

TRABALHO 3

SSC0600-2018

Matrizes Esparsas

Gabriell Tavares Luna - 10736400 **P0**

Vinícius Rodrigues Geraldo - 10438122 **P1**

Edson Andrade da Costa Filho - 10407051 **P2**

1.Introdução

O programa foi desenvolvido com a finalidade de implementação do manuseio de uma matriz esparsa, habilitando tarefas como: incluir ou excluir matriz, atribuir valor a um elemento, consultar valor de um elemento, consultar a soma dos valores dos elementos de uma linha ou coluna e calcular o determinante de uma matriz quadrada. A matriz foi estruturada em um **arranjo de listas encadeadas**, onde só são alocados valores diferentes de zero e uma estrutura auxiliar para controle.

Essa estrutura foi construída com base no capítulo sobre *Matrizes Esparsas*, presente na parte 2 do livro “C Completo e Total, 3ª Edição Revista e Atualizada”, do autor Herbert Schildt.

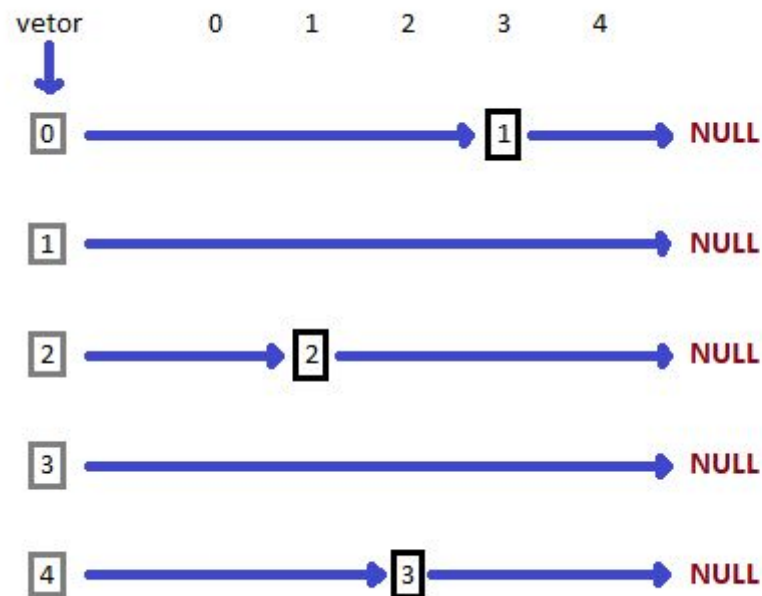
1.Implementação

Dada uma matriz esparsa $M(5 \times 5)$, podemos representar uma *matriz lógica e matriz física*. A *matriz lógica* é a matriz que o usuário pensa que existe no sistema.

Se a matriz tem dimensões de 5×5 , então a matriz lógica que suporta a matriz tem dimensões de 5×5 . Exemplo:

$$M = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 \end{bmatrix}$$

A *matriz física* é a matriz que realmente existe dentro do computador. Assim, a matriz ocupa espaço apenas para os elementos que estão em uso.



2. Descrição do Projeto

O projeto foi desenvolvido utilizando a IDE Code::Blocks 16.02 para o sistema operacional

Linux x64. O código foi compilado em Linux x64 utilizando o GCC 5.4.0 e em Windows x64 utilizando a versão 5.1.0 do GCC, sem problemas de compatibilidade. Os

headers do projeto são “stdio.h”, ”stdlib.h”, “malloc.h”, e o código fonte é “main.c”.

3. Tutorial

3.1 Compilando com o GCC:

Abra o terminal, ou Prompt de Comando, acesse a pasta onde está o código fonte “main.c”.

Para a compilação, execute: “gcc main.c -o Out”.

3.2 Executando:

- **Linux:**

Depois de compilado, execute o comando: “./Out”

- **Windows:**

Depois de compilado, execute o arquivo “Out.exe” que foi criado na pasta onde está o programa.

3.3 Primeiros passos:

Menu de opções:

```
Menu Principal
1. Criar uma matriz
2. Excluir uma matriz
3. Consulta do valor de uma posicao da matriz
4. Consulta da soma dos valores de uma linha da matriz
5. Consulta da soma dos valores de uma coluna da matriz
6. Atribuicao de um valor na posicao (i,j) da matriz
7. Calcular o determinante da matriz
8. Sair
```

1. Criar matriz.

Possibilita a criação de uma matriz “m” por “n”, onde “m” é quantidade de linhas e “n” a quantidade de colunas. Ele aloca um ponteiro chamado vetor, de tamanho “m”, que serve de auxiliar para a estrutura de lista encadeada. Para tal, foi criada a função `void Inicializa(matriz* m, int lin, int col)`.

```
Menu Principal
1. Criar uma matriz
2. Excluir uma matriz
3. Consulta do valor de uma posicao da matriz
4. Consulta da soma dos valores de uma linha da matriz
5. Consulta da soma dos valores de uma coluna da matriz
6. Atribuicao de um valor na posicao (i,j) da matriz
7. Calcular o determinante da matriz
8. Sair
<?>: 1

Modo: Criar Matriz
Entre com a quantidade de linhas: 100
Entre com a quantidade de colunas: 100
```

2.Excluir a matriz.

“Deleta” a matriz presente no programa, fazendo número de linhas e colunas ser zero e liberando a memória reservada para o vetor de linhas auxiliar. Para tal, foi criada a função `void Exclui(matriz* m)`.

```
Menu Principal
1. Criar uma matriz
2. Excluir uma matriz
3. Consulta do valor de uma posicao da matriz
4. Consulta da soma dos valores de uma linha da matriz
5. Consulta da soma dos valores de uma coluna da matriz
6. Atribuicao de um valor na posicao (i,j) da matriz
7. Calcular o determinante da matriz
8. Sair
<?>: 2

Modo: Excluir Matriz
Matriz excluida!
```

3.Consulta do valor de uma posição na matriz.

Consulta o elemento da matriz na linha e coluna desejada. As linhas variam de 0 a “m-1”, e as colunas variam de 0 a “n-1”. Para tal, foi criada a função `float ConsultaDado(matriz* m, int lin, int col)`.

```
Menu Principal
1. Criar uma matriz
2. Excluir uma matriz
3. Consulta do valor de uma posicao da matriz
4. Consulta da soma dos valores de uma linha da matriz
5. Consulta da soma dos valores de uma coluna da matriz
6. Atribuicao de um valor na posicao (i,j) da matriz
7. Calcular o determinante da matriz
8. Sair
<?>: 3

Modo: Consulta do Valor de uma posicao
Entre com a linha: 50
Entre com a coluna: 50
M[50][50]=0.00
```

4. Consulta da soma dos valor de uma linha da matriz.

Soma todos os valores de uma linha da matriz e lhe fornece o resultado da soma. Para tal, o usuário deve escolher a linha, que varia entre 0 e “m-1”. Para tal, foi criada a função `float SomaLinha(matriz* m, int lin)`.

```
Menu Principal
1. Criar uma matriz
2. Excluir uma matriz
3. Consulta do valor de uma posicao da matriz
4. Consulta da soma dos valores de uma linha da matriz
5. Consulta da soma dos valores de uma coluna da matriz
6. Atribuicao de um valor na posicao (i,j) da matriz
7. Calcular o determinante da matriz
8. Sair
<?>: 4

Modo: Consultar a soma das linhas da matriz
Entre com a linha: 50
Soma da linha 50 = 0.00
```

5. Consulta da soma dos valores de uma coluna da matriz.

Soma todos os valores de uma coluna da Matriz e lhe fornece o resultado da soma. Para tal, o usuário deve escolher a coluna, que varia entre 0 e “n-1”. Para tal, foi criada a função `float SomaColuna(matriz* m, int col)`.

```
Menu Principal
1. Criar uma matriz
2. Excluir uma matriz
3. Consulta do valor de uma posicao da matriz
4. Consulta da soma dos valores de uma linha da matriz
5. Consulta da soma dos valores de uma coluna da matriz
6. Atribuicao de um valor na posicao (i,j) da matriz
7. Calcular o determinante da matriz
8. Sair
<?>: 5

Modo: Consultar a soma das colunas da matriz
Entre com a coluna: 50
Soma da coluna 50 = 0.00
```

6. Atribuição de um valor na posição (i,j) da matriz.

Altera o valor de um elemento da matriz. O usuário de escolher a linha e coluna. As linhas variam de 0 a “m-1”, e as colunas variam de 0 a “n-1”. O valor inserido pode ser float. Para tal, foi criada a função `int MudaDado(matriz* m, int lin, int col, float val)`.

```
Menu Principal
1. Criar uma matriz
2. Excluir uma matriz
3. Consulta do valor de uma posicao da matriz
4. Consulta da soma dos valores de uma linha da matriz
5. Consulta da soma dos valores de uma coluna da matriz
6. Atribuicao de um valor na posicao (i,j) da matriz
7. Calcular o determinante da matriz
8. Sair
<?>: 6

Modo: Atribuir um valor na posicao (i,j) da matriz
Entre com a linha: 50
Entre com a coluna: 50
Entre com o valor: 1
M[50][50]=1.00
```

7. Calcular o determinante da Matriz

Se a matriz for quadrada, calcula seu determinante. Para tal cálculo, primeiro testa se existe alguma **fila nula** por meio da função `int LinhaNula(matriz* m)`, que implicaria em determinante nulo, se não, calcula o determinante utilizando o **Teorema de Laplace**, por meio da função `float Determinante(float *m, int ordem)`.

4. Outras informações

Determinante: Para calcular o determinante de uma matriz quadrada que não possui fila nula o programa utiliza uma função recursiva que implementa o Teorema de Laplace.

Teste do tempo para calcular o determinante em função da ordem da matriz:

O teste foi baseado no tempo para resolver o determinante da **matriz identidade de ordem n**, utilizando a função clock() da biblioteca “time.h” para determinar o tempo de resposta da chamada da função Determinante(). O computador utilizado para testes possui processador Core i7 - 7ª Ger. , memória de 8GB e sistema operacional Linux Ubuntu 18.04 x64. Para cada ordem foram realizadas quatro medidas, colhendo sua média para realizar uma normalização dos dados. O erro de cada medida é $\approx 0,000001$ s.

Pelo dados, pode-se inferir que o tempo cresce exponencialmente em função da ordem. Logo, calcular o determinante de uma matriz de ordem muito grande torna-se inviável tanto pelo consumo excessivo de memória quanto o elevado tempo de execução.

Tabela com os dados colhidos no experimento:

Ordem	Tempo(s)
1	0,000003
2	0,000005
3	0,000010
4	0,000021
5	0,000069
6	0,000371
7	0,002143
8	0,016350
9	0,106231
10	0,560285
11	5,17821825
12	62,73907325
13	755,134647

Plotando o gráfico:

