

Atividade Pr�tica Unidade II

An lise Comparativa de Algoritmos de Ordena  o

1. Entreg veis

- Relat rio T cnico (PDF)
 - Padr o de nomenclatura: <NOME>_<MATR CULA>.PDF
- Reposit rio no Github com c digos fonte

2. Formata  o do relat rio t cnico

- Introdu  o:
 - Objetivos do trabalho.
 - Breve contextualiza  o sobre a import ncia dos algoritmos de ordena  o.
- Fundamenta  o Te rica:
 - Descri  o sucinta de cada algoritmo (Bubble Sort, Insertion Sort, Selection Sort, Merge Sort, Quick Sort).
 - Complexidades te ricas (melhor caso, pior caso, caso m dio, espa o).
- Metodologia:
 - Descri  o dos tipos de dados gerados, precisando passar por conjuntos de dados dos tipos (aleat rios, quase ordenados, inversamente ordenados).
 - Tamanhos dos vetores testados.
 - M tricas coletadas (tempo, compara  es, trocas) e m todo de medi  o.
- Resultados e An lise:
 - Tabelas comparativas com dados brutos.
 - Gr ficos (tempo \times tamanho, compara  es \times algoritmo, etc.).
 - Discuss o cr tica (correspond ncia entre teoria e pr tica, anomalias observadas).
- Conclus es:
 - Ranking de desempenho dos algoritmos.
 - Aplica  es recomendadas para cada cen rio.

3. Código-Fonte

- Implementação dos 5 algoritmos em linguagem de programação (C ou C++).
- Código para geração de dados e medição de métricas.
- Organização clara (comentários, nomeação de variáveis, etc.).

4. Visualizações

- Gráficos comparativos (formato PNG/PDF).
- Legibilidade (rótulos, cores, títulos explicativos).

5. Critérios de Avaliação:

- Implementação Correta (30%):
 - Funcionalidade dos algoritmos.
 - Coleta precisa de métricas.
- Análise Experimental (40%):
 - Rigor metodológico.
 - Profundidade da discussão (teoria × prática).
- Clareza e Organização (20%):
 - Estrutura do relatório.
 - Qualidade das visualizações.
- Extras (10%):
 - Otimizações criativas.
 - Análises adicionais (ex: comparação com `std::sort`).