

4. O envio dos arquivos de solução é exclusivamente via classroom na data devidamente informada.
- Arquivos enviados por email o outro médio NÃO serão considerados e terá, por conseguinte, nota zero (0) na respectiva pergunta.
  - Envios fora da data informada terão uma **punição de 1 ponto por dia**.
  - O classroom permite fazer apenas um (1) único envio, então, assegure-se de carregar seu(s) arquivo(s) de resposta(s) antes de enviar pelo classrooom. Caso achar necessário, você pode usar qualquer compressor tipo winzip ou winrar e enviar uma pasta comprimida em formato .zip ou .rar.

### **Pergunta 1: Ranking de Algoritmos de Ordenação (10 pontos)**

Os algoritmos de ordenação são extremamente necessários em todo tipo de serviço software. Por esse motivo, você precisa ter diferentes implementações para usá-las quando for necessário. Visando esse objetivo, é solicitado que você implemente em Python um TAD que considere diferentes algoritmos de ordenação e que uma avaliação de desempenho com listas de diferentes tamanhos seja realizada.

A avaliação do desempenho dos algoritmos de ordenação é definida em função do tempo (quantidade de segundos/minutos) que um determinado algoritmo demora, na média, para ordenar uma lista de tamanho  $N$ . A média do desempenho é considerado da seguinte maneira: dado um algoritmo de ordenação  $X$ , devem ser geradas  $M$  listas de tamanho  $N$ , sendo que cada lista tem valores aleatórios. Com isso, haverá também  $M$  tempos de execução obtidos para cada uma das  $M$  execuções que ordenaram as  $M$  listas. A média do desempenho será, então, a soma total dos  $M$  tempos divididos por  $M$ .

Finalmente, deve-se criar automaticamente uma tabela que mostre, para cada algoritmo de ordenação, a média de execução em minutos obtido para  $M = 10$  listas de tamanho  $N$ , para diversos valores de  $N$ . Com isso, se teria uma tabela no seguinte formato:

	$N = 1000$	$N = 10000$	$N = 100000$	$N = 1000000$	$N = 10000000$
Algoritmo A					
Algoritmo B					
....					
Algoritmo Z					

Tenha em conta as seguintes restrições:

- Implementação seguindo um esquema de TAD que contenha todos os algoritmos de ordenação como métodos da classe (**1 ponto**)
- Geração dos resultados e armazenados em um arquivo externo em formato .XLS. (**1 ponto**)

# F a c u l d a d e

# IMPACTA

3. Implementação dos algoritmos bubble sort e mergesort. Obter resultados de desempenho médio para  $M = 10$  listas com valores aleatórios de tamanhos  $N = 1000$ ,  $N = 10000$ ,  $N = 100000$ ,  $N = 1000000$  e  $N = 10000000$ . **(3 pontos)**
4. Implementação do algoritmo insertion sort. Obter resultados de desempenho médio para  $M = 10$  listas com valores aleatórios de tamanhos  $N = 1000$ ,  $N = 10000$ ,  $N = 100000$ ,  $N = 1000000$  e  $N = 10000000$ . **(1 ponto)**
5. Implementação do algoritmo quicksort. Obter resultados de desempenho médio para  $M = 10$  listas com valores aleatórios de tamanhos  $N = 1000$ ,  $N = 10000$ ,  $N = 100000$ ,  $N = 1000000$  e  $N = 10000000$ . **(2 pontos)**
6. Implementação do algoritmo counting sort. Obter resultados de desempenho médio para  $M = 10$  listas com valores aleatórios de tamanhos  $N = 1000$ ,  $N = 10000$ ,  $N = 100000$ ,  $N = 1000000$  e  $N = 10000000$ . **(2 pontos)**

Qualquer assunto de implementação que não tenha sido considerado neste documento, você como desenvolvedor deverá realizar as suposições necessárias.