

Universidade de São Paulo

SSC0600 - Introdução a ciência da computação I

Trabalho 3 : Matrizes.

junho/2018

Índice

| | |
|---------------------------|----|
| Introdução..... | 3 |
| Descrição do Projeto..... | 4 |
| Tutorial..... | 5 |
| Outras informações..... | 10 |
| Referências..... | 10 |

1 - Introdução

Participantes:

P=0 : Antonio Sebastian Fernandes Rabelo 10797781

P=1 : Renata Oliveira Brito 10373663

P=2 : Helbert Moreira Pinto 10716504

P=3 : Victoria Helena Ianni 10255900

P=4 : Gustavo Machado Mendes 1082115

Este trabalho, visa a implementação de um programa que possa fazer operações básicas em uma matriz esparsa.

Matriz é um arranjo (tabela) retangular de números dispostos em linhas e colunas.

As linhas e colunas podem variar de 0 a quantidade desejada.

Podendo ser $3 \times 4 = A$ ou $3 \times 3 = B$ na figura 0.

$$\begin{matrix} A \\ 3 \times 4 \end{matrix} \begin{bmatrix} 1 & 0 & 4 & -3 \\ 2 & 5 & 3 & 4 \\ 9 & 8 & -2 & 1 \end{bmatrix} \qquad \begin{matrix} B \\ 3 \times 3 \end{matrix} \begin{bmatrix} 3 & 7 & 4 \\ 1 & 0 & 6 \\ 9 & 2 & 8 \end{bmatrix}$$

figura 00 : exemplo de matriz

Matriz esparsa é uma matriz com excessivo nº de elementos nulos (0), como exemplo abaixo:

$$\begin{matrix} C \\ 700 \times 900 \end{matrix} \begin{bmatrix} 1 & 0 & 0 & 3 & 0 & 0 & 0 & \dots & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & -1 & 4 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & \dots & 0 \end{bmatrix}$$

figura 00.1 : exemplo de matriz esparsa

A matriz da figura 00.1 mostra uma matriz esparsa com 630.000 elementos e 6 elementos não nulos.

Temos então algumas informações do uso de matriz esparsa como:

Vantagem:

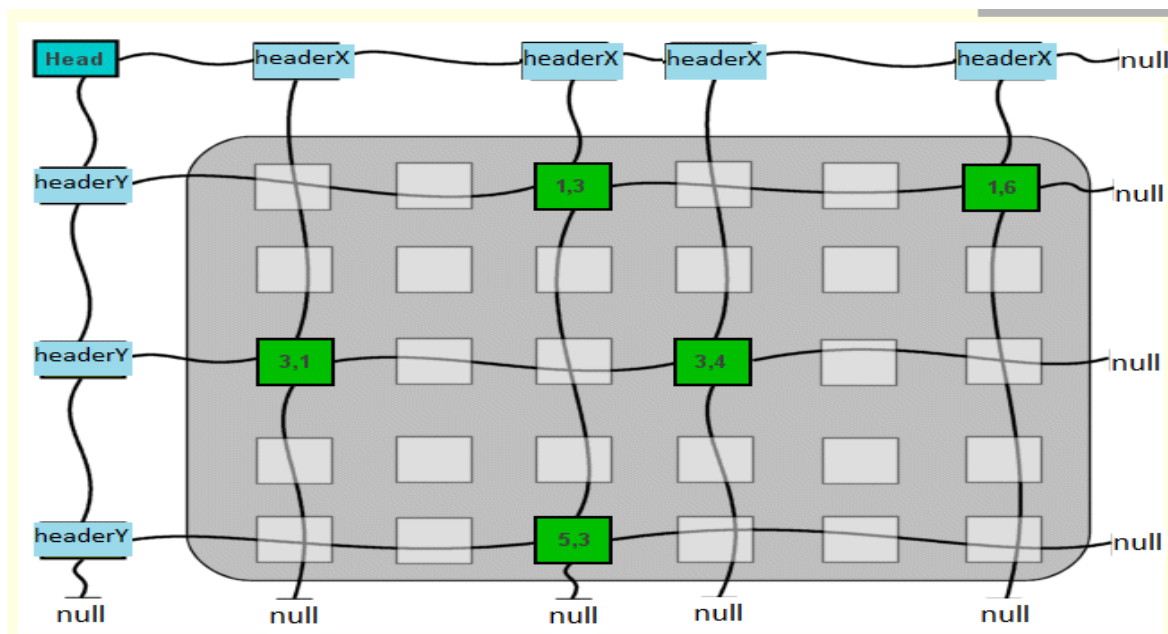
- O acesso é direto a cada elemento da matriz.

Desvantagem:

- Muito espaço para armazenar 0.

A matriz principal consiste em uma lista cruzada, segundo o modelo abaixo.

Implementamos este tipo de estrutura de dados, pois economiza processamento nas buscas e operações das funções implementadas.



No programa o usuário poderá escolher as opções por meio de um menu, as opções são:

1. Criação da matriz m por n.
2. Exclusão da matriz.
3. Inserir um valor na posição (x, y) da matriz
4. Consulta dos valores de uma posição (x, y) da matriz
5. Consulta da soma dos valores de cada linha da matriz
6. Consulta da soma dos valores de cada coluna da matriz

2 - Descrição do projeto

Para desenvolver o projeto, utilizamos a IDE de ambiente de desenvolvimento CodeBlocks (versão 17.12 - x86x64) e para compilar o fonte utilizamos o GCC. O sistema operacional utilizado foi o Windows 10 (versão 64bits). O código fonte está no arquivo main.c, o arquivo compilado é o main.o. Os headers utilizados foram:

- stdlib.h
- stdio.h
- conio.h

3 - Tutorial

- As telas iniciais são:



figura 01 : tela inicial

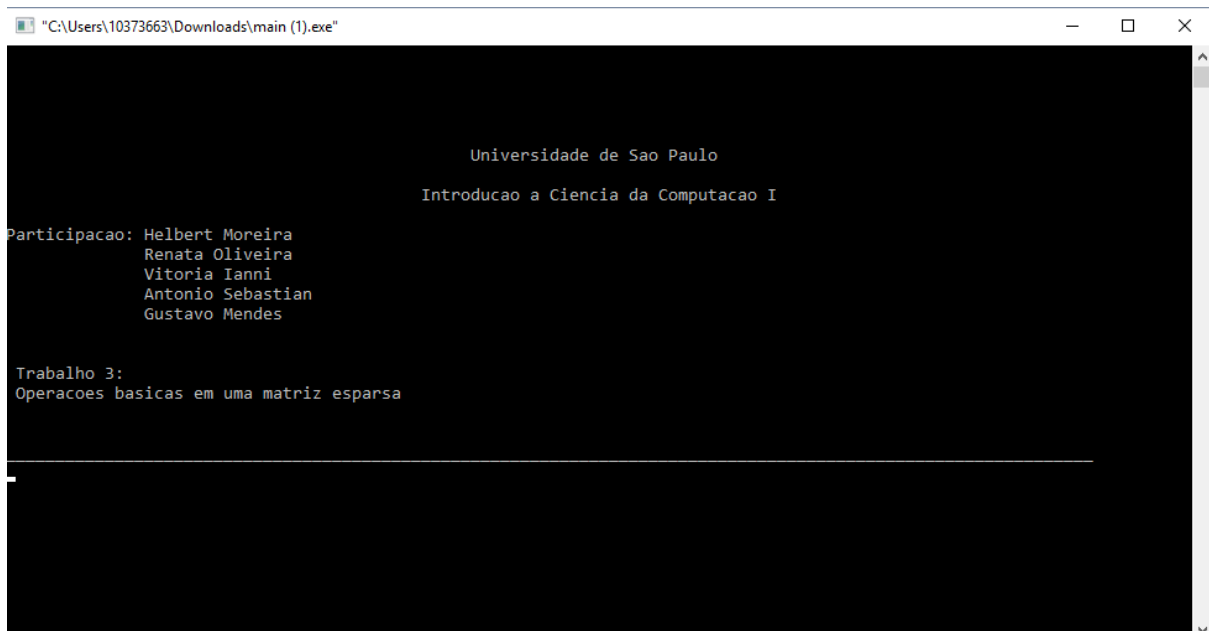


figura 02 : tela após o primeiro enter

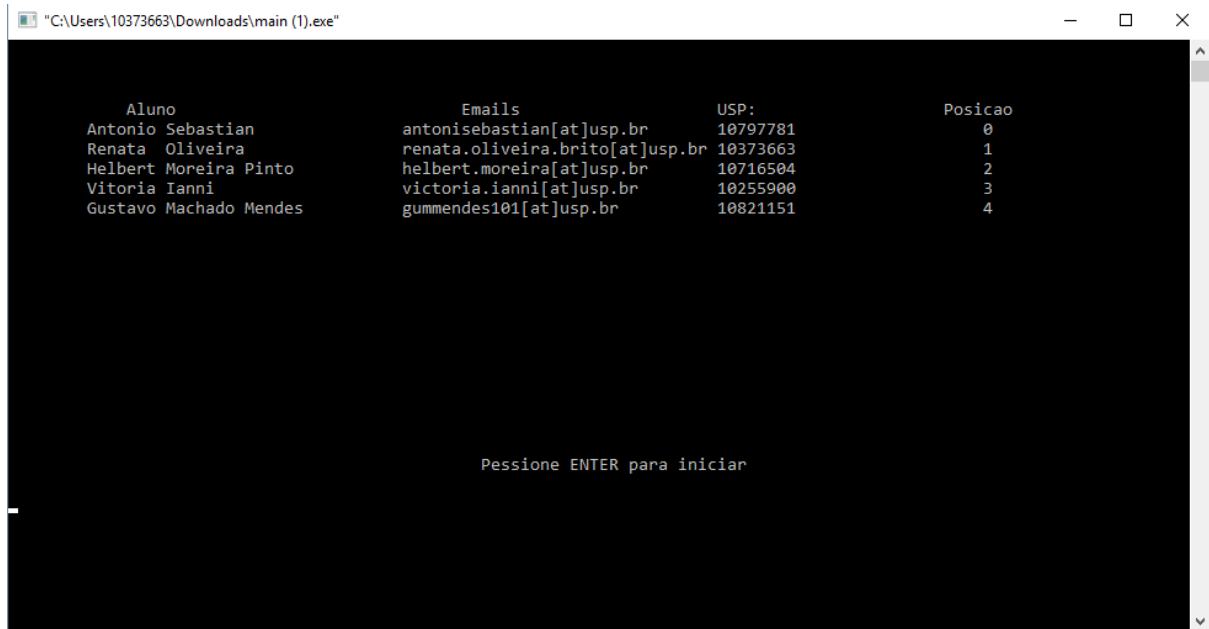


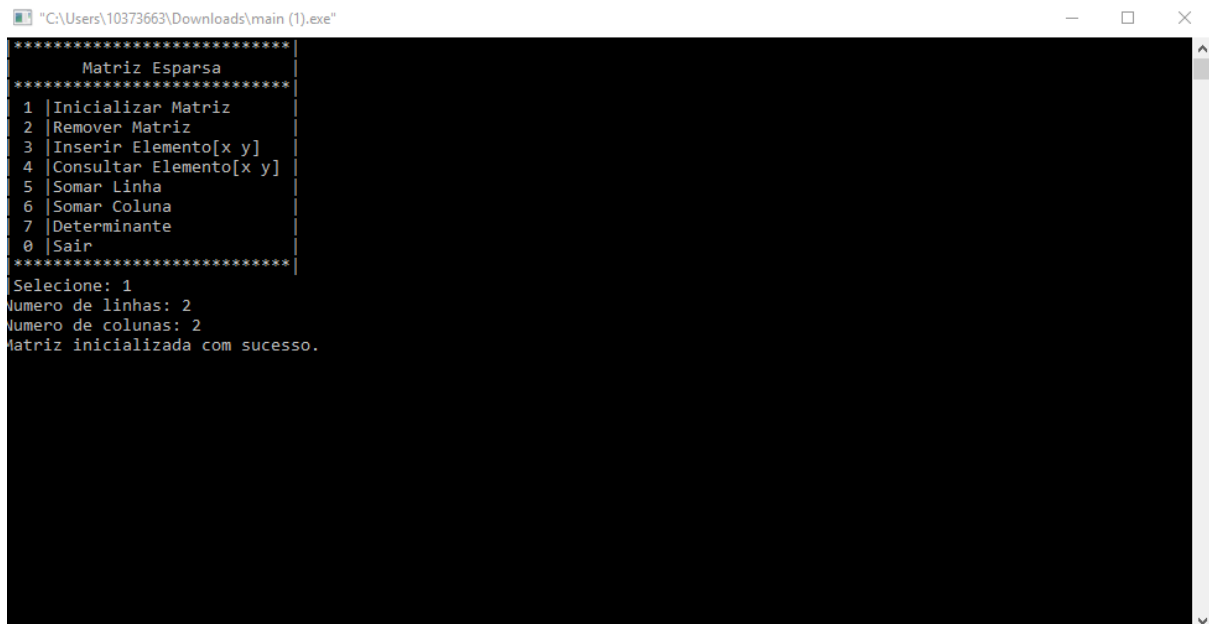
figura 03 : tela após o segundo enter

- O menu inicial:



figura 04 : tela menu principal

- Depois da escolha 1 da figura 04, a figura 05 mostra para digitar o número de linhas e colunas da matriz que deseja inicializar.



```
*****
Matriz Esparsa
*****
1 | Inicializar Matriz
2 | Remover Matriz
3 | Inserir Elemento[x y]
4 | Consultar Elemento[x y]
5 | Somar Linha
6 | Somar Coluna
7 | Determinante
0 | Sair
*****
Seleção: 1
Numero de linhas: 2
Numero de colunas: 2
Matriz inicializada com sucesso.
```

figura 05 : tela opção 1 do menu.

- Depois de inicializar a matriz e a opção 2 na figura 04, a figura 06 mostra que a matriz foi removida.



```
*****
Matriz Esparsa
*****
1 | Inicializar Matriz
2 | Remover Matriz
3 | Inserir Elemento[x y]
4 | Consultar Elemento[x y]
5 | Somar Linha
6 | Somar Coluna
7 | Determinante
0 | Sair
*****
Seleção: 2
Matriz removida com sucesso!
```

figura 06 : tela opção 2 do menu.

- Depois da opção 3 da figura 04, a figura 07 mostra que é necessário digitar a posição que deseja adicionar algum elemento e caso essa posição esteja ocupada o programa pergunta se deseja alterar o dado que já havia ou não.

```

*****
Matriz Esparsa
*****
1 | Inicializar Matriz
2 | Remover Matriz
3 | Inserir Elemento[x y]
4 | Consultar Elemento[x y]
5 | Somar Linha
6 | Somar Coluna
7 | Determinante
0 | Sair
*****
Selecione: 3
Digite a posicao [x y]: 0
1
Digite o valor: 5
Casa ocupada com o valor 1.00
Deseja alterar os dados? [S/N]

```

figura 07 : tela opção 3 do menu.

- Depois da escolha 4 da figura 04, a figura 08 mostra que precisa digitar a posição que deseja consultar o elemento.

```

*****
Matriz Esparsa
*****
1 | Inicializar Matriz
2 | Remover Matriz
3 | Inserir Elemento[x y]
4 | Consultar Elemento[x y]
5 | Somar Linha
6 | Somar Coluna
7 | Determinante
0 | Sair
*****
Selecione: 4
Digite a posicao [x y]: 0
1
[0,1]: 1.00

```

figura 08 : tela opção 4 do menu.

- Depois de adicionado os seguintes elemento em uma matriz 2x2.
- $(0,1) = 1$ $(0, 0) = 3$ $(1, 0) = 2$ $(1, 1) = 5$.
- E escolhido a opção 5 da figura 04, a figura 09 mostra a soma da linha 1 (pois foi a selecionada).

```

*****
Matriz Esparsa
*****
1 | Inicializar Matriz
2 | Remover Matriz
3 | Inserir Elemento[x y]
4 | Consultar Elemento[x y]
5 | Somar Linha
6 | Somar Coluna
7 | Determinante
0 | Sair
*****
Selezione: 5
Digite a linha: 1
Soma da linha 1: 7.00

```

figura 09 : tela opção 5 do menu.

- Com a mesma matriz da informações da figura 09, e escolhido a opção 6 da figura 04, a figura 10 mostra a soma da coluna 1 (pois foi a selecionada).

```

*****
Matriz Esparsa
*****
1 | Inicializar Matriz
2 | Remover Matriz
3 | Inserir Elemento[x y]
4 | Consultar Elemento[x y]
5 | Somar Linha
6 | Somar Coluna
7 | Determinante
0 | Sair
*****
Selezione: 6
Digite a coluna: 1
Soma da coluna 1: 6.00

```

figura 10 : tela opção 6 do menu.

4. OUTRAS INFORMAÇÕES

Só pode adicionar uma matriz por vez, para iniciar outra tem que remover a atual ou encerrar o programa e inicializar novamente.

Só adiciona um elemento por vez, depois de adicionado um ele volta ao menu para que possa escolher uma nova opção.

Se usuário tentar atualizar o valor de uma célula já preenchida com zero, ele remove toda a estrutura na lista cruzada para o elemento, pois o valor padrão é o zero e se usuário tentar salvar um valor zero em uma célula vazia, ele não insere nada . Isso ajuda a economizar espaço na memória.

A aplicação calcula somente determinantes de matrizes de ordem até 4.

5. Referências

http://wiki.icmc.usp.br/images/b/b1/Matrizes_esparsas_RAFR.pdf