

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.

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

THIAGO_MALHEIROS - NetBeans IDE 8.2

Arquivo Editar Exibir Navegar Código-Fonte Refatorar Executar Depurar Perfil Equipe Ferramentas Janela Ajuda

Projetos X

- THIAGO_MALHEIROS
 - Pacotes de Códigos-fonte
 - TRAB_VENDAS
 - ARQUIVO_DE_VENDAS.txt
 - ListaEncadeadaVendas.java
 - NolistaEncadeada.java
 - Venda.java
 - main.java
 - Pacotes de Teste
 - Bibliotecas
 - Bibliotecas de Testes

Código-Fonte

```
1 3,0117,V1,10
2 1,0317,V2,25
3 4,0317,V2,25
4 2,0517,V1,10
5 4,0617,V23,15
6 22,0717,V12,15
7 23,0117,V2,5
```

main.java

```
1 /*
2  * To change this license header, choose License Headers in Project
3  * To change this template file, choose Tools | Templates
4  * and open the template in the editor.
5  */
6 package TRAB_VENDAS;
7 import java.io.*;
8 import java.text.ParseException;
9
10 /**
11  *
12  * @author Thiago Malheiros Porcino
13  */
14 public class main
15 {
16     //conjuntos para cada tipo de busca onde
17     //cada elemento do vetor é um objeto do tipo lista encadeada de
18     public static ListaEncadeadaVendas[] set_ids;
19     public static ListaEncadeadaVendas[] set_ids_data;
20     public static ListaEncadeadaVendas[] set_data;
21 }
```

Console do Depurador X THIAGO_MALHEIROS (run-single) X

```
Depurador
Updating property file: C:\Users\Thiago\Desktop\edd\edd\build\build-3
Compiling 1 source file to C:\Users\Thiago\Desktop\edd\edd\build\classes
compile-single:
run-single:
----- VENDAS POR FILIAIS -----
INTERVALO (FILIAIS): DE:1 ATE:4
FILIAL: 1 DATA(mes/ano): 03/17 VENDAS: 25
FILIAL: 2 DATA(mes/ano): 05/17 VENDAS: 10
FILIAL: 3 DATA(mes/ano): 01/17 VENDAS: 10
FILIAL: 4 DATA(mes/ano): 03/17 VENDAS: 25
FILIAL: 4 DATA(mes/ano): 06/17 VENDAS: 15
SUBTOTAL DE VENDAS: 85
----- VENDAS POR FILIAIS E DATAS -----
INTERVALO (FILIAIS): DE:1 ATE:4
INTERVALO (DATA): DE: 03/17 ATE: 06/17
FILIAL: 1 DATA(mes/ano): 03/17 VENDAS: 25
FILIAL: 2 DATA(mes/ano): 05/17 VENDAS: 10
FILIAL: 4 DATA(mes/ano): 03/17 VENDAS: 25
FILIAL: 4 DATA(mes/ano): 06/17 VENDAS: 15
SUBTOTAL DE VENDAS: 75
----- VENDAS POR DATA -----
INTERVALO (DATA): DE: 03/17 ATE: 06/17
FILIAL: 1 DATA(mes/ano): 03/17 VENDAS: 25
FILIAL: 4 DATA(mes/ano): 03/17 VENDAS: 25
FILIAL: 2 DATA(mes/ano): 05/17 VENDAS: 10
FILIAL: 4 DATA(mes/ano): 06/17 VENDAS: 15
SUBTOTAL DE VENDAS: 75
CONSTRUÍDO COM SUCESSO (tempo total: 1 segundo)
```

EXECUÇÃO DA SOLUÇÃO 2

THIAGO_MALHEIROS - NetBeans IDE 8.2

Arquivo Editar Exibir Navegar Código-Fonte Refatorar Executar Depurar Perfil Equipe Ferramentas Janela Ajuda

Projetos X

- THIAGO_MALHEIROS
 - Pacotes de Códigos-fonte
 - TRAB_VENDAS
 - ARQUIVO_DE_VENDAS.txt
 - ArvoreAVL.java
 - ListaEncadeadaVendas.java
 - NolistaEncadeada.java
 - Venda.java
 - main.java
 - Pacotes de Teste
 - Bibliotecas
 - Bibliotecas de Testes

Código-Fonte

```
1 3,0117,V1,10
2 1,0317,V2,25
3 4,0317,V2,25
4 2,0517,V1,10
5 4,0617,V23,15
6 22,0717,V12,15
7 23,0117,V2,5
8
```

ListaEncadeadaVendas.java

```
1 /*
2  * and open the template in the editor.
3  */
4 package TRAB_VENDAS;
5 import java.io.*;
6 import java.text.ParseException;
7
8 /**
9  *
10  * @author Thiago Malheiros Porcino
11  */
12
13
14 public class main
15 {
16     public static ArvoreAVL t_filial = new ArvoreAVL();
17     public static ArvoreAVL t_data = new ArvoreAVL();
18     public static ListaEncadeadaVendas[] hash_linha_coluna;
19
20     public static void main(String[] args) throws ParseException, IOException {
21 }
```

Console do Depurador X THIAGO_MALHEIROS (run-single) X

```
Depurador
Updating property file: C:\Users\Thiago\Desktop\edd\edd\build\build-3
Compiling 1 source file to C:\Users\Thiago\Desktop\edd\edd\build\classes
compile-single:
run-single:
----- VENDAS POR FILIAIS -----
INTERVALO (FILIAIS): DE:1 ATE:4
FILIAL: 1 DATA(mes/ano): 03/17 VENDAS: 25
FILIAL: 2 DATA(mes/ano): 05/17 VENDAS: 10
FILIAL: 3 DATA(mes/ano): 01/17 VENDAS: 10
FILIAL: 4 DATA(mes/ano): 03/17 VENDAS: 25
FILIAL: 4 DATA(mes/ano): 06/17 VENDAS: 15
SUBTOTAL DE VENDAS: 85
----- VENDAS POR FILIAIS E DATAS -----
INTERVALO (FILIAIS): DE:1 ATE:4
INTERVALO (DATA): DE: 03/17 ATE: 06/17
FILIAL: 1 DATA(mes/ano): 03/17 VENDAS: 25
FILIAL: 2 DATA(mes/ano): 05/17 VENDAS: 10
FILIAL: 4 DATA(mes/ano): 03/17 VENDAS: 25
FILIAL: 4 DATA(mes/ano): 06/17 VENDAS: 15
SUBTOTAL DE VENDAS: 75
----- VENDAS POR DATA -----
INTERVALO (DATA): DE: 03/17 ATE: 06/17
FILIAL: 1 DATA(mes/ano): 03/17 VENDAS: 25
FILIAL: 4 DATA(mes/ano): 03/17 VENDAS: 25
FILIAL: 2 DATA(mes/ano): 05/17 VENDAS: 10
FILIAL: 4 DATA(mes/ano): 06/17 VENDAS: 15
SUBTOTAL DE VENDAS: 75
CONSTRUÍDO COM SUCESSO (tempo total: 1 segundo)
```