

UNIVERSIDADE FEDERAL DE JATAÍ

Instituto de Ciências Exatas e Tecnológicas
Bacharelado em Ciência da Computação
Linguagens Formais e Autômatos

Comparação de Algoritmos de Ordenação

Implemente diferentes algoritmos de ordenação estudados em sala e compare o desempenho (tempo de execução) em vetores de tamanhos variados.

Parte 1 – Implementação

Implemente os seguintes algoritmos de ordenação:

- Bubble Sort
- Selection Sort
- Insertion Sort
- Shell Sort
- Merge Sort
- Quick Sort

Parte 2 – Testes

1. Gerar vetores de inteiros com valores aleatórios.
2. Testar cada algoritmo com vetores de diferentes tamanhos. Sugestão de escala (dependendo do poder de processamento disponível):
 - 1.000 elementos
 - 5.000 elementos
 - 10.000 elementos
 - 50.000 elementos
 - 100.000 elementos
 - (se o computador suportar, testar também 500.000 ou 1.000.000 elementos para ver a diferença nos algoritmos mais ineficientes).

Parte 3 – Medição de Tempo

- Medir o tempo de execução de cada algoritmo em cada tamanho de vetor.

UNIVERSIDADE FEDERAL DE JATAÍ

Instituto de Ciências Exatas e Tecnológicas
Bacharelado em Ciência da Computação
Linguagens Formais e Autômatos

Comparação de Algoritmos de Ordenação

- Em C, podem usar a biblioteca **<time.h>** com **clock()** ou **<sys/time.h>** no Linux para obter maior precisão.
- Repetir cada execução algumas vezes (ex.: 5 vezes) e calcular a **média** para maior confiabilidade.

Parte 4 – Apresentação dos Resultados

1. Criar uma tabela comparativa com os tempos médios.
2. Se quiser, pode também plotar gráficos (linha ou barras) mostrando a relação entre tamanho do vetor *versus* tempo de execução para visualizar melhor as diferenças de desempenho.