

A empresa DataLogic, que atua na área de inteligência de dados, está avaliando diferentes abordagens para processar grandes volumes de informações coletadas em tempo real.

O time de desenvolvimento observou que o desempenho dos algoritmos de ordenação varia conforme o tamanho dos dados e a estratégia utilizada (iterativa ou recursiva).

Para apoiar essa análise, você foi designado para realizar uma pesquisa prática, testando e comparando o comportamento de diferentes algoritmos de ordenação e de funções recursivas.

Essa pesquisa experimental servirá de base para a implementação do seu código final.

Parte 1 — Pesquisa Prática (4,0 pontos)

1. Objetivo da pesquisa:

Investigar e comparar o desempenho e o funcionamento prático de algoritmos de ordenação recursivos e iterativos, por meio de experimentos simples realizados pelo próprio aluno.

2. Etapas da pesquisa:

a) Escolha um algoritmo recursivo (Merge Sort ou Quick Sort) e um algoritmo iterativo (Bubble Sort, Selection Sort ou Insertion Sort).

b) Pesquise o funcionamento teórico de cada um (em livros, artigos ou sites confiáveis) e descreva brevemente as etapas de cada algoritmo (máximo 10 linhas por algoritmo).

Qual algoritmo se mostrou mais rápido?

Há diferença perceptível conforme o tamanho do vetor?

Como a recursividade influencia o desempenho?

3. Entrega da pesquisa:

Elabore um pequeno relatório (de 1 a 2 páginas) com:

Introdução sobre o propósito da pesquisa;

Descrição dos algoritmos estudados;

Resultados dos testes realizados;

Análise comparativa dos resultados;

Conclusões e referências utilizadas.

Essa pesquisa deverá guiar o desenvolvimento do código final: os algoritmos estudados e testados deverão ser implementados na segunda parte da atividade.

Parte 2 — Desenvolvimento de Código (6,0 pontos)

Com base na pesquisa prática realizada:

1. Implemente um programa que contenha:

a) Uma função recursiva que calcule o fatorial de um número inteiro positivo;

b) A implementação recursiva do algoritmo estudado na pesquisa (Merge Sort ou Quick Sort);

c) A implementação iterativa do segundo algoritmo escolhido (Bubble Sort, Selection Sort ou Insertion Sort).

d) Realize testes práticos utilizando vetores de tamanhos diferentes (ex.: 10, 100 e 500 elementos).

e) Registre os tempos médios de execução (aproximados) e comente o comportamento observado:

2. Gere um vetor de 20 números inteiros aleatórios e realize os seguintes testes:

Exiba o vetor original;

Ordene o vetor com cada algoritmo;

Mostre os resultados e o tempo de execução aproximado de cada método.

3. Inclua, nos comentários do código ou em um arquivo separado:

Breve explicação sobre o funcionamento da recursividade no seu código;

Comparação entre os métodos de ordenação com base na sua pesquisa prática.

Entrega:

Arquivo ".pdf" contendo o relatório da pesquisa prática, e
Arquivo ".c" ocontendo o código-fonte desenvolvido.