



Elaborar uma página web contendo um relatório de comparação de dez Algoritmos de Ordenação, gerada a partir da descrição abaixo.

Faça uma aplicação em Java que compare dez Algoritmos de Ordenação da seguinte forma:

**1. Elabore o menu com as seguintes opções:**

- a. Ordenar utilizando BubbleSort (melhorado);
- b. Ordenar utilizando SelectionSort;
- c. Ordenar utilizando InsertionSort;
- d. Ordenar utilizando MergeSort;
- e. Ordenar utilizando QuickSort;
- f. Ordenar utilizando \_\_\_\_\_;
- g. Ordenar utilizando \_\_\_\_\_;
- h. Ordenar utilizando \_\_\_\_\_;
- i. Ordenar utilizando \_\_\_\_\_;
- j. Ordenar utilizando \_\_\_\_\_;

**2. Os Algoritmos de Ordenação a serem utilizados nas opções f-j acima são de livre escolha.**

**3. Cada opção do menu deve ordenar (de forma crescente) três coleções de números inteiros (ArrayList ou vetor) da seguinte forma:**

- a. Ordenar uma coleção de números já ordenados em ordem crescente (melhor caso);
- b. Ordenar uma coleção de números aleatórios (caso médio);
- c. Ordenar uma coleção de números ordenados em ordem decrescente (pior caso);

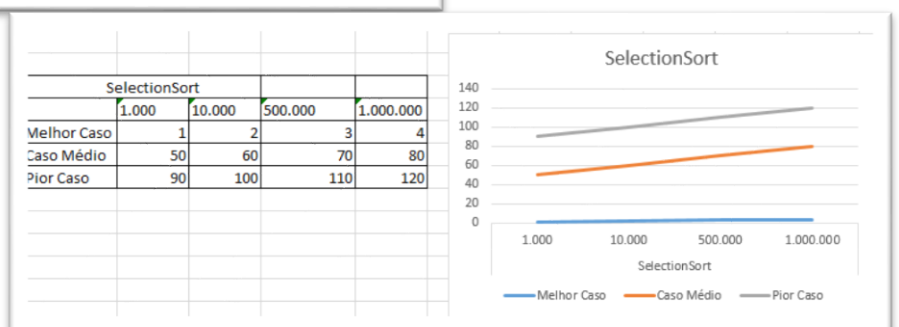
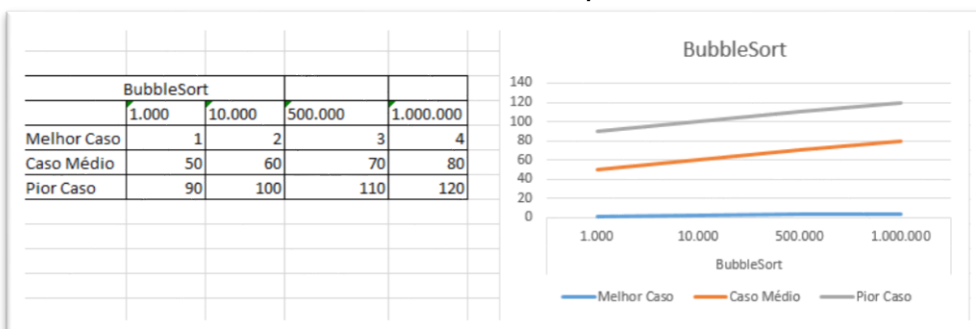
**4. Cada algoritmo deve ainda ser executado para coleções de tamanhos diferentes, respectivamente: 1.000, 10.000, 500.000, 1.000.000 posições, para cada um dos casos acima mencionados;**

5. Para cada execução (cada algoritmo para cada coleção de dados), medir o tempo necessário e registrar o valor (em segundos), de forma similar à tabela abaixo:

	BubbleSort						SelectionSort			
	1.000	10.000	500.000	1.000.000			1.000	10.000	500.000	1.000.000
Melhor Caso	x segundos	y segundos	z segundos	w segundos		Melhor Caso	x segundos	y segundos	z segundos	w segundos
Caso Médio	a segundos	b segundos	c segundos	...		Caso Médio	a segundos	b segundos	c segundos	...
Pior Caso	...	...	...	...		Pior Caso	...	...	...	...
	...outros algoritmos						...outros algoritmos			
	1.000	10.000	500.000	1.000.000		...	...	...	...	...
Melhor Caso	x segundos	y segundos	z segundos	w segundos		...	...	...	...	...
Caso Médio	a segundos	b segundos	c segundos	...		...	...	...	...	...
Pior Caso	...	...	...	...		...	...	...	...	...
		...					...			

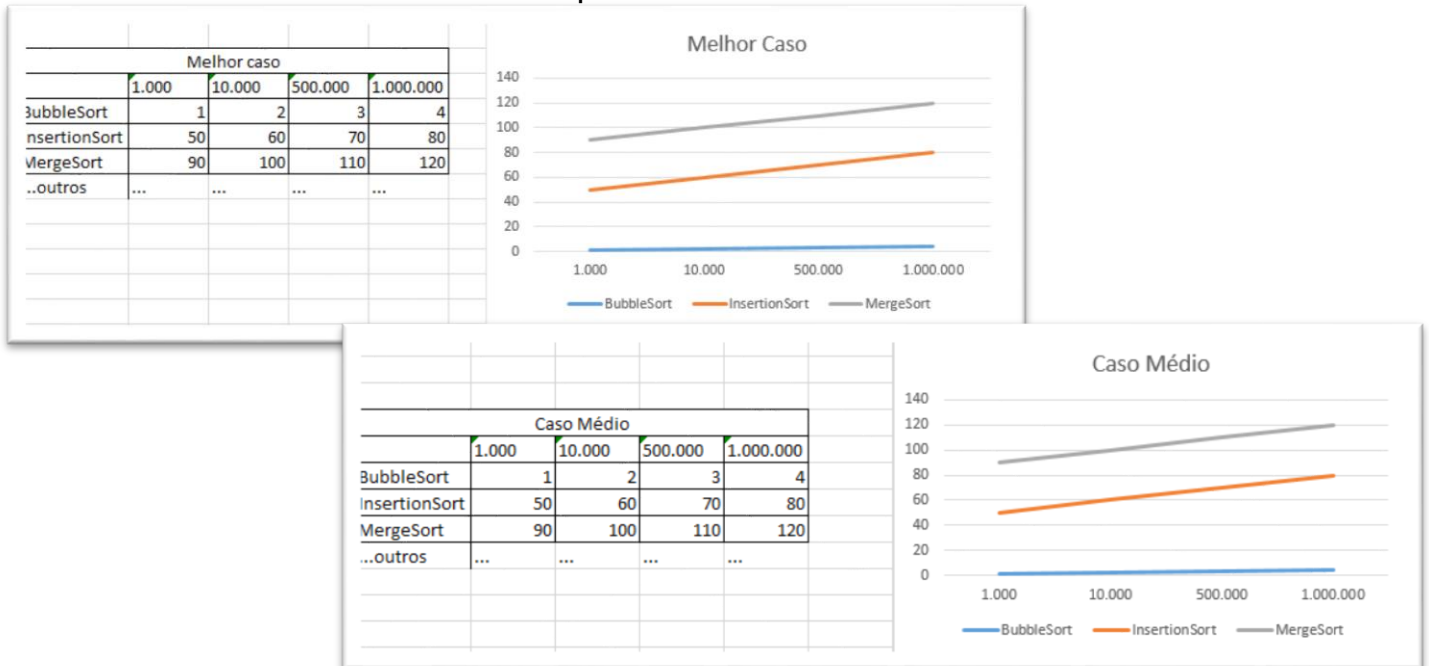
6. Ao final da tabulação de todos os resultados, **elaborar uma página web** (preferencialmente hospedada) contendo o relatório final das comparações. Esta página web deve apresentar seções para cada algoritmo, contendo:

- Título do Algoritmo, breve resumo, complexidade do algoritmo em cada caso, e código utilizado na linguagem Java;
  - Tempo necessário (em segundos) para as execuções de diferentes tamanhos em cada caso. Apresentar também, na página, a partir dos dados gerados:
    - tabelas com os tempos das execuções;
    - gráficos com os tempos das execuções, sendo:
      - um gráfico para cada algoritmo (tempo de execução por quantidade de números ordenados), apresentando três curvas (uma para cada caso (melhor caso, caso médio e pior caso));
- a. Exemplos:



2. três gráficos unificando todos os algoritmos, sendo um para cada caso (melhor caso, caso médio e pior caso);

a. Exemplos:



c. A forma de apresentação das informações na página web é de livre escolha, podendo cada algoritmo ser apresentado em uma página diferente, um modal, ou até mesmo na mesma página.

### Instruções de Envio:

Submeter, até o dia **06/12**, via moodle:

- tabelas com valores;
- códigos Java gerados;
- códigos web gerados (html, css, js e outros se houver);
- URL para acesso, caso hospedado;

Critérios para avaliação: 1,0 ponto por algoritmo + 1,0 extra para páginas hospedadas;

### Alguns códigos p/ auxílio:

- sortear número (ex: entre 0 e 9):

```
01. Random gerador = new Random();
02. int numero = gerador.nextInt(10);
```

- medir o tempo de uma determinada operação:

```
01. long t;  
02. t = System.nanoTime();  
03.     //operação 1  
04.     //operação 1  
05.     //operação 1  
06. t = System.nanoTime() - t;  
07. System.out.println("Tempo op1: " + t);  
08.  
09. t = System.nanoTime();  
10.     //operação 2  
11.     //operação 2  
12.     //operação 2  
13. t = System.nanoTime() - t;  
14. System.out.println("Tempo op2: " + t);
```

- conversão de nano segundos para segundos;

```
System.out.println("Tempo: " + t*Math.pow(10, -9) + " segundos.");
```