

# Matriz Esparsa (parte 2)

Prof. Jefferson T. Oliva

Algoritmos e Estrutura de Dados I (AE22CP)  
Engenharia de Computação  
Departamento Acadêmico de Informática (Dainf)  
Universidade Tecnológica Federal do Paraná (UTFPR)  
Campus Pato Branco

- Conversão de Matriz Numérica para Esparsa
- Outro Exemplo de TAD para Matrizes Esparsas

- Matriz Esparsa
  - Matriz bidimensional vs. matriz esparsa
  - Aplicações
  - TAD
    - Vetor de listas encadeadas, sendo cada uma referente a uma linha da matriz
    - Informações de cada elemento da lista encadeada: coluna, referência para o próximo elemento e o valor do item localizado na posição  $(l, c)$
    - Principais operações

## Conversão de Matriz Numérica para Esparsa

# Conversão de Matriz Numérica para Esparsa

- Implementação de uma função para inserção de elementos no início de uma lista encadeada
- Para cada linha da matriz numérica, percorrer do último elemento ao primeiro, utilizando a função sugerida no item anterior

- Inserção no início de uma lista encadeada:

```
static void inserir_primeiro(ListaE *l, int col, int item){  
    Cell *nova = criar_celula(item, col);  
    nova->next = l->head;  
    l->head = nova;  
}
```

# Conversão de Matriz Numérica para Esparsa

- Conversão de uma matriz de números inteiros para uma matriz esparsa:

```
Spa_Mat* converter(int **mat, int l, int c){
    int i, j;
    Spa_Mat* me = criar(l, c);

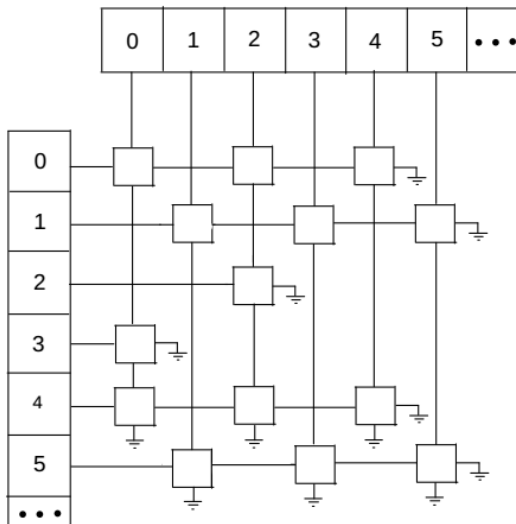
    for (i = 0; i < l; i++)
        for (j = c - 1; j >= 0; j-)
            if (mat[i][j] > 0)
                inserir_primeiro(me->lin[i], c, mat[i][j]);

    return me;
}
```

## Outro Exemplo de TAD para Matrizes Esparsas



## Outro Exemplo de TAD para Matrizes Esparsas



# Outro Exemplo de TAD para Matrizes Esparsas

- Estruturas de dados:

```
typedef struct Cell Cell;  
  
struct Cell{  
    int item, lin, col;  
    Cell *next;  
    Cell *down;  
};  
  
typedef struct ListaE ListaE;  
  
struct ListaE{  
    Cell *head;  
};
```

# Outro Exemplo de TAD para Matrizes Esparsas

- Estruturas de dados:

```
typedef struct Spa_Mat Spa_Mat;  
  
struct Spa_Mat{  
    int n_lin;  
    int n_col;  
    ListaE **lin;  
    ListaE **col;  
};
```

# Outro Exemplo de TAD para Matrizes Esparsas

- Cabeçalho de algumas funções:

```
Spa_Mat* criar(int l, int c);
```

```
Cell* criar_celula(int item, int l, int c);
```

```
int buscar(int item, Spa_Mat* mat);
```

```
void alterar(int item, int l, int c, Spa_Mat* mat);
```

# Outro Exemplo de TAD para Matrizes Esparsas

- Implementações:

```
Spa_Mat* criar(int l, int c){
    Spa_Mat* mat = malloc(sizeof(Spa_Mat));
    int i;

    mat->n_col = c;
    mat->n_lin = l;
    mat->lin = (ListaE**) malloc(sizeof(ListaE*) * l);
    mat->col = (ListaE**) malloc(sizeof(ListaE*) * c);

    for (i = 0; i < l; i++){
        mat->lin[i] = (ListaE*) malloc(sizeof(ListaE));
        mat->lin[i]->head = NULL;
    }

    for (i = 0; i < c; i++){
        mat->col[i] = (ListaE*) malloc(sizeof(ListaE));
        mat->col[i]->head = NULL;
    }

    return mat;
}
```

# Outro Exemplo de TAD para Matrizes Esparsas

- Implementações:

```
Cell* criar_celula(int item, int l, int c){  
    Cell *novo = NULL;  
  
    if (item > 0){  
        novo = (Cell*) malloc(sizeof(Cell));  
        novo->item = item;  
        novo->lin = l;  
        novo->col = c;  
        novo->next = NULL;  
        novo->down = NULL;  
    }  
  
    return novo;  
}
```

# Outro Exemplo de TAD para Matrizes Esparsas

- Implementações:

```
static int validar_ME(int l, int c, Spa_Mat* mat){
    return (Spa_Mat != NULL) && (l >= 0) && (l < mat->lin)
        && (c >= 0) && (l < mat->col);
}

int buscar(int l, int c, Spa_Mat* mat){
    int i, aux = 0;
    Cell *aux;

    if (validar_ME(l, c, mat)){
        aux = mat->lin[l]->head;

        while ((aux != NULL) && (aux->col > c))
            aux = aux->next;

        if ((aux != NULL) && (aux->col == c))
            return aux->item;
    }

    return 0;
}
```

# Outro Exemplo de TAD para Matrizes Esparsas

- Implementações:

```
static Cell* obter_celula_antes_linha(int l, ListaE* col){
    Cell *auxA = NULL;
    Cell *auxP = col->head;

    while ((auxP != NULL) && (auxP->lin < l)){
        auxA = auxP;
        auxP = auxP->down;
    }

    return auxA;
}
```



# Outro Exemplo de TAD para Matrizes Esparsas I

- Implementações:

```
void alterar(unsigned int item, int l, int c, Spa_Mat*
mat){
    Cell *auxLA, *auxLP, *auxCA, *novo;

    if (validar_pos_matriz(l, c, mat)){
        auxLA = NULL;
        auxLP = mat->lin[l]->head;

        while ((auxLP != NULL) && (auxLP->col < c)){
            auxLA = auxLP;
            auxLP = auxLP->next;
        }
    }
}
```

## Outro Exemplo de TAD para Matrizes Esparsas II

```
if (item > 0){
    if ((auxLP != NULL) && (auxLP->col == c))
        auxLP->item = item;
    else{
        novo = criar_celula(item, l, c);
        auxCA = obter_celula_antes_linha(l, mat->col[c]);

        if (auxLA != NULL){
            novo->next = auxLA->next;
            auxLA->next = novo;
        }else{
            novo->next = mat->lin[l]->head;
            mat->lin[l]->head = novo;
        }

        if (auxCA != NULL){
            novo->down = auxCA->down;
            auxCA->down = novo;
        }else{
            novo->down = mat->col[c]->head;
            mat->col[c]->head = novo;
        }
    }
}
```

## Outro Exemplo de TAD para Matrizes Esparsas III

```
    }  
}else if ((auxLP != NULL) && (auxLP->col == c)){  
    auxCA = obter_celula_antes_linha(l, mat->col[c]);  
    auxCA->down = auxLP->down;  
    auxLA->next = auxLP->next;  
  
    free(auxLP);  
}  
}  
}
```



Oliva, J. T.

Matrizes Esparsas. AE22CP – Algoritmos e Estrutura de Dados I.

*Notas de Aula.* Engenharia de Computação.

Dainf/UTFPR/Pato Branco, 2020.



Roman, N. T.; Digiampietri, L. A.

Matriz Esparsa. ACH2023 – Algoritmos e Estrutura de Dados I.

*Notas de Aula.* Sistemas de Informação. EACH/USP/São

Paulo, 2018.