TRABALHO DE VENDAS – ALGORITMO DE CONSULTA COM BUSCA ÓTIMA

ALUNO: THIAGO MALHEIROS PORCINO

Universidade Federal Fluminense Programa de Pós-graduação em Ciência da Computação TIC10002 – Estruturas de Dados e Algoritmos (2017.1) T1 – 26/04/2017

Trabalho de Implementação

Os dados de venda da equipe de vendas de uma empresa estão armazenados em um arquivo txt que obedece ao seguinte formato: filial, ano_mês, cod_vendedor, total_vendido.

Faça um programa que carregue os dados do arquivo em estruturas de dados e que permita responder expressões como:

- 1) total de vendas das filiais com códigos entre 10 e 20
- 2) total de vendas das filiais com códigos entre 10 e 20 nos meses de Jan/17 até Jun/17
- 3) total de vendas de todas as filiais nos meses de Ago/17 até Out/17

As estruturas de dados devem permitir acesso eficiente para todos os tipos de expressões e a conjunção de condições sobre filiais e datas devem ser realizadas por meio de operação de interseção de conjuntos. Cada resumo de venda (linha) do arquivo deve dar origem a somente uma instância (objeto) no programa. Utilize as estruturas de dados mais eficientes para cada tipo de pergunta.

Neste trabalho, foram desenvolvidas 2 soluções:

1) **Solução 1:** Utilização de Hash Bidimensional composto por Estruturas de Listas Simplesmente encadeadas.

Simplesmente encadeadas.

Nesta solução o objetivo era realizar a busca pelos elementos de venda em O(1) em uma tabela Hash bidimensional composta por elementos (vendas) em 3 listas simplesmente

encadeada, onde uma delas era uma matriz de listas.

Vantagens: Caso houvesse poucas colisões, a busca é a mais rápida possível em O(1)

uma vez que o acesso é constante.

Desvantagem: Caso houvessem muitas colisões, a busca iria sofrer retardo, uma vez que utilizando lista o programa seria obrigado a percorrer a lista por inteiro para achar

determinado elemento.

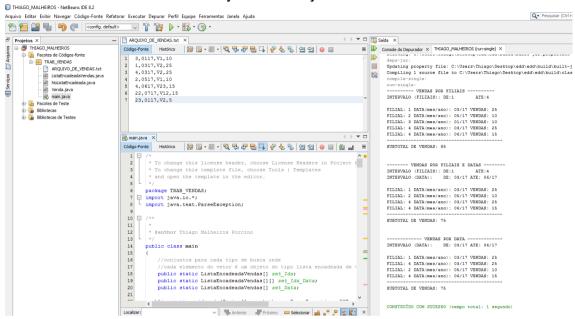
2) **Solução 2:** Utilização de Hash Bidimensional composto por estruturas de Árvores AVLs.

Nesta solução o objetivo era buscar os elementos da forma mais otimizada garantindo que essa busca continuasse eficiente caso houvessem muitas colisões. Utilizei uma tabela Hash bidimensional composta por elementos (vendas) em duas árvores.

Vantagens: Busca eficiente até mesmo quando ocorrerem muitas colisões.

Desvantagem: A busca não é O(1) mesmo com poucos elementos e poucas colisões.

EXECUÇÃO DA SOLUÇÃO 1



EXECUÇÃO DA SOLUÇÃO 2

