



Universidade Federal de Viçosa

Universidade Federal de Viçosa - Campus Florestal

Isabella Menezes Ramos 3474

Lucas Barros 3511

Vinícius Júlio 3495

Algoritmos e Estruturas de Dados I

Implementação do TAD Matriz Esparsa

Florestal - MG

24 de Setembro de 2019

Isabella Menezes Ramos 3474

Lucas Barros 3511

Vinicius Júlio 3495

Implementação do TAD Matriz Esparsa

1º trabalho prático de Algoritmos e Estruturas de Dados 1. Trabalho que visa a concretizar conceitos de Tipos Abstratos de Dados e de Listas encadeadas através da aplicação de matriz esparsa.

Professora: Prof^a. Thais R. M. Braga Silva

Disciplina: Algoritmos e Estruturas de Dados 1

Florestal - MG

24 de Setembro de 2019

Sumário

- 1 – Introdução**
- 2 – Estratégia e organização**
- 3 – Desenvolvimento**
- 4 – Conclusão**
- 5 – Referências Bibliográficas**

1 Introdução

O trabalho consiste em concretizar os conceitos de Tipo Abstrato de Dados e Listas implementadas por encadeamento através da aplicação de Matriz Esparsas.

Matrizes esparsas são matrizes nas quais a maioria das posições é preenchida por zeros ou nulos. Para economizar um espaço significativo de memória, empregaremos alocação encadeada, utilizando lista, para justamente armazenar apenas os termos diferentes de zeros (ou nulos).

A partir disso, temos como objetivo criar uma matriz de Clientes x Produtos se comportando, respectivamente, como linha x coluna onde em cada célula da matriz teremos uma lista de compras com as quantidades e as datas de compras feita pelo Cliente X ao comprar o Produto Y.

2 Estratégia e Organização

Primeiramente nós dedicamos os primeiros dias para entender todas as exigências requeridas para o desenvolvimento do trabalho prático. Procuramos então entender todo o conceito por trás da matriz a partir de listas duplamente encadeadas e listas circulares, já que de certa forma, essa parte parecia ser a mais desafiadora durante o processo.

Dado todo o processo de entendimento das exigências, começamos a elaborar o sistema a partir da geração da matriz e a testando com valores dentro de cada célula como é dito na descrição do trabalho prático. Primeiramente nós desenvolvemos um TAD para a matriz, inicialmente seguindo a implementação apresentada no livro de Nivio Ziviani, criando a célula cabeça principal, e esta irá gerar uma lista de células cabeças para a direita (criando listas de colunas) e para baixo (listas de linhas). Essa mesma célula cabeça inicialmente possuía os apontadores Primeiro e Último para percorrer a lista, porém devido a se tratar de uma lista circular, não havia a necessidade de se ter estes ponteiros, já que a lista circular não possui início e fim, então fizemos somente o apontador principal que sempre terá o valor da célula cabeça e dessa forma já conseguiríamos percorrer toda a matriz. Após a discussão sobre os usos de apontadores seguimos as seguintes etapas.

1. Inicializar a Matriz de valores;
2. Elaboração do menu o main para testes;
3. Inserir células na matriz e imprimi-la para testes;
4. Criação da lista de Produtos e suas funções básicas;
5. Implementar a lista de Produtos substituindo o item valor na matriz;
6. Criar funções qtdCompraProduto e qtdCompraClientes;
7. Leitura por arquivo da matriz.

3 Desenvolvimento do Trabalho

Inicializar a Matriz de valores

Para inicializar a matriz criamos uma célula cabeça principal que usa um apontador “direita” para criar a lista de células cabeça das colunas e o apontador “abaixo” para criar a lista de células cabeça das linhas. Assim como especificado, a célula cabeça principal possui índice (-1, -1), as células cabeça de linha possuem índice -1 no campo linha e as de coluna possuem índice -1 no campo coluna.

Elaboração do menu no main para testes

Para facilitar a realização dos testes no programa, criamos um menu contendo uma lista de opções de operações a serem escolhidas pelo usuário, que inclui tanto a inserção de dados manualmente quanto por leitura de arquivo.

Inserir células na matriz e imprimi-la para testes

Para a inserção na matriz, inicialmente a função percorre a lista de colunas até a coluna desejada, após isso, percorre a lista da mesma em busca do índice desejado indicando o valor da linha, e então é feita a inserção da célula de acordo com a implementação do Ziviani. Após isso percorremos a lista de linhas até a linha desejada, em seguida percorremos a lista da mesma em busca do índice desejado para conectar a célula na lista.

A função para imprimir a matriz percorre a lista de colunas imprimindo os índices e listas de produtos das células. Se é identificado que uma célula possui o índice linha igual a -1 (célula cabeça da linha), a função passa a percorrer e imprimir a linha abaixo dela. O ciclo para quando é encontrada uma célula com índice -1 na coluna (célula cabeça da coluna).

Criação da lista de Produtos e suas funções básicas

A lista de produtos é uma lista encadeada cujas células possuem como valor os produtos, contendo as datas das compras e a quantidade de produtos comprados. Essa lista de Produtos é inserida nas células da matriz. Possui funções de inicialização, inserção de produtos e de imprimir a lista.

Implementar a lista de Produtos substituindo o item valor na matriz

Substituímos o TItem que possuía apenas um valor, assim como pedido nos itens a, b, c, por TProdutos, que possui os valores de data e quantidade de produto comprado. O TItem com valor único foi usado como teste durante a implementação da matriz e das listas.

Criar funções qtdCompraProduto e qtdCompraClientes

As funções qtdCompraporProduto e qtdCompraporClientes foram implementadas de forma semelhante.

Na qtdCompraporProduto acessamos a célula cabeça da coluna para procurar o produto e somar a lista de produtos, ou seja é somado o total de compras de um produto efetuado por todos os clientes. Na qtdCompraporClientes percorremos a linha do cliente, acessando a lista de produtos e contando quantas compras foram efetuadas no total.

Leitura por arquivo da matriz

A criação da matriz é feita pela leitura de um arquivo definido pela função LeMatriz, que recebe como parâmetro o ponteiro para arquivo, a lista como apontador da matriz, a quantidade de linhas e colunas da matriz que são lidas no main com a função de arquivo, e o TProdutos.

Inicialmente lemos os 2 valores da primeira linha e inicializamos as linhas e colunas com os números lidos, que representam o tamanho da matriz. A partir da segunda linha utilizamos dois while, um percorrendo até o final do arquivo linha por linha e pegando os 2 primeiros valores como cliente e produto, enquanto o restante é tratado em outro while, dentro do primeiro, que divide os valores entre data e produto até que se encontre uma quebra de linha, fazendo com que o primeiro while se repita.

4 Conclusão

Com a execução do trabalho, percebemos a importância, as vantagens e desvantagens do uso de listas encadeadas, e o quanto vantajoso é a implementação das listas encadeadas para as matrizes esparsas.

Como as matrizes esparsas possuem a maior parte de seus campos nulos, a implementação por meio de listas encadeadas permite uma grande economia de memória, evitando a necessidade de alocar espaços que ficariam vazios.

5 Referências Bibliográficas

1 - ZIVIANI, Nivio. Projeto de algoritmos: com implementação em Pascal e C. Cengage Learning, 2004.