

MC202 (Estruturas de Dados) - 1s2021

Tarefa 05: Listas simplesmente ligadas e variações

Prof^o Ricardo Dahab

Assistente: Elisa Dell'Arriva

Assistente: Jônatas Trabuco Belotti

Instituto de Computação - UNICAMP

Sobre a Tarefa

O objetivo desta tarefa é que o(a) aluno(a) se familiarize com o uso de listas ligadas, em especial, listas circulares. O exercício consiste em fazer a leitura, escrita e soma de matrizes quadradas esparsas.

Representação de Matrizes Esparsas

Dizemos que uma matriz é *esparsa* se apenas uma pequena fração de seus elementos é diferente de zero. De forma a evitar desperdício de memória, representamos apenas seus elementos não nulos. Como a quantidade de elementos não nulos é desconhecida, utilizamos listas ligadas para representá-los. De modo que:

- cada linha e cada coluna é representada por uma lista ligada circular, com nó cabeça, dos seus elementos não nulos;
- cada elemento é representado por uma *struct* com os seguintes campos:

```
typedef struct elemento {
    int valor;
    int linha;
    int coluna;
    struct elemento *prox_na_linha;
    struct elemento *prox_na_coluna;
}Elemento;
```

Assim, `prox_na_linha` aponta para o próximo elemento da linha `linha` e `prox_na_coluna` aponta para o próximo elemento da coluna `coluna`.

- Os nós cabeça das linhas formam uma lista ligada circular com nó cabeça. O campo `linha` desses nós cabeça contém o número da respectiva linha, já o campo `coluna` desses nós contém o valor -1 . O mesmo acontece com os nós cabeça das colunas e os seus campos `coluna` e `linha`. Essas duas listas compartilham o mesmo nó cabeça, a que chamamos de *super nó cabeça*. Os campos `linha` e `coluna` desse super nó contém os números de linhas e de colunas da matriz.

A Figura 1 ilustra a representação em listas da seguinte matriz esparsa:

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ -1 & 1 & 0 \end{bmatrix}$$

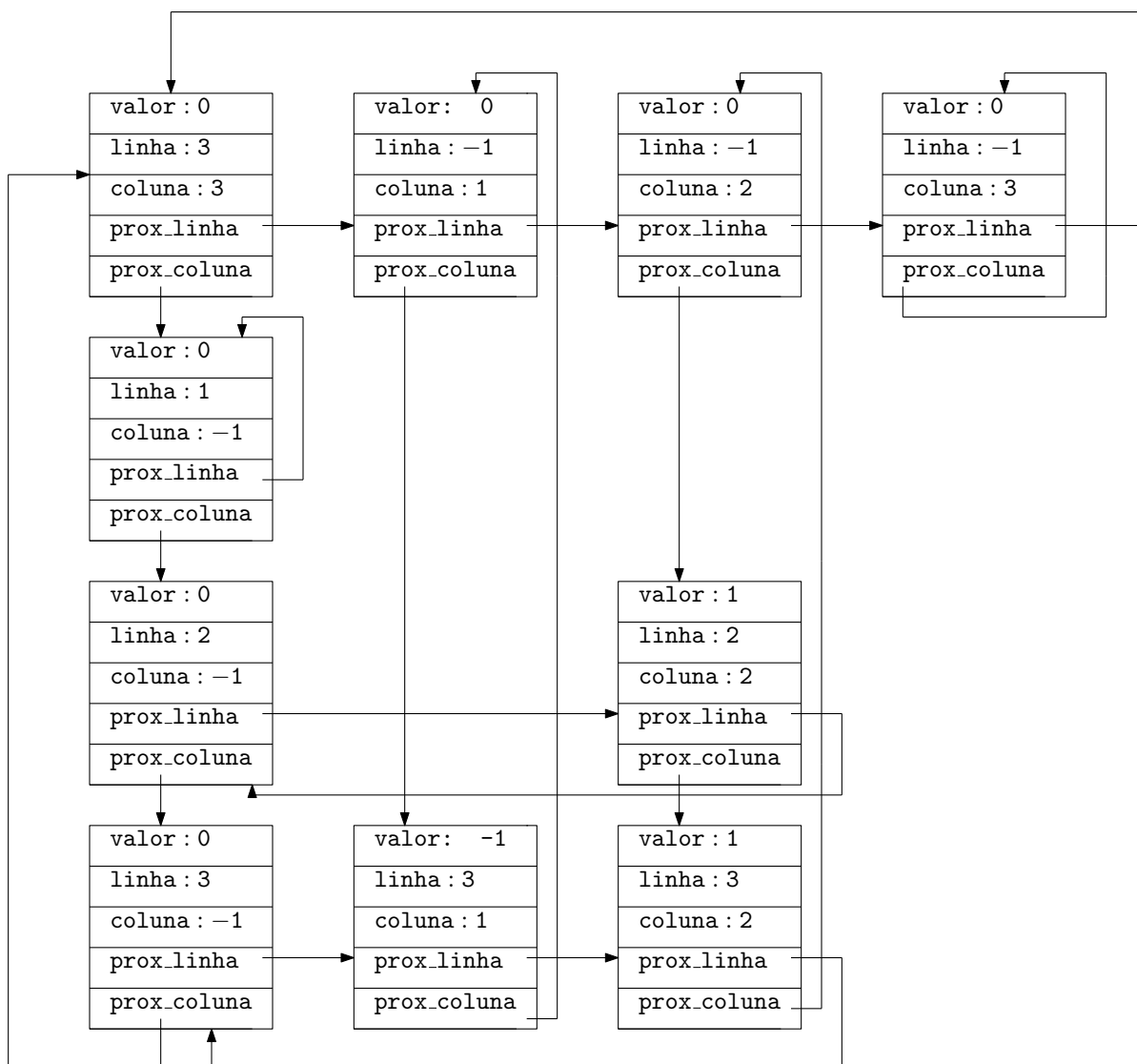


Figura 1: Representação em listas ligadas circulares, com nó cabeça, de uma matriz esparsa 3×3 .

Formato da entrada e saída

As matrizes desta tarefa serão sempre quadradas de ordem n , $1 \leq n \leq 50$.

A primeira linha da entrada contém um inteiro n entre 1 e 50, que representa a ordem da matriz. A seguir, virão as linhas com os valores não nulos da primeira matriz no seguinte formato:

`<número_linha> <número_coluna> <valor_elemento>`

Os valores referentes a primeira matriz se encerram quando for lida a linha: `-1 -1 -1`.

Em seguida virão as linhas referentes aos elementos não nulos da segunda matriz. Também no formato:

`<número_linha> <número_coluna> <valor_elemento>`

Mais uma vez os valores se encerram quando for lida a linha: `-1 -1 -1`.

Após realizar a leitura e o armazenamento das duas matrizes, seu programa deve criar uma terceira matriz para armazenar o resultado da soma das duas primeiras. Como saída você deve imprimir a matriz resultante, com um espaço entre cada elemento. O último caractere da saída deve obrigatoriamente ser um `\n`.

Exemplo 1.

Entrada:

```
4
1 1 1
2 4 -1
4 2 1
-1 -1 -1
2 2 -1
3 3 -1
4 1 1
-1 -1 -1
```

Saída:

```
1 0 0 0
0 -1 0 -1
0 0 -1 0
1 1 0 0
```

Exemplo 2.

Entrada:

```
5
1 1 1
3 3 1
5 5 1
-1 -1 -1
1 1 1
1 5 1
3 3 -1
5 1 1
5 5 1
-1 -1 -1
```

Saída:

```
2 0 0 0 1
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
1 0 0 0 2
```

Importante

- A representação de matriz esparsa em listas deve, obrigatoriamente, seguir a especificação descrita neste enunciado.
- O resultado da soma entre as duas matrizes de entrada deve, obrigatoriamente, ser armazenado em uma terceira matriz.
- É obrigatório liberar toda memória alocada dinamicamente. Para fazer essa verificação, recomendamos o uso da ferramenta *valgrind*.

O não cumprimento de qualquer uma dessas regras implicará em nota zero nesta tarefa.

Alguns avisos e lembretes

- A página da disciplina no SuSy é <https://susy.ic.unicamp.br:9999/mc202abc>.
- Para submeter, utilizem somente os dígitos numéricos do RA e a senha da DAC.
- O número máximo de submissões é 20.
- Esta tarefa tem peso 2 (dois).
- A tarefa estará aberta até 31/05/2021, às 23h59.