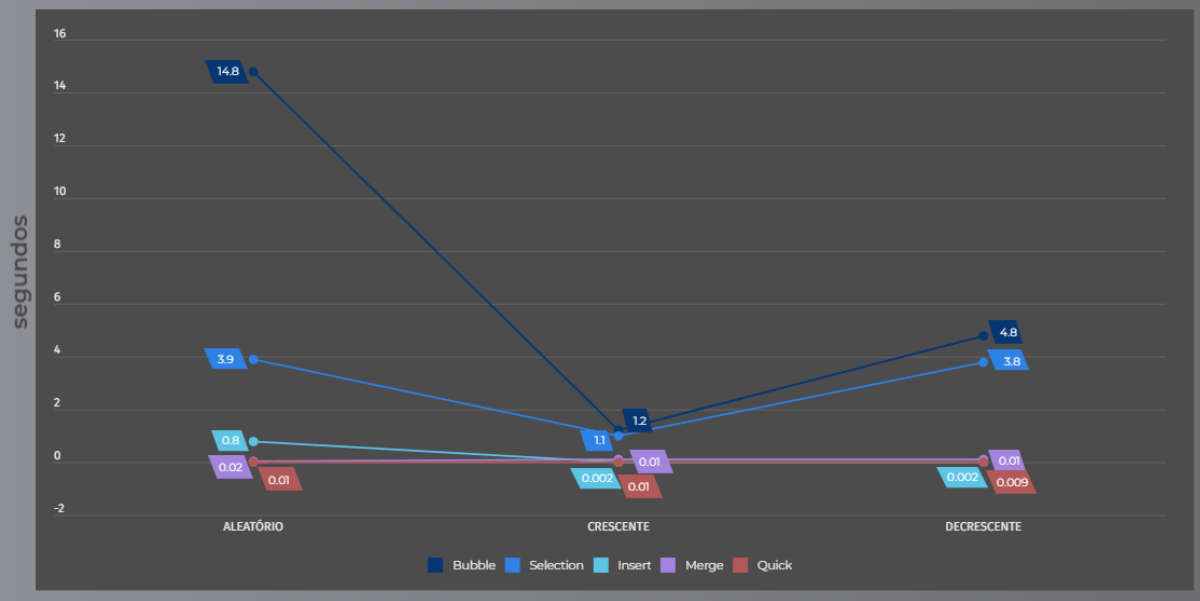
GRÁFICOS



Vetor [10.000]



Vetor [100.000]



Vetor [500.000]

