Universidade Federal de Goiás – UFG
Instituto de Informática – INF
Bacharelados (Núcleo Básico Comum)

Algoritmos e Estruturas de Dados 1 – 2022/2

Torneio de Resolução de Problemas 03 - Listas lineares

Turma: INF0286 – Prof. Cedric Luiz de Carvalho

01 Considere um vetor de números inteiros v, cujo tamanho é expresso pela constante inteira positiva n_v, com $n_v \in \mathbb{N}^*$. Este vetor é utilizado para representar uma lista linear de números inteiros $\mathscr{L}$, cuja quantidade de *nós* atual é sempre armazenada na variável global $n_{\mathscr{L}}$. Obviamente $n_{\mathscr{L}} \leq n$.
Escreva um programa que seja capaz de manipular a lista $\mathscr{L}$ por meio de funções que realizem as seguintes operações:

1. criar uma lista inicialmente vazia;
2. inserir um número inteiro x no final da lista;
3. inserir um número inteiro x no início da lista;
4. remover o número inteiro que está na última posição da lista;
5. remover o número inteiro que está na primeira posição da lista;
6. apresentar, no dispositivo de saída padrão do sistema computacional, todos os números presentes na lista, do primeiro ao último;
7. apresentar a quantidade de números inteiros existentes na lista;
8. apresentar o número inteiro que está no início da lista;
9. apresentar o número inteiro dque está na última posição da lista.

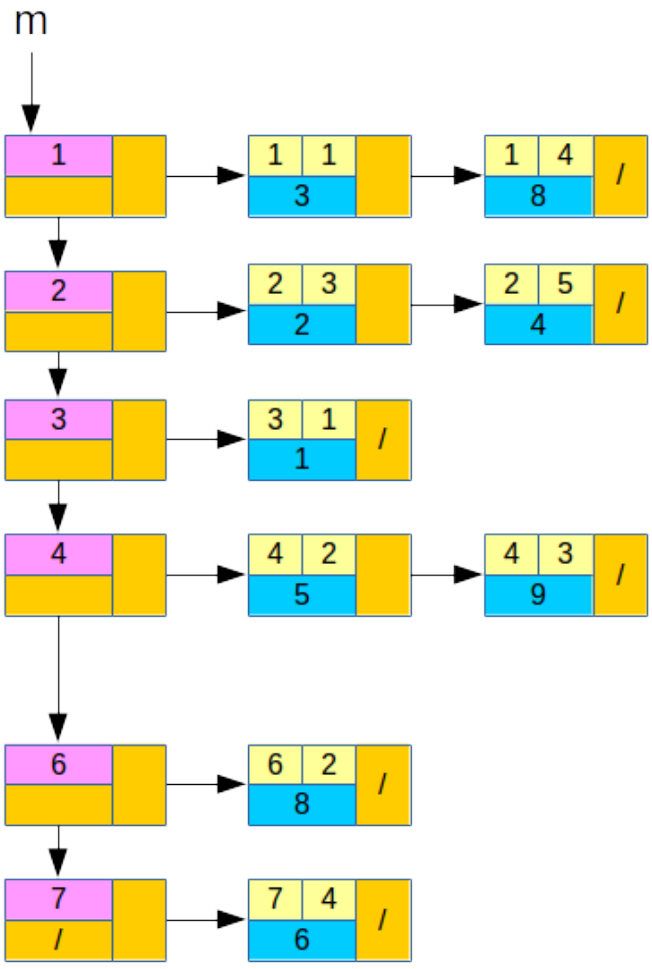
Observação: Todas as funções devem prever a possibilidade da ocorrência de *falhas* durante sua operação. Por exemplo: ao tentar inserir um novo número inteiro na lista, pode acontecer de não haver mais espaço disponível para armazená-lo. Nestas situações, a função deverá retornar o valor (-1) (menos um) como indicador de que algum tipo de erro ocorreu. Se houver sucesso na realização da operação, a função deverá retornar 0 (zero).

02 Quando estudamos a estrutura de dados *matriz*, vimos que uma matriz é dita *esparsa* quando possui uma grande quantidade de elementos "*nulos*" (ou seja, quando os elementos são numéricos, normalmente iguais a 0 – zero).
Sabe-se que matrizes esparsas têm aplicações diversas na Engenharia e na Física (por exemplo, na resolução de malhas de circuitos elétricos), na Matemática (resolução de sistemas de equações lineares), na Biologia (representação de propriedades de cadeias de DNA ou RNA) e na Computação (armazenamento de dados em planilhas de cálculo eletrônicas).

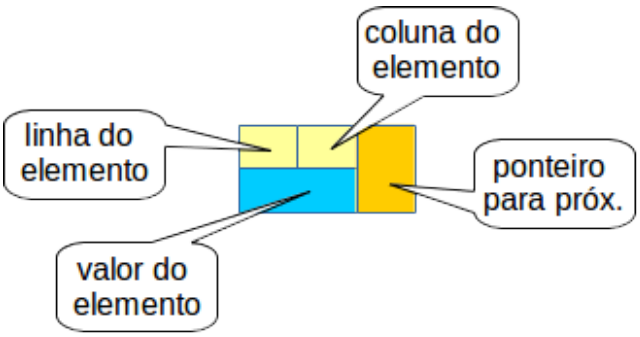
Uma das maneiras de representa uma matriz esparsas é utilizar um conjunto de *listas lineares encadeadas* que apontem para os elementos da matriz que são diferente de 0 (zero). Por exemplo: Seja a matriz M, de ordem 7 por 5, a seguir:

$$M = \begin{bmatrix} 3 & 0 & 0 & 8 & 0 \\ 0 & 0 & 2 & 0 & 4 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 5 & 9 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 6 & 0 \end{bmatrix}$$

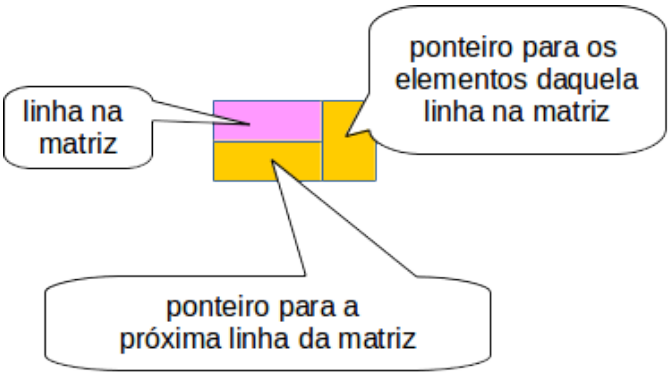
Cada lista linear encadeada representa uma das "*linhas*" da matriz M, havendo também uma "*lista*" em que cada elemento aponta para o primeiro elemento das listas anteriores. Veja uma representação da matriz esparsa anterior utilizando esta proposta:



Na figura anterior, cada uma das "linhas" representa uma linha da matriz, onde somente os elementos não nulos estão armazenados. Assim, o elemento que está na posição $(1,1)$ da matriz – que vale 3 – está representado pelo *nó* $[1,1,3]$, e aponta para o elemento que está na posição $(1,4)$ – que vale 8 (oito) – e está representado por $[1,4,8]$. Cada *nó* que representa um elemento da matriz tem a seguinte estrutura:



Observando a vertical (lateral esquerda), tem-se uma lista que indica cada uma das "linhas" da matriz que possuem elementos não nulos nela. Por isso estão indicadas as linhas de números $1, 2, 3, 4, 6$ e 7, já que na linha 5 todos os elementos são nulos. Cada *nó* desta lista tem a seguinte estrutura:



Portanto, o entrelaçamento destas duas listas lineares ligadas representa a matriz esparsa $\mathscr{M}$ de maneira eficiente quanto ao uso de armazenamento.
Você deve:

(a) Criar um programa que seja capaz de manipular matrizes esparsas utilizando a representação descrita anteriormente;

(b) O programa deve permitir a leitura dos elementos não nulos da matriz a partir do dispositivo de entrada padrão do sistema computacional, bem como a impressão da matriz de maneira tabular (ou seja, com elementos dispostos em linhas e colunas, apresentando também os elementos nulos da matriz);

(c) O programa deve permitir a realização da operação de atribuição de um determinado valor v a uma certa posição (i, j) especificada;

(d) O programa deve permitir a consulta a uma certa posição (i, j) especificada.

03 Considere uma pilha $\mathscr{P}$, de números inteiros, que será representada por uma *lista linear simplesmente encadeada* (LLSE) p e, portanto, devem estar disponíveis as seguintes operações:

1. `CriarPilha(p)`: cria a pilha p, inicialmente vazia;
2. `Empilhar(p, x)`: Empilha, na pilha p, o número inteiro x;
3. `Desempilhar(p)`: Retorna o número inteiro x que está no *topo* da pilha p, retirando-o desta pilha;

4. `estaVazia(p)`: Retornar `verdadeiro` se a pilha *p* está vazia. Do contrário, retorna `falso`.

Escreva um programa que implemente as funções anteriormente definidas e, a partir delas, outra função denominada `removeChave(p, x)` que deverá remover o número inteiro *x* da pilha *p*, mas utilizando somente operações de *empilhar* e *desempilhar*. Ao final da execução desta função, a pilha *p* deverá ser idêntica à originalmente recebida por ela, exceto pela ausência do número inteiro *x* cuja remoção foi solicitada.
Observação: Considere que não há repetição de números na pilha *p*.

04 Uma *fila* é dita ser uma "*fila por prioridade*" quando os elementos são nela inseridos não no final (e removidos no início) mas, sim, de acordo com uma informação que determina a prioridade do elemento sendo inserido: quanto mais prioritário, mas próximo do início da fila o elemento é inserido. O elemento de maior prioridade ocupa a primeira posição da fila.
Assim, considere uma fila $\mathscr{L}$ em que cada um de seus elementos possui um número inteiro positivo $p \in [0, 100]$ para representar a prioridade daquele elemento, sendo 0 (zero) a maior prioridade possível e 100 a menor.
Construa um programa que seja capaz de solicitar ao usuário que forneça *n* elementos (*n* deverá ser lido previamente, sendo que $1 \leq n \leq 100$) para a fila $\mathscr{L}$, com prioridades diversas (inclusive com a possibilidade de repetição de prioridades). Depois disso, apresente a lista $\mathscr{L}$ do primeiro ao último elemento.
Observação: Caberá a você decidir qual será a forma de representação que empregará para implementar $\mathscr{L}$.

5 Desenvolva uma aplicação que permitia manipular números inteiros de grande extensão: até 300 (trezentos) dígitos por número. Para representar cada número utilize uma *lista linear encadeada* para representar cada um dos números inteiros. Você deverá implementar as operações de soma, subtração, multiplicação e divisão inteira entre dois números. Evidentemente é necessário que estes números possam ser *lidos* e *impressos*. Outra operação desejada é saber se um dado número é, ou não, *primo*.
Observação: Fica a seu critério escolher detalhes sobre a representação por meio de lista linear encadeada: simples, dupla, com ou sem nó descritor, etc.