UNIVERSIDADE FEDERAL DO CEARÁ

Universidade Federal do Ceará

CAMPUS QUIXADÁ

Jorge Eduardo Silva Sousa

Matheus Conrado Pires

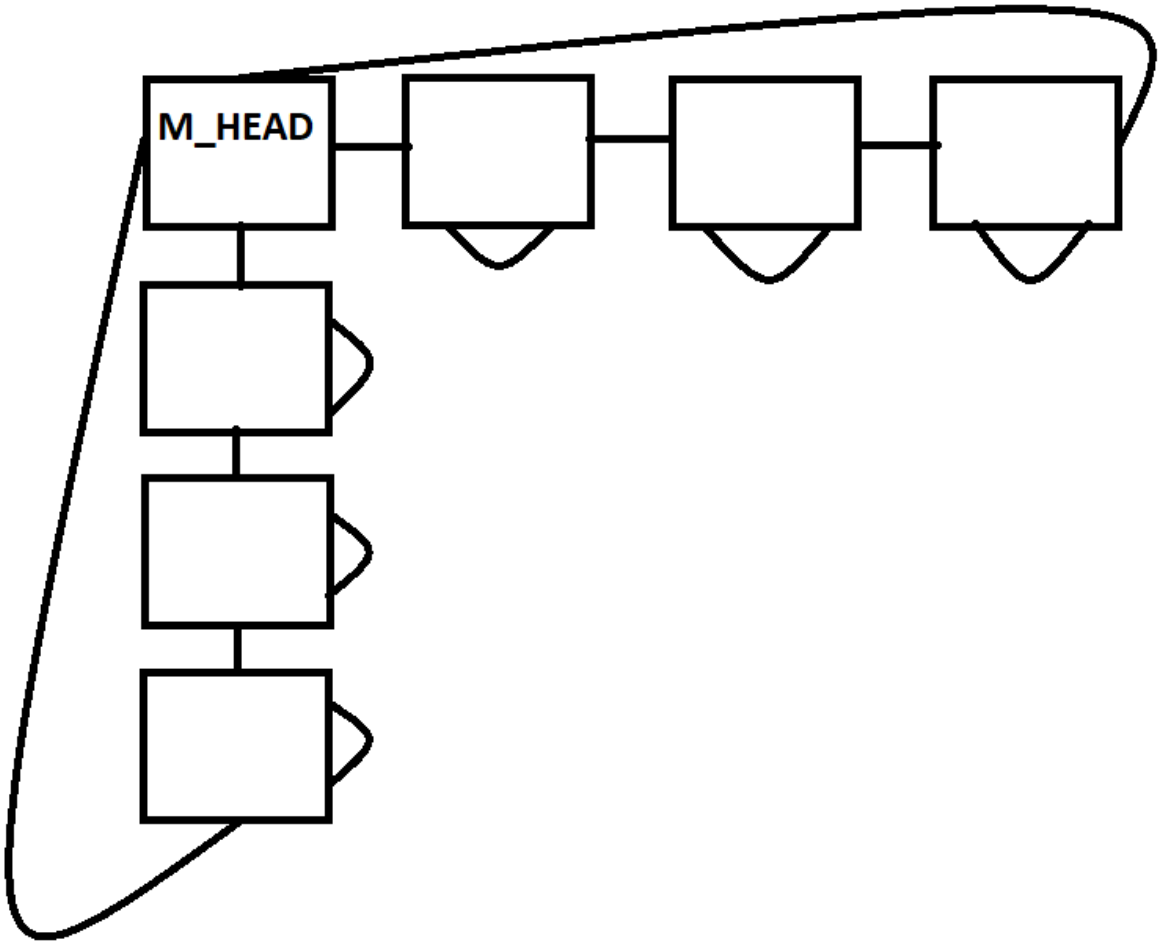
Ciência da Computação

Relatório do Projeto de Estrutura de Dados - Matrizes Esparsas

1. INTRODUÇÃO E IMPLEMENTAÇÃO

Não irei explicar o que é uma matriz esparsa, irei pular para a parte prática, ou seja, o que fizemos para montar nossa matriz esparsa. De início pensamos em ter um m_head na posicao [-1][-1] para demarcar o nó inicial da nossa matriz e fazer os nós sentinelas nascerem por meio dele, como na figura abaixo:

Figura 01 - Desenho mostrando o esquema da matriz esparsa.

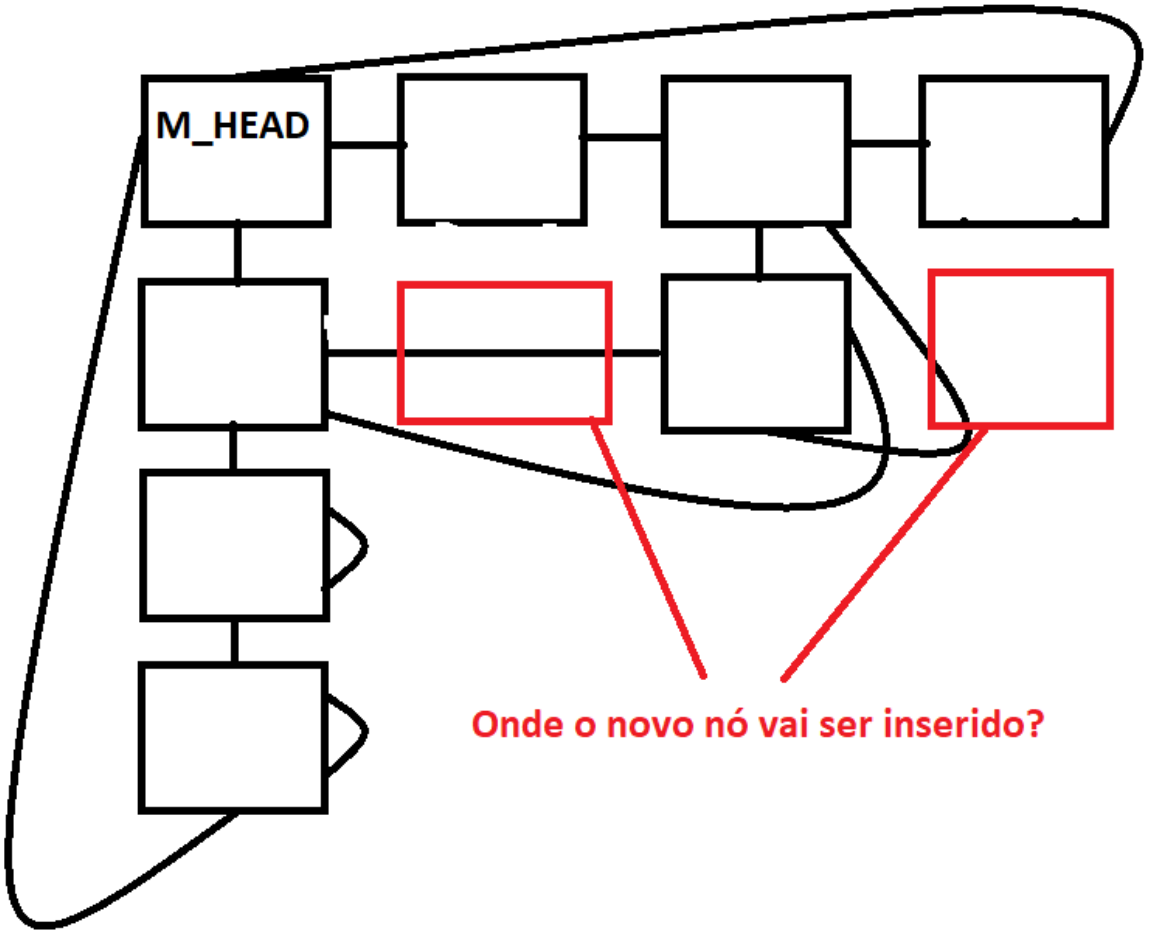


Fonte: Autor.

E assim foi construída a matriz, porém, ainda faltava inserir nós na matriz de fato, com isso, foi analisado os casos para o insert, chegamos na conclusão de que eram necessários 4 casos globais, esses casos eram: Caso 01: coluna está vazia e linha

está vazia, Caso 02: coluna tem nó e linha está vazia, Caso 03: coluna está vazia e linha tem no, e por fim, Caso 04: coluna e linha tem nó. Dentro de cada caso, podíamos ver mais casos, por exemplo, se no caso de linha com nó, o nó que desejamos inserir vai ser antes deste no já existente, exatamente nesse nó já existente ou depois desse nó já existente. imagem abaixo ilustra essa dúvida:

Figura 02 - Casos dentro dos casos.



Fonte: Autor.

Após criar todos esses casos chegamos em um lugar, todos os nós que criamos eram alocados corretamente, porém, por intermédio de colegas, ouvimos que fizemos muitas condições, o que pode prejudicar a complexidade do nosso

programa, mas o que importa é que foi feito com nossa lógica, pensamos todos os casos e chegamos no que foi feito e entregue, até sem precisar fazer pesquisa ou procurar referências na internet, só com o que a gente aprendeu de verdade na disciplina.

2. DIVISÃO DE TRABALHO ENTRE A DUPLA

Não foi feita nenhum tipo de divisão, o trabalho foi feito de maneira linear, onde nos reunimos por meio de chamadas no discord, compartilhamos a tela e íamos codando ao mesmo tempo, como uma programação pareada, ora Jorge codava e Matheus acompanhava dando ideias, ora o inverso, e assim foi feito até o fim do projeto.

3. FUNÇÕES AUXILIARES

Na main, foi decidido fazer um menu e deixá-la interativa, por isso foi necessário criar um função que servia para ler uma matriz, e outra que servia para inserir a matriz dentro de um vetor de matrizes por meio de alocação dinâmica. na classe, também criamos dois gets, um para rows e outra para cols, para sabermos quantas linhas e colunas tinha aquele objeto.

4. DIFICULDADES

Acredito que houveram poucas dificuldades, o método insert que deveria ser o mais problemático, se mostrou simples de construir, talvez pelo planejamento prévio antes de fazê-lo. com certeza o que mais deu dor de cabeça foi o método print, pois de início ignoramos o get que já tinha no código, e queríamos construir um print do zeros mesmo, mas em vez disso, usamos o get, e fizemos um simples for dentro de for para printar uma matriz normal, o que mudava era que chamávamos o get com os valores de i e de j, dentro do get ele retornava o valor do nó, se não houvesse nó, retornava um 0.

PROBLEMAS: nos finalmentes, achamos um bug, ao usar nossa main interativa, caímos nesse bug, porém se comentar a main interativa, e criar uma matriz e dar o delete, ela apagam tranquilamente, sem erros, mas na main interativa temos o seguinte problema, ela cria a matriz, aloca dinamicamente, e simplesmente já deleta ela, tornando inviável o uso da main interativa, pois isso, se quiser usar o destrutor, mandaremos ele comentando, junto com o código na main que roda ele, comentando também. **PARA USAR O DESTRUTOR, comente a main interativa e descomente o código do destrutor na classe e o final da int main que roda um código que testa esse destrutor.**

5. LISTAGEM DE TESTES

Fizemos testes com soma e multiplicação, onde por si só, se utiliza de todas as funções do código, primeiro a soma:

Figura 03 - Exemplo de Soma.

$$A = \begin{bmatrix} 1 & 2 \\ 5 & 4 \end{bmatrix} \quad B = \begin{bmatrix} 1 & 2 \\ 5 & 4 \end{bmatrix}$$

$$A+B= \begin{bmatrix} 1+1 & 2+2 \\ 5+5 & 4+4 \end{bmatrix}$$

$$C = \begin{bmatrix} 2 & 4 \\ 10 & 8 \end{bmatrix}$$

Fonte: Enigmus Academy.

Mostrando a matrix A, inserindo e printando no programa:

```
Matrizes disponiveis:
0 - 2x3
1 - 2x2
2 - 2x2
Matriz: 1
1 2
5 4
========================================================================================================
1 - Criar Matriz
2 - Somar Matrizes
3 - Multiplicar Matrizes
4 - Printar uma matriz
5 - Sair
========================================================================================================
Opcao:
```

Mostrando a matrix B, inserindo e printando no programa:

```
Matrizes disponiveis:
0 - 2x3
1 - 2x2
2 - 2x2
Matriz: 2
1 2
5 4
========================================================================================================
1 - Criar Matriz
2 - Somar Matrizes
3 - Multiplicar Matrizes
4 - Printar uma matriz
5 - Sair
========================================================================================================
Opcao:
```

Resultado abaixo:

```
Matriz resultante:
2 4
10 8
========================================================================================================
1 - Criar Matriz
2 - Somar Matrizes
3 - Multiplicar Matrizes
4 - Printar uma matriz
5 - Sair
========================================================================================================
Opcao:
```

Agora o exemplo de multiplicação:

$$A = \begin{bmatrix} 2 & 3 \\ 4 & 6 \end{bmatrix} \quad B = \begin{bmatrix} 1 & 3 & 0 \\ 2 & 1 & 1 \end{bmatrix}$$

$$A \times B = \begin{bmatrix} 2 & 3 \\ 4 & 6 \end{bmatrix} \times \begin{bmatrix} 1 & 3 & 0 \\ 2 & 1 & 1 \end{bmatrix} =$$

$$\begin{bmatrix} 2 \times 1 + 3 \times 2 & 2 \times 3 + 3 \times 1 & 2 \times 0 + 3 \times 1 \\ 4 \times 1 + 6 \times 2 & 4 \times 3 + 6 \times 1 & 4 \times 0 + 6 \times 1 \end{bmatrix} =$$

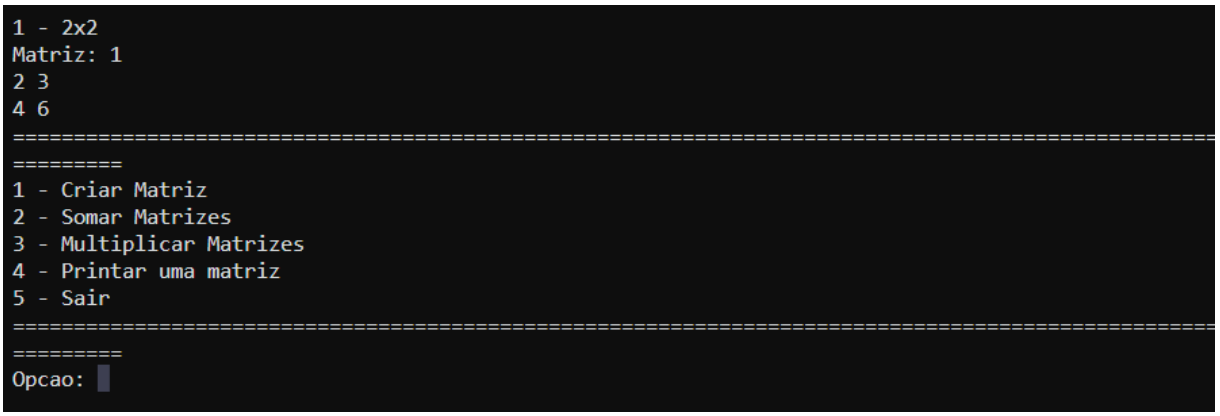
$$\begin{bmatrix} 2 + 6 & 6 + 3 & 0 + 3 \\ 4 + 12 & 12 + 6 & 0 + 6 \end{bmatrix} =$$

$$\begin{bmatrix} 8 & 9 & 3 \\ 16 & 18 & 6 \end{bmatrix}$$

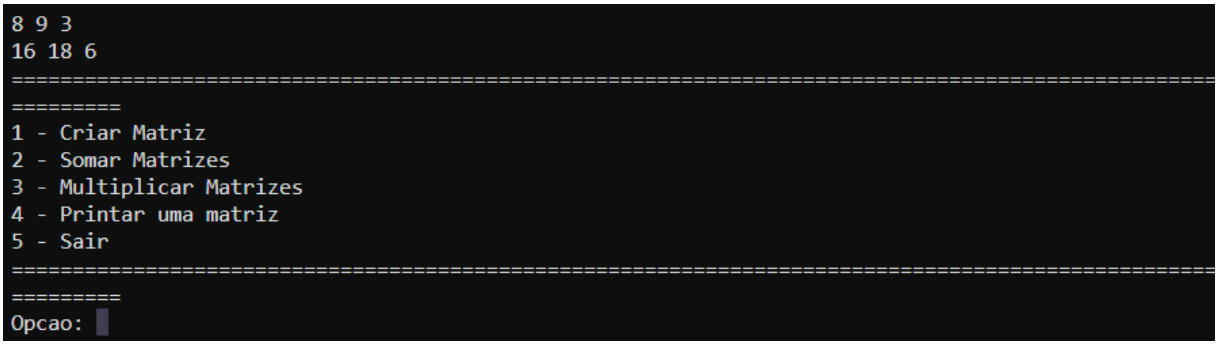
Primeiro, a matriz B está no arquivo e já foi lida para o programa:

```
Matrizes disponiveis:
0 - 2x3
1 - 2x2
Matriz: 0
1 3 0
2 1 1
=============================================================================================================
=========
1 - Criar Matriz
2 - Somar Matrizes
3 - Multiplicar Matrizes
4 - Printar uma matriz
5 - Sair
=============================================================================================================
=========
Opcao:
```

Insiro a matrix A digitando, e printo:



Multiplico A.B e resulta nisso:



6 ANÁLISE DE COMPLEXIDADE

1. **void SparseMatrix::insert(int row, int col, double value)**

```
void SparseMatrix::insert(int row, int col, double value){
    Node *auxRow = m_head;
    for(int i = 0; i <= row; i++){
       auxRow = auxRow->down;        // C1 * n
    }
    Node *auxCol = m_head;
    for(int i = 0; i <= col; i++){
        auxCol = auxCol->right;      //C2 * n
    }
    if(auxRow->right == auxRow && auxCol->down == auxCol){  // C3
        Node *ajuda = new Node(auxRow, auxCol, row, col, value);
        auxRow->right = ajuda;
        auxCol->down = ajuda;
    }
    else if(auxRow->right == auxRow && auxCol->down != auxCol){ //C4
        Node *MoovableCol = auxCol;
        while(MoovableCol->down != auxCol){
            MoovableCol = MoovableCol->down;
        }
        if(row == MoovableCol->row){
            MoovableCol->value = value;
        }
        if(row < MoovableCol->row){
            Node *novo = new Node(auxRow, auxCol->down, row, col, value);
            auxRow->right = novo;
            auxCol->down = novo;
        }
        else if(row > MoovableCol->row){
            Node *novo = new Node(auxRow, auxCol, row, col, value);
            MoovableCol->down = novo;
            auxRow->right = novo;
        }
    }else if(auxRow->right != auxRow && auxCol->down == auxCol){ //C5
        Node *MoovableRow = auxRow;
        while(MoovableRow->right != auxRow){
            MoovableRow = MoovableRow->right;
        }
        if(col == MoovableRow->col){
            MoovableRow->value = value;
        }
        if(col < MoovableRow->col){
            Node *novo = new Node(auxRow->right, auxCol, row, col, value);
            auxRow->right = novo;
            auxCol->down = novo;
        }
        else if(col > MoovableRow->col){
            Node *novo = new Node(auxRow, auxCol, row, col, value);
            MoovableRow->right = novo;
            auxCol->down = novo;
        }
    }
```

```
else {
        Node MoovableRow = auxRow;
        while(MoovableRow->right != auxRow){
            MoovableRow = MoovableRow->right;    //C6 n
        }
        Node MoovableCol = auxCol;
        while(MoovableCol->down != auxCol){
            MoovableCol = MoovableCol->down;     //C7 n
        }
        if(col == MoovableRow->col && row == MoovableCol->row){ //C8
            MoovableRow->value = value;
        }
        if(col < MoovableRow->col && row < MoovableCol->row){    //C9
            Node novo = new Node(auxRow->right, auxCol->down, row, col, value);
            auxRow->right = novo;
            auxCol->down = novo;
        }
        else if(col > MoovableRow->col && row > MoovableCol->row){    //C10
            Nodenovo = new Node(auxRow, auxCol, row, col, value);
            MoovableRow->right = novo;
            MoovableCol->down = novo;
        }
        else if(col < MoovableRow->col && row > MoovableCol->row){    //C11
            Node novo = new Node(auxRow->right, auxCol, row, col, value);
            auxRow->right = novo;
            MoovableCol->down = novo;
        }
        else if(col > MoovableRow->col && row < MoovableCol->row){    //C12
            Nodenovo = new Node(auxRow, auxCol->down, row, col, value);
            MoovableRow->right = novo;
            auxCol->down = novo;
        }
        if(auxCol->down->col == col && auxRow->right->row == row && auxCol->down != auxCol && auxRow-
>right != auxRow){            //C13
            Node MovRow = auxCol;
            while(MovRow->down->row != row){
                MovRow = MovRow->down;            //C14 n
            }
            MovRow = MovRow->down;
            MovRow->value = value;
        }
    }
}
// C1(n) + C2(n) + C3 + C4 + C5 + C6(n) + C7(n) + C8 + C9 + C10 + C11 + C12 + C13 + C14(n)
// A = C3 + C4 + C5 + C8 + C9 + C10 + C11 + C12 + C13\
// B = C1 + C2 + C6 + C7 + C14)
// C = A + B
// A + C1(n) + C2(n) + C6(n) + C7(n) + C14(n)
// A + n(C1 + C2 + C6 + C7 + C14)
// A + n(B) <= A(n) + B(n)
// A + n(B) <= C(n)
// A + n(B) <= O(n)
```

2. **double SparseMatrix::get(int row, int col):**

```
double SparseMatrix::get(int row, int col){
    Node *auxRow = m_head->down;//C1
    while(auxRow->row != row){
        auxRow = auxRow->down;//C2 * (n)
    }
    Node *MoovableRow = auxRow;//C3
    while(MoovableRow->col != col){
        MoovableRow = MoovableRow->right;//C4 * (n)
        if(MoovableRow->col == -1){//C5 * (n)
            return 0;
        }
    }
    if(MoovableRow->col == col){//C6
        return MoovableRow->value;
    }
    return 0;
}
/*
C1+C2*n+C3+C4*n+C5*n+C6
C1+C2n+C3+C4n+C5N+C6
C1+C3+C6 = A
C2N+C4N+C5N = n(C2+C4+C5)
C2+C4+C5 = B
A+nB
A+nB <= An+Bn
A+nB <= n(A+B)
A+B = C
A+nB <= n(C)
A+nB <= O(n)
*/
```

3. **SparseMatrix *sum(SparseMatrix* A, SparseMatrix* B)**

```cpp
SparseMatrix *sum(SparseMatrix* A, SparseMatrix* B){
    if(A->rows() != B->rows() || A->cols() != B->cols()){ // C1
        throw "ERRO: Matrizes de tamanhos diferentes!";
    }
    SparseMatrix *C = new SparseMatrix(A->rows(), A->cols());
    for(int i = 0; i < A->rows(); i++){
        for(int j = 0; j < A->cols(); j++){
            double value = A->get(i, j) + B->get(i, j); // (C2 * n + C3 * n) * n²
            C->insert(i, j, value); // C4 * n
        }
    }
    return C;
}
// C1 + (C2(n) + C3(n)) * n² + C4(n)
// A = C1
// B = C2 + C3
// C = C4
// A + n³(C2+C3) + C4(n)
// A + B(n³) + C(n)
// B(n³) + C(n)
// O(n³ + n)
```

7 CONCLUSÃO

Apesar do bug na main interativa com o destrutor, acho que entendemos bem o que é cada elemento de estrutura de dados dentro dessa matriz, vários elementos de listas e várias coisas que vimos nas aulas, foi divertido, e espero que o professor consiga descobrir o causador desse bug, para tentarmos resolver