Relatório de Merge Sort

integrantes : Felipe Rodrigues Ribeiro DRE:119052031

Igor Torres Torres da Costa DRE:119034669

O código desenvolvido implementa um algoritmo completo de ordenação externa para arquivos CSV muito grandes, que não cabem integralmente na memória principal. A ordenação externa é uma técnica essencial em sistemas que lidam com grandes volumes de dados, pois permite ordenar arquivos maiores do que a capacidade de RAM disponível, dividindo-os em partes menores chamadas de "runs". O processo inicia com a função que divide o arquivo original em pequenos blocos, cada um limitado por um tamanho máximo de linhas, garantindo que cada run possa ser carregado em memória e ordenado com segurança. Após serem ordenados individualmente, esses blocos são salvos em arquivos temporários.

Em seguida, ocorre a fase de mesclagem, onde todos os runs temporários são combinados em um único arquivo final, já completamente ordenado. Para realizar isso de forma eficiente, o código utiliza um min-heap (fila de prioridade) que sempre mantém no topo o menor (ou maior, se a ordenação for decrescente) item entre os runs abertos, garantindo que a fusão aconteça de maneira gradual e ordenada.

A lógica de comparação é controlada por uma classe chamada HeapItem, que encapsula os registros lidos e permite controlar se a ordenação será crescente ou decrescente, sem depender de truques inseguros com inversão de sinal. Além disso, existe uma função auxiliar que tenta converter as chaves de ordenação para número, possibilitando ordenações numéricas ou alfabéticas, dependendo do tipo de dado contido na coluna escolhida.

O sistema também possui uma parte dedicada à geração de arquivos CSV sintéticos para teste. Um gerador automático cria arquivos com milhares de linhas, contendo colunas como identificador, nome aleatório, CPF fictício e um valor numérico aleatório, simulando um cenário realista para avaliar a ordenação externa.

Ao final da execução, o código limpa os arquivos temporários e remove os arquivos de teste criados, mantendo o ambiente organizado. A clareza das mensagens impressas durante o processo permite acompanhar todas as etapas: divisão em runs, quantidade de runs gerados, início e conclusão da fusão e confirmação do sucesso. Assim, o código se mostra robusto, bem estruturado, modular e totalmente preparado para ser adaptado ou reutilizado em sistemas maiores que exigem ordenação de grandes volumes de dados de forma eficiente.