



UNIVERSIDADE FEDERAL DO PIAUÍ  
CENTRO DE CIÊNCIAS DA NATUREZA  
DEPARTAMENTO DE COMPUTAÇÃO  
DISCIPLINA: PROJETO E ANÁLISE DE ALGORITMOS (2023-1)  
PROF: Guilherme Avelino

### Trabalho Prático de Projeto e Análise de Algoritmos

#### PARTE I

#### Implementação e Comparação de Algoritmos de Ordenação

A análise assintótica de algoritmos é o tipo de análise mais utilizado em ciência da computação. Contudo, ela pode não deixar claro as diferenças de desempenho de algoritmos que possuem a mesma função de complexidade. Dessa forma, a análise experimental funciona como um critério de desempate. Esse tipo de análise consiste em executar os algoritmos para diversas entradas e ao final comparar seus tempos de processamento.

O objetivo dessa primeira parte do trabalho é realizar a implementação e comparação de desempenho dos principais algoritmos de ordenação, são eles:

- Ordenação pelo método da bolha (*bubblesort*);
- Ordenação por inserção (*insertionsort*);
- Ordenação por intercalação (*mergesort*);
- Ordenação por Heap (*heapsort*);
- Ordenação rápida (*quicksort*).

**\*OBS1: os algoritmos devem ser implementados seguindo o mais próximo possível os algoritmos descritos em sala de aula, os quais têm como fonte principal o livro do Cormen. Implementações que não seguirem esses algoritmos serão descartadas.**

Todos os algoritmos devem ser testados em três condições:

- Vetor em ordem crescente;
- Vetor em ordem decrescente;
- Vetor aleatório.

Para cada uma das possibilidades acima serão feitos testes com vetores dos seguintes comprimentos:

- 100;
- 1.000;
- 5.000;
- 30.000;
- 50.000;
- 100.000;
- 150.000;
- 200.000.

Para cada tamanho e condição especificado acima o algoritmo deve ser executado, no mínimo, três vezes. O **tempo** de execução de um algoritmo e a **quantidade de comparações** para um determinado tipo de entrada e determinado tamanho será a média das três execuções.

**\*OBS2: não realizar o teste dos algoritmos em paralelo. Avaliar cada algoritmo em uma dada condição e tamanho de entrada por vez.**

## PARTE II

### Proposta de Algoritmo de Ordenação Híbrido

Com os resultados da primeira parte em mãos, vocês verificarão que dependendo do tamanho e disposição dos elementos do vetor de entrada uns algoritmos funcionam melhor do que outros. O objetivo da segunda parte do trabalho é criar um algoritmo de ordenação “híbrido” (AOH). Este AOH terá como algoritmo principal o Mergesort, Heapsort ou QuickSort e algoritmo secundário será escolhido a partir dos testes realizados.

**Pode ainda ser escolhido um algoritmo de ordenação em tempo linear para compor o AOH, desde que devidamente justificado e garantido as condições necessárias para seu uso.**

O trabalho consiste em escolher quais algoritmos usar para criar o AOH e quando o processamento passará de um para o outro. Para o AOH algoritmos de ordenação de tempo linear podem ser utilizados se julgar que ele é aplicável dada as entradas possíveis.

No relatório o grupo **deve justificar as escolhas** acima e mostrar o desempenho do AOH em formato de tabelas e gráficos

#### Apresentação dos Resultados (Partes I e II)

Os algoritmos podem ser implementados em qualquer linguagem de programação. Porém, no relatório devem ser especificadas as condições que foram utilizadas na execução dos testes, ou seja, devem ser especificadas:

- a. Linguagem de Programação e Compilador;
- b. Sistema Operacional;
- c. Processador;
- d. Capacidade e velocidade da memória RAM;

Os resultados devem ser apresentados em formato de gráfico relacionando o tamanho da entrada e:

- a. Tempo de execução;
- b. Quantidade de comparações (condições de comandos de decisão e laços de repetição avaliadas);

Dessa forma, para cada algoritmo devem ser apresentados dois gráficos para cada um dos quesitos de avaliação citados acima.

**\*OBS3: alternativamente, gráficos unindo os resultados de todos os algoritmos (ou separado em grupos de comportamento similares) p/ entradas grandes pode ser útil p/ permitir a comparação dos resultados entre algoritmos.**

O relatório do trabalho deve ser feito em formato de artigo (modelo SBC) e deve ter no máximo 15 páginas de conteúdo. Para cada um dos algoritmos de ordenação deve ser feita uma explanação das principais características do algoritmo (estratégia, complexidade, ...) e apresentação dos resultados na forma de gráfico e tabela. São obrigatórios:

- a. Título;
- b. Autores;
- c. Resumo;
- d. Introdução;
- e. Metodologia;

f. Resultados

g. Conclusão: deve ser feita uma análise dos resultados esperados com os resultados obtidos;

h. Bibliografia;

### **CRONOGRAMA**

- **04/07/2023:** Entrega da implementação dos algoritmos de ordenação (PARTE I). Algoritmos deve ser implementados separadamente (arquivos), porém deve ter um programa que permita testar a execução de todos eles. Essa execução de teste deve contemplar as 3 condições descritas anteriormente, porém apenas para vetores de tamanho pequeno (500 é o suficiente).
- **11/07/2023:** Entrega final. Código da implementação da Parte II, entrega do relatório final e apresentação dos resultados.