

### **Trabalho 1 - Comparativo entre Métodos de Ordenação**

O objetivo deste trabalho é implementar e comparar o desempenho de dois algoritmos de ordenação e busca.

Implemente um algoritmo de cada grupo abaixo:

GRUPO A	Grupo B
Insertion Sort Selection Sort Bubble Sort	Merge Sort Quick Sort Shell Sort

Além dos algoritmos de ordenação, implemente os seguintes métodos de busca:

- Busca Linear
- Busca Binária

#### **1 Metodologia e Requisitos:**

Utilize vetores de inteiros preenchidos aleatoriamente com valores variando de 0 a 99.999. Os vetores devem ter os seguintes tamanhos: 50, 500, 5.000, 50.000 e 500.000 elementos.

**Para cada método de ordenação, avalie as seguintes métricas:**

- Tempo de execução
- Número de trocas
- Número de iterações (soma de todos os passos/loops executados pelo algoritmo)

**Para cada método de busca, avalie as seguintes métricas:**

- Tempo de execução
- Posição encontrada (index)
- Obs: Utilize apenas o melhor método de ordenação para a análise de busca. Compare buscas apenas entre vetores desordenados e ordenados, desconsiderando o tempo de ordenação e focando somente no tempo de busca do elemento.

## **2 Protocolo Experimental:**

- Utilize o mesmo vetor em todos os métodos para garantir a consistência dos resultados e permitir uma comparação justa entre os diferentes algoritmos de ordenação.
- Execute a rotina de teste ao menos 10 vezes, utilizando vetores distintos a cada execução.
- Apresente a média e o desvio padrão das métricas obtidas.

**OBS: Este trabalho visa a implementação dos métodos de busca e ordenação. Não deve-se utilizar funções “prontas” (builtin) ou bibliotecas, pois desta maneira seria minimizado o emprego da lógica adequada.**

## **Perguntas de Pesquisa:**

P1) É possível elencar um método de ordenação que seja superior em todos os casos?

P2) É possível inferir alguma relação entre tamanho do vetor vs performance do algoritmo baseado em sua lógica de ordenação?

## **Relatório “Científico”:**

- Descrever de maneira sucinta os métodos implementados.
- Descrever o protocolo de teste
- Apresentar os resultados de maneira clara, utilizando tabelas, gráficos entre outras análises que julgar necessário.
- Discutir criticamente os resultados, elucidando ao leitor os *insights* obtidos. A discussão dos resultados deve responder as perguntas de pesquisa elencadas.

## **Avaliação e Considerações:**

- Grupos de 3 a 5 pessoas
- Implementação em C ou Python
- Código-Fonte (.c ou .py) compilável ou executável
- Relatório no formato de artigo (Padrão SBC)

## **Rúbricas de Avaliação:**

### **a) Implementação e Lógica de Programação (5 pts):**

- O código deve funcionar e atender todos os requisitos definidos neste trabalho.
- Os métodos de ordenação e busca devem estar implementados corretamente, utilizando a lógica adequada.
- As métricas devem ser computadas corretamente e apresentadas na tela.
- O código deve ser reproduzível a partir da mesma 'seed' utilizada?

### **b) Relatório (5 pts):**

- O relatório apresenta os elementos básicos (introdução, algoritmos, metodologia, discussão de resultados e conclusão) ?
  - Os métodos estão explicados corretamente, ainda que de maneira sucinta, conforme solicitado.
  - A metodologia descreve o protocolo experimental, permitindo a clarificação de como o algoritmo é testado, avaliado ? É possível apenas pela leitura, reimplementar as avaliações?
  - A discussão e exposição de resultados convence o leitor que as perguntas de pesquisa foram respondidas adequadamente ? Os resultados parecem adequados ?
-