

**ENTREGAR PELO MOODLE ATÉ O FINAL DA AULA DE HOJE - 09/08**

A entrega é um arquivo Excel contendo as tabelas com o resultado dos testes e os gráficos de cada algoritmo.

O objetivo desse exercício é revisar alguns algoritmos de ordenação de arrays e analisar as suas complexidades de forma prática.

Para realizar o exercício você deve utilizar as classes BubbleSort, InsertionSort, MergeSort e QuickSort disponíveis no repositório da disciplina (em aula03).

Repositório [https://github.com/leonardoheredia/alect2-202302/tree/main/aula03\\_analise\\_algoritmos\\_ordenacao](https://github.com/leonardoheredia/alect2-202302/tree/main/aula03_analise_algoritmos_ordenacao)

Cada classe de ordenação realiza a ordenação do array e disponibiliza o número de operações realizadas, bem como o tempo de execução através de propriedades.

Por exemplo:

```
BubbleSort bubble = new BubbleSort();
bubble.ordenar(meuArrayBubble);
operacoes = bubble.getOperacoes();
```

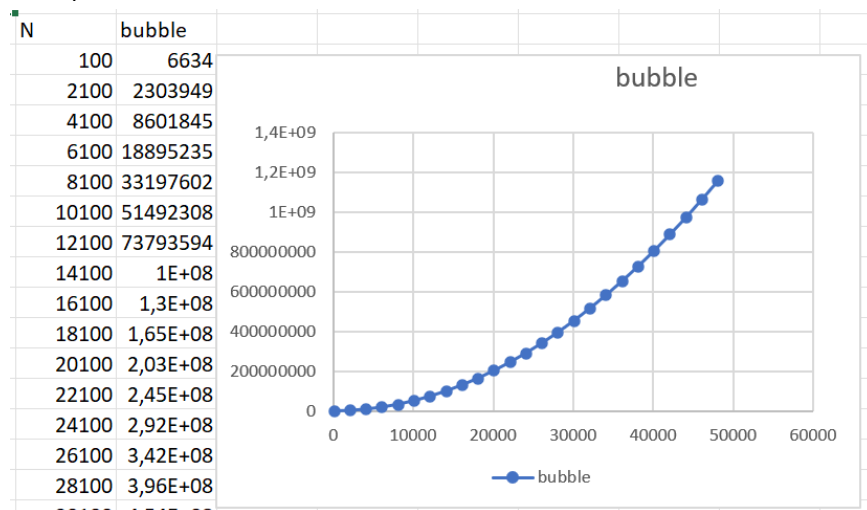
O número de operações realizadas para ordenar o array está disponível através do método getOperacoes().

A atividade consiste em gerar centenas de testes (de 100 até 50,000) com arrays de diferentes tamanhos e executar a ordenação do array usando os quatro algoritmos, coletando a quantidade de operações de cada execução.

Uma vez tendo a quantidade de operações basta colocar em um Excel e gerar um gráfico para identificar a classe de complexidade.

Para entregar a atividade colocar todos os gráficos dentro de um documento apenas.

Exemplo:



Tendo isso você deve preencher a tabela abaixo com a complexidade usando a notação O.

Algoritmo	Pior caso	Melhor caso
Bubble Sort		
Insertion Sort		
Merge Sort		
Quick Sort		