

## Orientações

- **Data de Entrega:** 07/11/2025 até as 23h59min
- Este trabalho deve ser elaborado EM TIMES DE NO MÁXIMO **03** ALUNOS
- Insira o seu Time no Canvas em Pessoas-> TAREFA1
- O Time deverá elaborar o que é pedido e entregar os arquivos via Canvas
- Este exercício deverá ser entregue para fins de arredondamento de média, se necessário.
- **Caso seja verificado que houve o uso de alguma IA para a resolução, o exercício não será considerado e a média será mantida.**
- Trabalhe somente com os comandos vistos em sala de aula.
- **É PROIBIDO o uso de comandos não trabalhados e, ou, discutidos com o professor**

## IDEIA:

**Análise Empírica de Dois Algoritmos de Ordenação** - Esta tarefa tem como principal objetivo a compreensão e análise de dois métodos de ordenação, vistos durante as aulas da disciplina de Estrutura de Dados.

Consiste na **simulação desses métodos** usando diferentes tipos de estruturas.

## TAREFA:

Os vetores deverão ser ordenados na **ordem DECRESCENTE**. O Time **DEVERÁ TRABALHAR** com as **alterações necessárias nos algoritmos apresentados em sala de aula. Só serão aceitos trabalhos que tenham adaptado os algoritmos dados em aula.**

Para cada método serão consideradas estruturas de tamanho  $5 \cdot 10^5$ , ou seja, 500.000 posições

Métodos a serem implementados:

- InsertionSort
- QuickSort

A ordenação será realizada num vetor de estruturas. A estrutura é constituída de 2 campos:

- Um valor inteiro - **será a chave da ordenação, maior ou igual a 100.000**
- Um valor real com valor superior a 100.000

Os valores, para preenchimento do vetor, deverão ser gerados de duas diferentes formas:

- Para cada forma serão gerados e rodados **20 vetores diferentes**, e o seu tempo de execução obtido. Esse tempo será usado para realizar uma análise Estatística.

Para obtenção do tempo utilize a biblioteca TIME.h. Inicie a contagem do tempo no começo da execução da função de ordenação e finalize antes de sair da mesma

**1. Aleatoriamente – chave e número real**

Como os casos têm que ser os mesmos para todos os métodos será necessário **trabalhar com o conceito de sementes** para a geração de números aleatórios. Os dois campos da estrutura serão gerados aleatoriamente

**2. Vetor em Ordem Contrária, ou seja, CRESCENTE**

**Chave em ORDEM CRESCENTE**, gere um número aleatório entre 100.000 e 300.000, a partir daí acrescente 100 a cada nova posição

**Número real ALEATORIAMENTE**

## **ATENÇÃO!!!!**

Para que a **análise seja válida**, todos os métodos têm que ser testados para o mesmo conjunto de dados e na mesma máquina

**Entregas:**

- O Time deverá entregar **DOIS** arquivos:

- Relatório - Arquivo com extensão pdf**, contendo:

- Capa com Nome e RA dos Alunos em Ordem Alfabética**

- Apresentação dos Métodos de Ordenação e Análise dos Resultados**

**PARA CADA MÉTODO O TIME DEVERÁ:**

- Apresentar o Algoritmo do Método com as alterações necessárias para a **Ordenação Decrescente**. É necessário também as sementes utilizadas para a geração dos dados. **Não escreva código de programa no relatório.**
- Apresentar os dados considerando 3 situações, para cada tipo (aleatório, decrescente)
  - Melhor Caso** – melhor tempo obtido pelo método para um determinado tamanho de vetor
  - Pior Caso** - pior tempo obtido pelo método para um determinado tamanho de vetor
  - Tempo Médio** – tempo médio obtido pelo método para um determinado tamanho de vetor

Para facilitar a apresentação trabalhe com as tabelas abaixo

<b>Método Insertion-Sort (Caso Aleatório)</b>	
<b>Pior Tempo de Execução:</b>	
<b>Melhor Tempo de Execução</b>	
<b>Tempo Médio de Execução</b>	
<b>Desvio Padrão do Tempo de Execução</b>	
<b>Mediana do Tempo de Execução</b>	

**Método Insertion-Sort (Caso Crescente)**

Pior Tempo de Execução:

Melhor Tempo de Execução

Tempo Médio de Execução

Desvio Padrão do Tempo de Execução

Mediana do Tempo de Execução

**Método Quick-Sort (Caso Aleatório)**Pivô =  $X[LI]$ Pivô =  $X[LS]$ Pivô =  $X[MEIO]$ 

Pior Tempo de Execução:

Melhor Tempo de Execução

Tempo Médio de Execução

Desvio Padrão do Tempo de Execução

Mediana do Tempo de Execução

**Método Quick-Sort (Caso Crescente)**Pivô =  $X[LI]$ Pivô =  $X[LS]$ Pivô =  $X[MEIO]$ 

Pior Tempo de Execução:

Melhor Tempo de Execução

Tempo Médio de Execução

Desvio Padrão do Tempo de Execução

Mediana do Tempo de Execução

## Comparação entre os Métodos

Comparação Entre os Métodos (Caso Aleatório)		
	Insertion	Quick (melhor opção)
Pior Tempo de Execução:		
Melhor Tempo de Execução		
Tempo Médio de Execução		
Desvio Padrão do Tempo de Execução		
Mediana do Tempo de Execução		

Comparação Entre os Métodos (Caso Crescente)		
	Insertion	Quick (melhor opção)
Pior Tempo de Execução:		
Melhor Tempo de Execução		
Tempo Médio de Execução		
Desvio Padrão do Tempo de Execução		
Mediana do Tempo de Execução		

### 3. Conclusão

A partir das tabelas geradas o Time irá fazer uma análise dos métodos, considerando as estatísticas obtidas, bem como, uma análise comparativa entre os métodos avaliando, discutindo as características dos dados, se houver, que levam o método a um melhor desempenho

### 4. Referência Bibliográfica – Se forem de sites colocar a data de acesso

**ENTREGAR:** O arquivo contendo o **Código Fonte, extensão .c, compatível com o CodeBlocks**, com as devidas orientações para que eu possa testá-lo como por exemplo, os valores de semente usados para a geração dos dados aleatórios. O Relatório deve ser entregue em **.pdf**