

Trabalho Prático: Análise de Algoritmos de Ordenação

Tema: Comparação de Desempenho e Análise de Métodos de Ordenação

Objetivo:

Implementar e analisar o desempenho dos seguintes seis algoritmos de ordenação:

- Bubble Sort
- Selection Sort
- Insertion Sort
- Merge Sort
- Heap Sort
- Quick Sort

Cada algoritmo deve ser testado em três bases de dados diferentes para verificar o tempo de execução, o número de comparações e trocas realizadas, além de outros fatores relevantes, como o uso de memória (se aplicável).

Descrição do Trabalho:

Implementação:

Desenvolva em linguagem C os seis métodos de ordenação listados acima. Cada método deve ser implementado em uma função separada para facilitar a comparação.

Bases de Dados:

Utilizar três conjuntos de dados com características distintas:

- **Base 1:** Um vetor pequeno (100 elementos) com números gerados aleatoriamente.
- **Base 2:** Um vetor médio (1000 elementos) em ordem decrescente.
- **Base 3:** Um vetor grande (10.000 elementos) com 50% dos elementos já ordenados.

Métricas de Comparação:

Para cada conjunto de dados, deve ser registrada:

- Tempo de execução (em milissegundos)
- Número de comparações
- Número de trocas
- Memória utilizada (se possível)

Relatório:

Cada aluno (ou grupo) deve entregar um relatório contendo:

- A implementação dos algoritmos.

- Tabelas comparativas com as métricas para cada conjunto de dados.
- Gráficos de comparação para cada métrica (ex.: tempo de execução vs. tamanho do vetor).
- Discussão dos resultados, indicando em quais casos cada algoritmo se saiu melhor e por quê.
- Análise das vantagens e desvantagens de cada método, com base nos resultados obtidos.

Critérios de Avaliação:

Critério	Peso (%)
Correção	30%
Completeness	20%
Apresentação	20%
Desempenho	20%
Discussão	10%

- **Correção:** Algoritmos implementados corretamente e resultados conforme esperado.
- **Completeness:** Análise para todos os conjuntos de dados e algoritmos.
- **Apresentação:** Relatório bem formatado com gráficos e discussões detalhadas.
- **Desempenho:** Código eficiente e uso adequado de recursos computacionais.
- **Discussão:** Análise crítica dos resultados, com embasamento teórico.

Datas Importantes:

- **Data de Apresentação do Trabalho:** 11/10/2024 (Durante a aula prática)
- **Data Final para Entrega:** 31/10/2024 (até as 23h59 via e-mail ou plataforma online)

Observações:

- Trabalho individual.
- Todos os gráficos e análises devem ser gerados com base nos resultados obtidos nos testes. Códigos sem análise não serão considerados completos.
- Use ferramentas como `clock()` ou bibliotecas de temporização para medir o tempo de execução.