

TRABALHO PRÁTICO: ALGORITMOS DE ORDENAÇÃO

OBJETIVO: Implementar algoritmos de ordenação, mensurar seus tempos de execução e analisar o comportamento observado em função de sua complexidade assintótica. Com base nas observações realizadas, projetar e implementar uma variante híbrida de um algoritmo de ordenação, de forma a combinar as características de algoritmos que apresentaram um desempenho satisfatório para problemas de pequena escala com os algoritmos que apresentaram melhor desempenho para problemas de grande escala. Assim sendo, são pedidas as seguintes tarefas:

1. Implementação

Implemente os seguintes algoritmos de ordenação:

- *Bubblesort*;
- *Insertion sort*;
- *Merge sort*;
- *Quicksort*.

2. Geração de Dados de Entrada

Gere vetores aleatórios de tamanhos variados (n), por exemplo: $n = [10, 10^2, 10^3, 10^4, 10^5, 10^6, 10^7, 10^8]$. Certifique-se de criar vetores com números inteiros aleatórios.

3. Mensuração dos Tempos de Execução

Meça os tempos médios de execução de cada algoritmo para os diferentes tamanhos de entrada. Devem ser realizadas ao menos 10 aferições do tempo de execução de cada algoritmo, para cada caso de teste.

4. Análise dos Resultados

Plote gráficos comparativos dos tempos de execução observados em função de n para cada algoritmo. Comente se o comportamento prático dos algoritmos está de acordo com sua análise assintótica teórica. Além disso, deseja-se responder à seguinte pergunta: existe algum n para o qual algum dos algoritmos *Bubblesort* e *Insertion sort*, notadamente com complexidade assintótica $O(n^2)$, sejam executados mais rapidamente que os algoritmos *Merge sort* e *Quicksort*, com complexidade assintótica $O(n \log n)$?

5. Motivação e Especificação de um Algoritmo Híbrido

Considerando a análise dos resultados obtidos na etapa anterior, selecione um dos dois algoritmos: *Bubblesort* ou *Insertion sort*. A ideia é combinar o algoritmo selecionado com o *Merge sort* ou o *Quicksort*, de forma a obter um desempenho prático ainda melhor. Assim, deve-se modificar o *Merge sort* ou o *Quicksort* para usar o *Bubblesort* ou *Insertion sort* em subvetores de tamanho menor ou igual a k , onde k é um parâmetro a ser definido nas experimentações (os alunos devem fazer experimentos para indicar qual o valor ideal do parâmetro, de forma a maximizar o desempenho do algoritmo híbrido).

6. Comparação de desempenho

Compare o tempo médio de execução do algoritmo híbrido com os tempos dos algoritmos puros para os mesmos tamanhos de entrada.



UNIVERSIDADE FEDERAL DO CEARÁ - UFC
CAMPUS DE CRATEÚS
CURSO: CIÊNCIA DA COMPUTAÇÃO
DISCIPLINA: PROJETO E ANÁLISE DE ALGORITMOS
PROFESSOR: RAFAEL MARTINS BARROS

INSTRUÇÕES:

- O trabalho deve ser realizado em grupos de até 4 alunos. Cada grupo será capitaneado por um aluno a ser escolhido pelo professor da disciplina, na aula do dia 09/12/2024;
 - Os demais integrantes de cada grupo serão definidos durante a aula do dia 09/12/2024;
- Os arquivos de implementação e relatório, o qual deve incluir a descrição da implementação de cada algoritmo, gráficos comparativos e análise de resultados, devem ser enviados via SIGAA, até a data limite estipulada no sistema;
- Os alunos devem montar uma apresentação do trabalho realizado, a qual deverá ocorrer no dia 06/01/2025;
- A implementação e apresentação do trabalho valerão parte da nota da avaliação da Unidade 1. Mais especificamente, o trabalho vale 3,0 (três) pontos da primeira avaliação.