

Estruturas de Dados

Matrizes e Matrizes Esparsas

Professores: Anisio Lacerda
Wagner Meira Jr.

Matrizes

Matrizes são estruturas extremamente versáteis e amplamente utilizadas.

O objetivo deste módulo é entender um pouco mais desse tipo de dado e conhecer algumas maneiras de atacar um de seus maiores problemas: consumo de espaço.

Matrizes

Assim como nos outros TADs que estudamos, vamos ilustrar algumas operações, são elas:

- Construir uma matriz
- Obter e alterar elementos da matriz.
- Imprimir seus elementos
- Produto de uma matriz com um vetor **denso**.
- Soma de duas matrizes.

Antes de entendermos a diferença entre matrizes densas e esparsas, vamos criar um TAD básico de matrizes com essas operações.

Matrizes

Observação importante!

As implementações nesta seção não necessariamente são as mais otimizadas, o propósito delas é ilustrar e exemplificar o TAD.

Matrizes

```
class Matriz{  
    public:  
        Matriz(int linhas, int colunas, int leTeclado);  
        ~Matriz();  
  
        double Get(int linha, int coluna);  
        void Set(int linha, int coluna, double valor);  
        void Imprime();  
        static Matriz Soma(Matriz A, Matriz B);  
        static double* Produto(Matriz A, double* v, int n);  
    private:  
        double** M;  
        int l, c;  
        void LeMatriz();  
};
```

Matrizes

```
Matriz::Matriz(int linhas, int colunas, int leTeclado){  
    l = linhas,  
    c = colunas;  
    M = new double*[l];  
    if(!leTeclado){  
        for(int i = 0; i < l; i++){  
            M[i] = new double[c];  
            for(int j = 0; j < c; j++){  
                M[i][j] = 0;  
            }  
        }  
    }  
    else {  
        LeMatriz();  
    }  
}
```

Matrizes

```
void Matriz::LeMatriz() {  
    for(int i = 0; i < l; i++) {  
        M[i] = new double[c];  
        for(int j = 0; j < c; j++) {  
            cin >> M[i][j];  
        }  
    }  
}
```

```
Matriz::~~Matriz() {  
    for(int i = 0; i < l; i++) {  
        delete[] (M[i]);  
    }  
    delete[] (M);  
}
```

Matrizes

```
double Matriz::Get(int linha, int coluna){  
    if(linha < 0 || linha >= 1 || coluna < 0 || coluna >= c){  
        throw "Posicao invalida!!!\n";  
    }  
    return M[linha][coluna];  
}
```

```
void Matriz::Set(int linha, int coluna, double valor){  
    if(linha < 0 || linha >= 1 || coluna < 0 || coluna >= c){  
        throw "Posicao invalida!!!\n";  
    }  
    M[linha][coluna] = valor;  
}
```


Matrizes

```
void Matriz::Imprime() {  
    for(int i = 0; i < l; i++) {  
        for(int j = 0; j < c; j++) {  
            cout << M[i][j] << " ";  
        }  
        cout << "\n";  
    }  
}
```

Matrizes

```
double* Matriz::Produto(Matriz A, double* v, int n){
    if(A.c != n){
        throw "Dimensoes das matrizes incompativeis\n";
    }

    double* s = new double[n];
    for(int i = 0; i < n; i++){
        s[i] = 0;
    }

    for(int i = 0; i < n; i++){
        for(int j = 0; j < n; j++){
            s[i] += A.M[i][j] * v[j];
        }
    }

    return s;
}
```

Matrizes

```
Matriz Matriz::Soma(Matriz A, Matriz B) {  
    if(A.l != B.l || A.c != B.c) {  
        throw "Dimensoes das matrizes incompativeis\n";  
    }  
    Matriz C = Matriz(A.l, A.c, 0);  
    for(int i = 0; i < A.l; i++) {  
        for(int j = 0; j < A.c; j++) {  
            C.Set(i, j, A.M[i][j]+B.M[i][j]);  
        }  
    }  
    return C;  
}
```

Matrizes densas vs esparsas

- Uma matriz é **densa** quando a maior parte de seus elementos **não** são 0, e é **esparsa** quando a maior parte dos elementos é 0.

Matrizes densas vs esparsas

- Uma matriz é **densa** quando a maior parte de seus elementos **não** são 0, e é **esparsa** quando a maior parte dos elementos é 0.
- Note que no TAD dos slides anteriores, caso a matriz seja esparsa, muita memória e computação nas operações serão gastos com os elementos nulos.

Matrizes densas vs esparsas

- Uma matriz é **densa** quando a maior parte de seus elementos **não** são 0, e é **esparsa** quando a maior parte dos elementos é 0.
- Note que no TAD dos slides anteriores, caso a matriz seja esparsa, muita memória e computação nas operações serão gastos com os elementos nulos.
- Nosso objetivo é encontrar formas de representar matrizes as quais reduzimos o gasto de memória e computação com elementos nulos.

Matrizes esparsas

- Nosso objetivo é encontrar uma representação que omita os elementos nulos.

Matrizes esparsas

- Nosso objetivo é encontrar uma representação que omita os elementos nulos.
- Uma forma simples de se pensar é apenas endereçar os elementos de interesse, no nosso caso os não nulos.

Matrizes esparsas

- Nosso objetivo é encontrar uma representação que omita os elementos nulos.
- Uma forma simples de se pensar é apenas endereçar os elementos de interesse, no nosso caso os não nulos.
- Chamamos essa forma de representar matrizes esparsas de *Coordinate list* (COO).

Matrizes esparsas (COO)

- Nessa representação para cada elemento iremos armazenar três informações:

Matrizes esparsas (COO)

- Nessa representação para cada elemento iremos armazenar três informações:
 - A linha onde o elemento está.

Matrizes esparsas (COO)

- Nessa representação para cada elemento iremos armazenar três informações:
 - A linha onde o elemento está.
 - A coluna onde o elemento está.

Matrizes esparsas (COO)

- Nessa representação para cada elemento iremos armazenar três informações:
 - ❑ A linha onde o elemento está.
 - ❑ A coluna onde o elemento está.
 - ❑ O valor do elemento.

Dessa forma nossa estrutura será um arranjo em que cada “linha” armazena essas três informações.

Exemplo COO

Exemplo de uma matriz e sua representação

3	0	0	15
0	0	4	0
0	7	0	0
0	0	0	12

linhas:

0	0	1	2	3
---	---	---	---	---

colunas:

0	3	2	1	3
---	---	---	---	---


valores:

3	15	4	7	12
---	----	---	---	----

Exemplo COO

Apesar da apresentação visual, por questões de localidade de referência, cada entrada será representada como uma “**linha**” em nossa estrutura.

3	0	0	15
0	0	4	0
0	7	0	0
0	0	0	12



linhas:	0	0	1	2	3
colunas:	0	3	2	1	3
valores:	3	15	4	7	12

Outro exemplo COO (um pouco maior)

	1	2	3	4	5	6	7	8
1								
2			A			B		
3			C					
4			D					
5			E					
6					F	G	H	
7								
8								

8x8



	lin	col	info
1	2	3	A
2	2	6	B
3	3	3	C
4	4	3	D
5	5	3	E
6	6	5	F
7	6	6	G
8	6	7	H

8x3

Matrizes esparsas (COO)

```
class MatrizCOO{  
    public:  
        MatrizCOO(int linhas, int colunas, int leTeclado);  
        ~MatrizCOO();  
  
        double Get(int linha, int coluna);  
        void Set(int linha, int coluna, double valor);  
        void Imprime();  
        static MatrizCOOSoma (MatrizCOO A, MatrizCOO B);  
        static double* Produto (MatrizCOO A, double* v, int t);  
  
    private:  
        double** M;  
        int tamanho, tamanhoMAX, l, c;  
        void LeMatriz();  
};
```

Matrizes esparsas (COO)

Observação: Ao longo desta implementação faremos uso de algumas funções auxiliares. Suas implementações serão omitidas pois são simples e não tem relação direta com o conteúdo do módulo.

Matrizes esparsas (COO)

```
double** GetArranjoTamanhoN(int N);
```

Esta função retorna um arranjo com N linhas e 3 colunas com 0 em todos os elementos.

Matrizes esparsas (COO)

```
void InsereElemento(int l, int c, int v);
```

Esta função insere um elemento de valor v na linha l e coluna c em nossa representação, aumentando o tamanho do arranjo caso necessário.

Além disso, após a inserção ela ordena os elementos por linha coluna.

Matrizes esparsas (COO)

```
void RemoveElemento(int p);
```

Esta função remove um elemento na posição p da nossa representação.

Além disso, após a remoção ela ordena os elementos por linha coluna.

Matrizes esparsas (COO)

- **Construir uma matriz.**
- Obter e alterar elementos da matriz.
- Imprimir seus elementos.
- Produto de uma matriz com um vetor denso.
- Soma de duas matrizes.

Matrizes esparsas (COO)

- **Construir uma matriz.**
- Obter e alterar elementos da matriz.
- Imprimir seus elementos.
- Produto de uma matriz com um vetor denso.
- Soma de duas matrizes.

Para ilustrar a construção, vamos supor que iremos receber os dados da matriz linha por linha, incluindo os elementos nulos.

Matrizes esparsas (COO)

```
MatrizCOO::MatrizCOO(int linhas, int colunas, int leTeclado){  
    l = linhas;  
    c = colunas;  
    tamanho = 0;  
  
    if(!leTeclado){  
        tamanhoMAX = l;  
        M = GetArranjoTamanhoN(tamanhoMAX);  
    }  
    else{  
        LeMatriz();  
    }  
}
```


Matrizes esparsas (COO)

```
MatrizCOO::LeMatriz() {  
    tamanhoMAX = floor(0.3*l*c);  
    M = GetArranjoTamanhoN(tamanhoMAX);  
    double x;  
    for(int i = 0; i < l; i++){  
        for(int j = 0; j < c; j++){  
            cin >> x;  
            InsereElemento(i, j, x);  
        }  
    }  
}
```

Exemplo COO

Considere a matriz abaixo. Vamos construir sua representação COO.

3	0	0	15
0	0	4	0
0	7	0	0
0	0	0	12

Exemplo COO

Vamos construir nossa tabela que irá armazenar as triplas de cada elemento não nulo.

3	0	0	15
0	0	4	0
0	7	0	0
0	0	0	12

linhas:

colunas:

valores:

Exemplo COO

Agora iremos ler os dados da matriz, e considerar apenas os elementos não nulos.

3	0	0	15
0	0	4	0
0	7	0	0
0	0	0	12

linhas:

0

colunas:

0

valores:

3

Exemplo COO

Agora iremos ler os dados da matriz, e considerar apenas os elementos não nulos.

3	0	0	15
0	0	4	0
0	7	0	0
0	0	0	12

linhas:

0	0
---	---

colunas:

0	3
---	---

valores:

3	15
---	----

Exemplo COO

Agora iremos ler os dados da matriz, e considerar apenas os elementos não nulos.

3	0	0	15
0	0	4	0
0	7	0	0
0	0	0	12

linhas:

0	0	1
---	---	---

colunas:

0	3	2
---	---	---

valores:

3	15	4
---	----	---

Exemplo COO

Agora iremos ler os dados da matriz, e considerar apenas os elementos não nulos.

3	0	0	15
0	0	4	0
0	7	0	0
0	0	0	12

linhas:

0	0	1	2
---	---	---	---

colunas:

0	3	2	1
---	---	---	---

valores:

3	15	4	7
---	----	---	---

Exemplo COO

Agora iremos ler os dados da matriz, e considerar apenas os elementos não nulos.

3	0	0	15
0	0	4	0
0	7	0	0
0	0	0	12

linhas:

0	0	1	2	3
---	---	---	---	---

colunas:

0	3	2	1	3
---	---	---	---	---

valores:

3	15	4	7	12
---	----	---	---	----

Exemplo COO

- Note que da forma que a matriz foi lida no exemplo, os elementos estão ordenados por linha e por coluna, mas isso não é um requisito do formato.

Exemplo COO

- Note que da forma que a matriz foi lida no exemplo, os elementos estão ordenados por linha e por coluna, mas isso não é um requisito do formato.
- No entanto é ideal que os elementos sejam mantidos ordenados, para que as operações possam ser executadas de forma mais eficiente.

Matrizes esparsas (COO)

- Construir uma matriz.
- **Obter e alterar elementos da matriz.**
- Imprimir seus elementos.
- Soma de duas matrizes.
- Produto de uma matriz com um vetor denso.

Matrizes esparsas (COO)

```
double MatrizCOO::Get(int linha, int coluna){
    if(linha < 0 || linha >= 1 || coluna < 0 || coluna >= c){
        throw "Posicao inválida!!!\n";
    }
    for(int i = 0; i < tamanho; i++){
        if(linha == M[i][0] && coluna == M[i][1])
            return M[i][2];
    }
    return 0;
}
```


Exemplo COO

Para obter um elemento na linha i coluna j vamos varrer nossa tabela comparando os valores de linha e coluna.

linhas:	0	0	1	2	2	3
colunas:	0	3	2	1	3	3
valores:	3	15	4	7	8	12

Exemplo COO


Vamos procurar pelo elemento linha 2 coluna 3.



linhas:	0	0	1	2	2	3
colunas:	0	3	2	1	3	3
valores:	3	15	4	7	8	12

Exemplo COO


Vamos procurar pelo elemento linha 2 coluna 3.



linhas:	0	0	1	2	2	3
colunas:	0	3	2	1	3	3
valores:	3	15	4	7	8	12

Exemplo COO


Vamos procurar pelo elemento linha 2 coluna 3.



linhas:	0	0	1	2	2	3
colunas:	0	3	2	1	3	3
valores:	3	15	4	7	8	12

Exemplo COO


Vamos procurar pelo elemento linha 2 coluna 3.



linhas:	0	0	1	2	2	3
colunas:	0	3	2	1	3	3
valores:	3	15	4	7	8	12

Exemplo COO


Vamos procurar pelo elemento linha 2 coluna 3.



linhas:	0	0	1	2	2	3
colunas:	0	3	2	1	3	3
valores:	3	15	4	7	8	12

Exemplo COO


Vamos procurar pelo elemento linha 2 coluna 3.



linhas:	0	0	1	2	2	3
colunas:	0	3	2	1	3	3
valores:	3	15	4	7	8	12

Exemplo COO

Vamos procurar pelo elemento linha 2 coluna 3.



linhas:	0	0	1	2	2	3
colunas:	0	3	2	1	3	3
valores:	3	15	4	7	8	12

Exemplo COO

- Qual a complexidade para encontrar um elemento?

Exemplo COO

- Qual a complexidade para encontrar um elemento?
 - A quantidade de elementos que uma matriz $n \times n$ pode ter é n^2 , no entanto esse limite seria muito “folgado”.

Exemplo COO

- Qual a complexidade para encontrar um elemento?
 - ❑ A quantidade de elementos que uma matriz $n \times n$ pode ter é n^2 , no entanto esse limite seria muito “folgado”.
 - ❑ Como nós temos que por hipótese nossa matriz é esparsa, vamos chamar de k a quantidade de elementos não nulos presentes na matriz.

Exemplo COO

- Qual a complexidade para encontrar um elemento?
 - ❑ A quantidade de elementos que uma matriz $n \times n$ pode ter é n^2 , no entanto esse limite seria muito “folgado”.
 - ❑ Como nós temos que por hipótese nossa matriz é esparsa, vamos chamar de k a quantidade de elementos não nulos presentes na matriz.
 - ❑ Dessa forma a complexidade é $O(k)$.

Matrizes esparsas (COO)

```
void MatrizCOO::Set(int linha, int coluna, double valor){
    if(linha < 0 || linha >= 1 || coluna < 0 || coluna >= c){
        throw "Posicao inválida!!!\n";
    }
    for(int i = 0; i < tamanho; i++){
        if(linha == M[i][0] && coluna == M[i][1]){
            if(valor == 0) RemoveElemento(i);
            else M[i][2] = valor;
            return;
        }
    }
    if(valor == 0) return;
    InsereElemento(linha, coluna, valor);
}
```

Exemplo COO

Desejamos alterar o elemento na linha 0 coluna 1 para 20.

linhas:	0	0	1	2	2	3
colunas:	0	3	2	1	3	3
valores:	3	15	4	7	8	12

Exemplo COO

Desejamos alterar o elemento na linha 0 coluna 1 para 20.

Como o elemento nessa posição é 0, devemos incluir um novo elemento na representação

linhas:	0	0	1	2	2	3	0
colunas:	0	3	2	1	3	3	1
valores:	3	15	4	7	8	12	20

Exemplo COO

Observe que agora os elementos da nossa representação não estão mais ordenados.

linhas:	0	0	1	2	2	3	0
colunas:	0	3	2	1	3	3	1
valores:	3	15	4	7	8	12	20

Exemplo COO

Observe que agora os elementos da nossa representação não estão mais ordenados. Podemos utilizar o método da inserção para inserir ordenado em $O(k)$.

linhas:	0	0	0	1	2	2	3
colunas:	0	1	3	2	1	3	3
valores:	3	20	15	4	7	8	12

Matrizes esparsas (COO)

- Construir uma matriz.
- Obter e alterar elementos da matriz.
- Imprimir seus elementos.
- Soma de duas matrizes.
- Produto de uma matriz com um vetor denso.

Para as últimas três funções, iremos assumir que os elementos da nossa representação estão ordenados por linha e coluna.

Matrizes esparsas (COO)

- Construir uma matriz.
- Obter e alterar elementos da matriz.
- **Imprimir seus elementos.**
- Soma de duas matrizes.
- Produto de uma matriz com um vetor denso.

Matrizes esparsas (COO)

```
void MatrizCOO::Imprime() {  
    int k = 0;  
    for(int i = 0; i < l; i++){  
        for(int j = 0; j < c; j++){  
            if(M[k][0] == i && M[k][1] == j){  
                cout << M[k][2] << " ";  
                k++;  
            }  
            else cout << "0 ";  
        }  
        cout << "\n";  
    }  
}
```


Exemplo COO

Vamos executar o código de impressão neste exemplo.

linhas:

colunas:

valores:

0	0	1	2	2	3
0	3	2	1	3	3
3	15	4	7	8	12

Exemplo COO

Vamos executar o código de impressão neste exemplo.

	0	1	2	3
0				
1				
2				
3				




linhas:
colunas:
valores:

0	0	1	2	2	3
0	3	2	1	3	3
3	15	4	7	8	12

Exemplo COO

Vamos executar o código de impressão neste exemplo.

	0	1	2	3
0				
1				
2				
3				



linhas:	0	0	1	2	2	3
colunas:	0	3	2	1	3	3
valores:	3	15	4	7	8	12

Exemplo COO

Vamos executar o código de impressão neste exemplo.

	0	1	2	3
0	3			
1				
2				
3				

linhas:

colunas:

valores:



	0	1	2	2	3
0	0				
0	3	2	1	3	3
3	15	4	7	8	12

Exemplo COO

Vamos executar o código de impressão neste exemplo.

	0	1	2	3
0	3	0		
1				
2				
3				

linhas:

colunas:

valores:



	0	1	2	2	3
0	0				
0	3	2	1	3	3
3	3	4	7	8	12

Exemplo COO

Vamos executar o código de impressão neste exemplo.

	0	1	2	3
0	3	0	0	
1				
2				
3				

linhas:

colunas:

valores:



0	0	1	2	2	3
0	3	2	1	3	3
3	15	4	7	8	12

Exemplo COO

Vamos executar o código de impressão neste exemplo.

	0	1	2	3
0	3	0	0	15
1				
2				
3				

linhas:
colunas:
valores:

0	0	1	2	2	3
0	3	2	1	3	3
3	15	4	7	8	12

Exemplo COO


Vamos executar o código de impressão neste exemplo.

	0	1	2	3
0	3	0	0	15
1	0			
2				
3				

linhas:

colunas:

valores:



0	0	1	2	2	3
0	0	3	2	1	3
3	3	15	4	7	8
12					

Exemplo COO

Vamos executar o código de impressão neste exemplo.

	0	1	2	3
0	3	0	0	15
1	0	0		
2				
3				

linhas:

colunas:

valores:



	0	0	1	2	2	3
0	0	3	2	1	3	3
3	3	15	4	7	8	12

Exemplo COO


Vamos executar o código de impressão neste exemplo.

	0	1	2	3
0	3	0	0	15
1	0	0	4	
2				
3				

linhas:

colunas:

valores:



0	0	1	2	2	3
0	0	3	2	3	3
1	3	15	4	7	8
2					
3					

Exemplo COO

Vamos executar o código de impressão neste exemplo.

	0	1	2	3
0	3	0	0	15
1	0	0	4	0
2				
3				

linhas:

colunas:

valores:

0	0	1	2	2	3
0	3	2	1	3	3
3	15	4	7	8	12



Exemplo COO

Vamos executar o código de impressão neste exemplo.

	0	1	2	3
0	3	0	0	15
1	0	0	4	0
2	0			
3				

linhas:

colunas:

valores:

	0	0	1	2	2	3
0	0	3	2	1	3	3
3	3	15	4	7	8	12



Exemplo COO

Vamos executar o código de impressão neste exemplo.

	0	1	2	3
0	3	0	0	15
1	0	0	4	0
2	0	7		
3				

linhas:
colunas:
valores:

0	0	1	2	2	3
0	3	2	1	3	3
3	15	4	7	8	12

Exemplo COO

Vamos executar o código de impressão neste exemplo.

	0	1	2	3
0	3	0	0	15
1	0	0	4	0
2	0	7	0	
3				

linhas:

colunas:

valores:

	0	0	1	2	2	3
0	0	3	2	1	3	3
3	3	15	4	7	8	12



Exemplo COO

Vamos executar o código de impressão neste exemplo.

	0	1	2	3
0	3	0	0	15
1	0	0	4	0
2	0	7	0	8
3				

linhas:
colunas:
valores:

0	0	1	2	2	3
0	3	2	1	3	3
3	15	4	7	8	12



Exemplo COO

Vamos executar o código de impressão neste exemplo.

	0	1	2	3
0	3	0	0	15
1	0	0	4	0
2	0	7	0	8
3	0			

linhas:

colunas:

valores:

0	0	1	2	2	3
0	3	2	1	3	3
3	15	4	7	8	12



Exemplo COO

Vamos executar o código de impressão neste exemplo.

	0	1	2	3
0	3	0	0	15
1	0	0	4	0
2	0	7	0	8
3	0	0		

linhas:

colunas:

valores:

0	0	1	2	2	3
0	3	2	1	3	3
3	15	4	7	8	12



Exemplo COO

Vamos executar o código de impressão neste exemplo.

	0	1	2	3
0	3	0	0	15
1	0	0	4	0
2	0	7	0	8
3	0	0	0	

linhas:

colunas:

valores:

0	0	1	2	2	3
0	3	2	1	3	3
3	15	4	7	8	12



Exemplo COO

Vamos executar o código de impressão neste exemplo.

	0	1	2	3
0	3	0	0	15
1	0	0	4	0
2	0	7	0	8
3	0	0	0	12

linhas:
colunas:
valores:

0	0	1	2	2	3
0	3	2	1	3	3
3	15	4	7	8	12

Exemplo COO

- Qual a complexidade para imprimir os elementos?

Exemplo COO

- Qual a complexidade para imprimir os elementos?
 - Apesar de termos uma representação esparsa da nossa matriz, passamos por todos os índices.

Exemplo COO

- Qual a complexidade para imprimir os elementos?
 - ❑ Apesar de termos uma representação esparsa da nossa matriz, passamos por todos os índices.
 - ❑ Logo temos $O(n^2)$.

Matrizes esparsas (COO)

- Construir uma matriz.
- Obter e alterar elementos da matriz.
- Imprimir seus elementos.
- **Soma de duas matrizes.**
- Produto de uma matriz com um vetor denso.

Matrizes esparsas (COO)

```
MatrizCOO MatrizCOO::Soma (MatrizCOO A, MatrizCOO B) {  
    if (A.l != B.l || A.c != B.c) {  
        throw "Dimensoes das matrizes incompativeis\n";  
    }  
  
    MatrizCOO C = MatrizCOO(A.l, A.c, 0);  
    int kA = 0, kB = 0;  
    while(kA < A.tamanho && kB < B.tamanho) {  
        //Parte 1  
    }  
    while(kA < A.tamanho) {  
        //Parte 2  
    }  
    while(kB < B.tamanho) {  
        //Parte 3  
    }  
    return C;  
}
```


Matrizes esparsas (COO)

```
//Parte 1
```

```
while(kA < A.tamanho && kB < B.tamanho){  
    int iA = A.M[kA][0], jA = A.M[kA][1];  
    int iB = B.M[kB][0], jB = B.M[kB][1];  
    if((iA == iB) && (jA == jB)){  
        C.InsereElemento(iA, jA, A.M[kA][2] + B.M[kB][2]);  
        kA++;  
        kB++;  
    }  
    else if((iA == iB && jA < jB) || (iA < iB)){  
        C.InsereElemento(iA, jA, A.M[kA][2]);  
        kA++;  
    }  
    else{  
        C.InsereElemento(iB, jB, B.M[kB][2]);  
        kB++;  
    }  
}
```

Matrizes esparsas (COO)

```
//Parte 2
```


```
while(kA < A.tamanho){  
    C.InsereElemento(A.M[kA][0], A.M[kA][1], A.M[kA][2]);  
    kA++;  
}
```

```
//Parte 3
```


```
while(kB < B.tamanho){  
    C.InsereElemento(B.M[kB][0], B.M[kB][1], B.M[kB][2]);  
    kA++;  
}
```

Exemplo COO

Vamos efetuar a soma de duas representações.



l:	0	0	1	2	2	3
c:	0	3	2	1	3	3
v:	3	15	4	7	8	12



l:	0	0	2
c:	0	4	1
v:	8	35	2


l:

c:

v:


Exemplo COO

Vamos efetuar a soma de duas representações.



l:	0	0	1	2	2	3
c:	0	3	2	1	3	3
v:	3	15	4	7	8	12


l:	0
c:	0
v:	11



l:	0	0	2
c:	0	4	1
v:	8	35	2


Exemplo COO

Vamos efetuar a soma de duas representações.



l:	0	0	1	2	2	3
c:	0	3	2	1	3	3
v:	3	15	4	7	8	12


l:	0	0
c:	0	3
v:	11	15



l:	0	0	2
c:	0	4	1
v:	8	35	2


Exemplo COO

Vamos efetuar a soma de duas representações.



l:	0	0	1	2	2	3
c:	0	3	2	1	3	3
v:	3	15	4	7	8	12


l:	0	0
c:	0	3
v:	11	15



l:	0	0	2
c:	0	4	1
v:	8	35	2


Exemplo COO

Vamos efetuar a soma de duas representações.



I:	0	0	1	2	2	3
C:	0	3	2	1	3	3
V:	3	15	4	7	8	12


I:	0	0	0
C:	0	3	4
V:	11	15	35



I:	0	0	2
C:	0	4	1
V:	8	35	2


Exemplo COO

Vamos efetuar a soma de duas representações.



l:	0	0	1	2	2	3
c:	0	3	2	1	3	3
v:	3	15	4	7	8	12


l:	0	0	0
c:	0	3	4
v:	11	15	35




l:	0	0	2
c:	0	4	1
v:	8	35	2

Exemplo COO

Vamos efetuar a soma de duas representações.



l:	0	0	1	2	2	3
c:	0	3	2	1	3	3
v:	3	15	4	7	8	12



l:	0	0	2
c:	0	4	1
v:	8	35	2

l:	0	0	0	1
c:	0	3	4	2
v:	11	15	35	4

Exemplo COO

Vamos efetuar a soma de duas representações.



I:	0	0	1	2	2	3
C:	0	3	2	1	3	3
V:	3	15	4	7	8	12



I:	0	0	2
C:	0	4	1
V:	8	35	2

I:	0	0	0	1
C:	0	3	4	2
V:	11	15	35	4

Exemplo COO

Vamos efetuar a soma de duas representações.



l:	0	0	1	2	2	3
c:	0	3	2	1	3	3
v:	3	15	4	7	8	12



l:	0	0	2
c:	0	4	1
v:	8	35	2

l:	0	0	0	1	2
c:	0	3	4	2	1
v:	11	15	35	4	9

Exemplo COO

Vamos efetuar a soma de duas representações.



l:	0	0	1	2	2	3
c:	0	3	2	1	3	3
v:	3	15	4	7	8	12



l:	0	0	2
c:	0	4	1
v:	8	35	2

l:	0	0	0	1	2	2	3
c:	0	3	4	2	1	3	3
v:	11	15	35	4	9	8	12

Exemplo COO

Vamos efetuar a soma de duas representações.

l:	0	0	1	2	2	3
c:	0	3	2	1	3	3
v:	3	15	4	7	8	12

l:	0	0	2
c:	0	4	1
v:	8	35	2

l:	0	0	0	1	2	2	3
c:	0	3	4	2	1	3	3
v:	11	15	35	4	9	8	12

Exemplo COO

- Qual a complexidade para somar duas representações?

Exemplo COO

- Qual a complexidade para somar duas representações?
 - Assuma que as representações possuem respectivamente k_1 e k_2 elementos.

Exemplo COO

- Qual a complexidade para somar duas representações?
 - Assuma que as representações possuem respectivamente k_1 e k_2 elementos.
 - Nesse caso nossa complexidade é $O(\max(k_1, k_2))$.

Matrizes esparsas (COO)

- Construir uma matriz.
- Obter e alterar elementos da matriz.
- Imprimir seus elementos.
- Soma de duas matrizes.
- **Produto de uma matriz com um vetor denso.**

Matrizes esparsas (COO)

```
double* MatrizCOO::Produto(MatrizCOO A, double* v, int n){
    if(A.c != n){
        throw "Dimensoes da entrada incompativeis\n";
    }

    double* s = new double[n];
    for(int i = 0; i < n; i++){
        s[i] = 0;
    }

    for(int i = 0; i < A.tamanho; i++){
        int linha = A.M[i][0];
        int coluna = A.M[i][1];
        s[linha] += A.M[i][2] * v[coluna];
    }

    return s;
}
```


Exemplo COO

Vamos multiplicar uma representação por um vetor denso u e obter um novo vetor denso w .

		u					w				
I:	0	0	1	2	2	3	4	0	0	0	0
C:	0	3	2	1	3	3	3	0	0	0	0
V:	3	15	4	7	8	12	2	0	0	0	0
							1				

Exemplo COO

A linha indica em qual posição de w devemos somar nosso produto.



l :	0	0	1	2	2	3
c :	0	3	2	1	3	3
v :	3	15	4	7	8	12

u	w
4	0
3	0
2	0
1	0


Exemplo COO

A coluna indica com qual elemento de u devemos multiplicar o elemento da nossa representação.

						u	w
						4	0
I:	0	0	1	2	2	3	0
C:	0	3	2	1	3	2	0
V:	3	15	4	7	8	1	0

Exemplo COO

Então $w[0] += 3*4$




I:	0	0	1	2	2	3
C:	0	3	2	1	3	3
V:	3	15	4	7	8	12

u
4
3
2
1

w
12
0
0
0

Exemplo COO

$w[0] += 15 * 1$




I:	0	0	1	2	2	3
C:	0	3	2	1	3	3
V:	3	15	4	7	8	12

u	w
4	27
3	0
2	0
1	0

Exemplo COO

$$w[1] += 4 * 2$$



I:	0	0	1	2	2	3
C:	0	3	2	1	3	3
V:	3	15	4	7	8	12

u	w
4	27
3	8
2	0
1	0

Exemplo COO

$$w[2] += 7 * 3$$



I:	0	0	1	2	2	3
C:	0	3	2	1	3	3
V:	3	15	4	7	8	12

<i>u</i>
4
3
2
1

<i>w</i>
27
8
21
0

Exemplo COO

$$w[2] += 8 * 2$$




I:	0	0	1	2	2	3
C:	0	3	2	1	3	3
V:	3	15	4	7	8	12

<i>u</i>	<i>w</i>
4	27
3	8
2	37
1	0

Exemplo COO

$$w[3] += 12 * 1$$

							u	w
							4	27
							3	8
							2	37
							1	12
l :	0	0	1	2	2	3		
c :	0	3	2	1	3	3		
v :	3	15	4	7	8	12		

Exemplo COO

E por fim, obtemos o vetor w .

							u	w
i :	0	0	1	2	2	3	4	27
c :	0	3	2	1	3	3	3	8
v :	3	15	4	7	8	12	2	37
							1	12

Exemplo COO

- Qual a complexidade para multiplicar uma representação por um vetor?

Exemplo COO

- Qual a complexidade para multiplicar uma representação por um vetor?
 - Note que passamos por cada elemento da representação apenas uma vez. Logo a operação não depende necessariamente do tamanho original da matriz.

Exemplo COO

- Qual a complexidade para multiplicar uma representação por um vetor?
 - Note que passamos por cada elemento da representação apenas uma vez. Logo a operação não depende necessariamente do tamanho original da matriz.
 - Portanto, podemos concluir que a complexidade é $O(k)$.