
FILAS SEQUENCIAIS

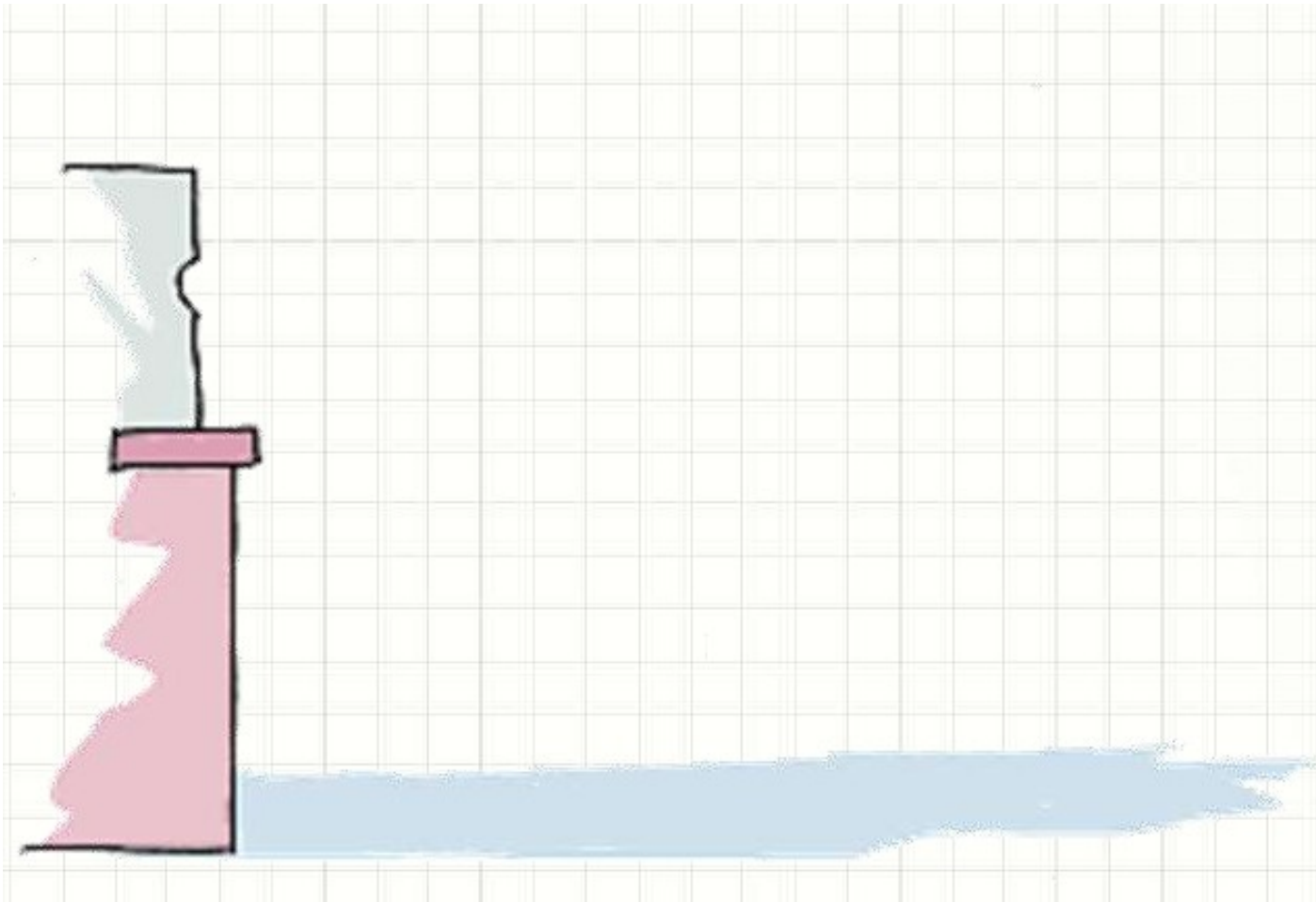
Filas Sequenciais

ORDEM DE ENTRADA
X
ORDEM DE SAÍDA

Filas Sequenciais

Ordem de Entrada x Saída

- É comum: ordem de entrada → saída
- Exemplo: fila do banco



Filas Sequenciais

Ordem de Entrada x Saída

- É comum: ordem de entrada → saída
- Exemplo: fila do banco



Filas Sequenciais

Ordem de Entrada x Saída

- É comum: ordem de entrada → saída
- Exemplo: fila do banco



Filas Sequenciais

Ordem de Entrada x Saída

- É comum: ordem de entrada → saída
- Exemplo: fila do banco



Filas Sequenciais

Ordem de Entrada x Saída

- É comum: ordem de entrada → saída
- Exemplo: fila do banco



Filas Sequenciais

Ordem de Entrada x Saída

- É comum: ordem de entrada → saída
- Exemplo: fila do banco



Filas Sequenciais

Ordem de Entrada x Saída

- É comum: ordem de entrada → saída
- Exemplo: fila do banco



Filas Sequenciais

Ordem de Entrada x Saída

- É comum: ordem de entrada → saída
- Exemplo: fila do banco



Filas Sequenciais

Ordem de Entrada x Saída

- É comum: ordem de entrada → saída
- Exemplo: fila do banco



Filas Sequenciais

Filas

- Estrutura de dados Fila: Lista FIFO
 - FIFO: First In, First Out
 - Primeiro a entrar... É o primeiro a sair
- Inserir: sempre no fim da lista (fim da fila)
 - Remover: sempre do início da lista (início da lista)
 - Isso é útil em software?
- Sim, em muitos casos!

Filas Sequenciais

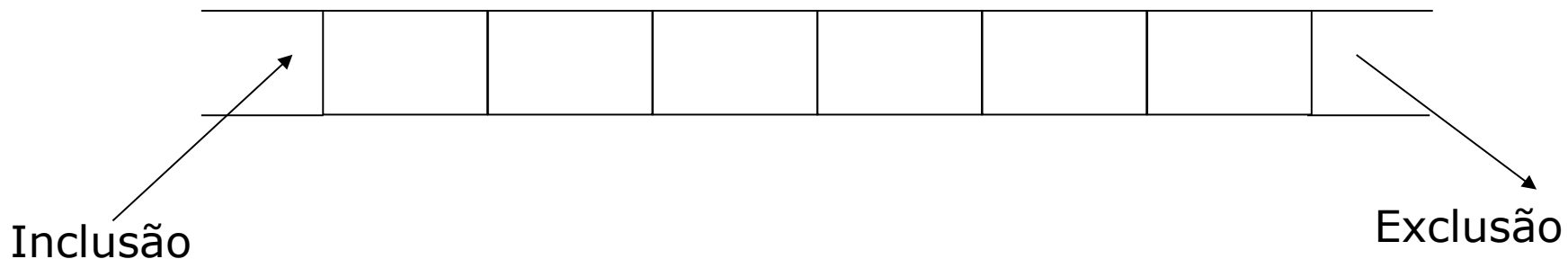
Funcionamento de Filas

- É imposto um critério que restringe a inserção e remoção de elementos que compõem um conjunto de dados.
- Critérios
 - LIFO (Last In First Out)
 - Pilhas
 - FIFO (First In First Out)
 - Filas

- **Filas – Operações básicas**
 - Criar uma Fila
 - Inserir um elemento
 - Remover um elemento
 - Consultar o primeiro elemento da Fila
 - Listar todos os elementos

Fila - vetor

- Representação seqüencial
 - Dois apontadores: frente (cabeça) e fim (cauda)
 - Array com m posições
 - Frente=Fim=0 → Fila vazia



Fila

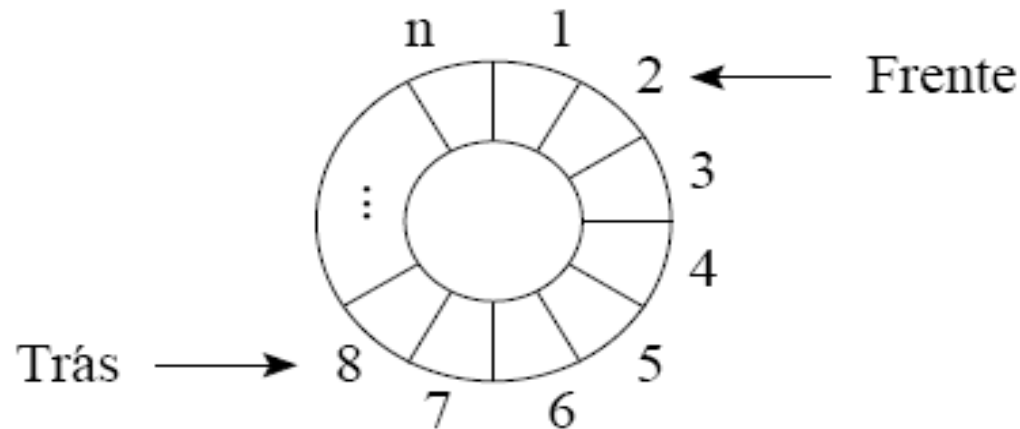
- O valor de n pode se tornar igual a m e impedir que novas inserções ocorram, apesar da fila conter poucos elementos.
- Inserção
 - Expansão da parte de trás
- Remoção
 - Contração da parte da frente

Fila

- Fila tende a caminhar pela memória ocupando a parte de trás e descartando espaço na parte da frente.
- Pode alcançar o limite de memória com poucas inserções e remoções.

Fila circular

- O elemento seguinte a $\text{fila}[m]$ é $\text{fila}[1]$
- Fim (Trás) \rightarrow Indexação da próxima posição na qual um elemento recém-chegado será inserido na fila



Fila circular

- Enqueue (Q, x) (*Sem teste de overflow*)

Q[Fim]:=x;

If Fim = comprimento[Q] then

Fim=1;

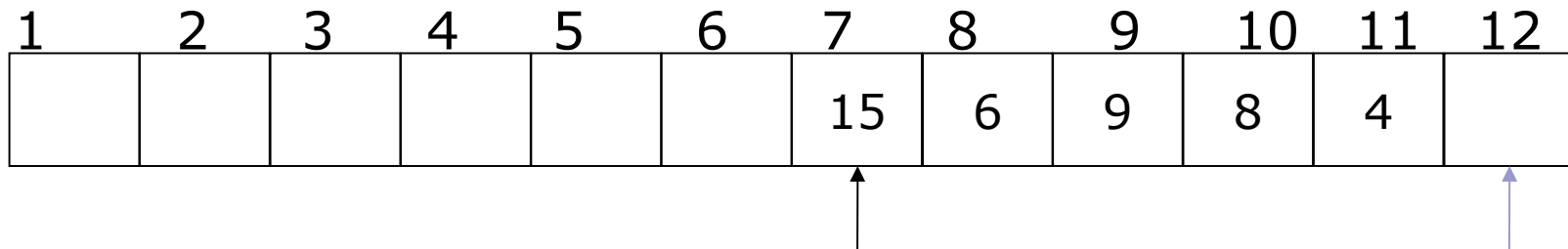
else

Fim:=Fim+1;

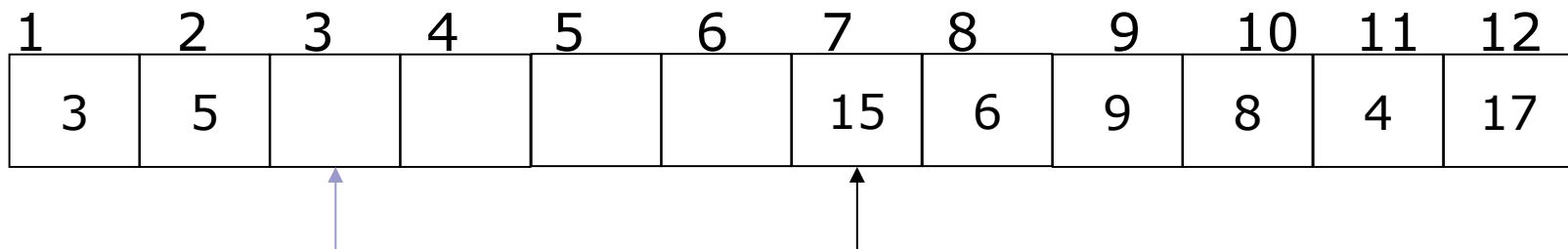
Fila circular

- Dequeue (Q)
 $x := Q[\text{inicio}]$;
 If $\text{inicio} = \text{comprimento}[Q]$ then
 $\text{inicio} := 1$
 else
 $\text{inicio} := \text{inicio} + 1$;
 return x ;

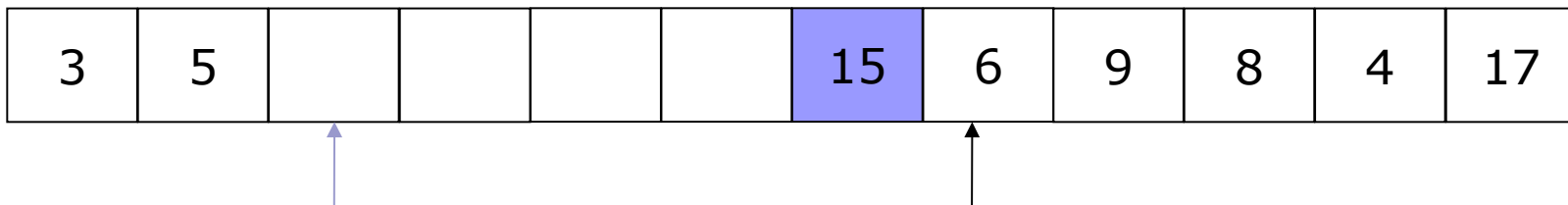
Fila circular



17,3,5

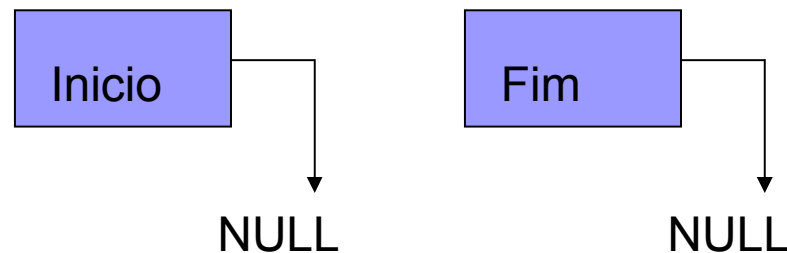


dequeue



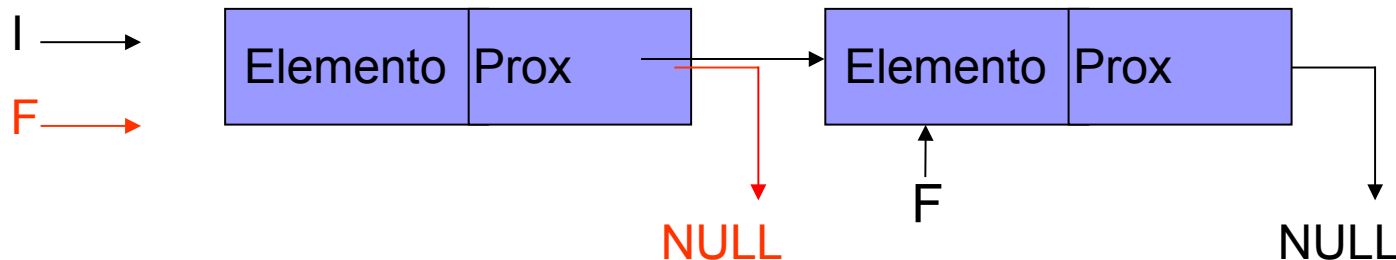
Fila encadeada - Operações básicas

- Criar uma fila
 - Declarar dois ponteiros
 - Inicio
 - Fim



Fila encadeada - Operações básicas

- **Inserir um elemento**
 - Criar o nó com o elemento, apontando para NULL (ou NIL)
 - Se a fila estiver vazia:
 - Os ponteiros início e fim apontam para o novo nó
 - Se a fila não está vazia
 - O último elemento da fila aponta para o novo nó
 - O ponteiro Fim aponta para o novo nó



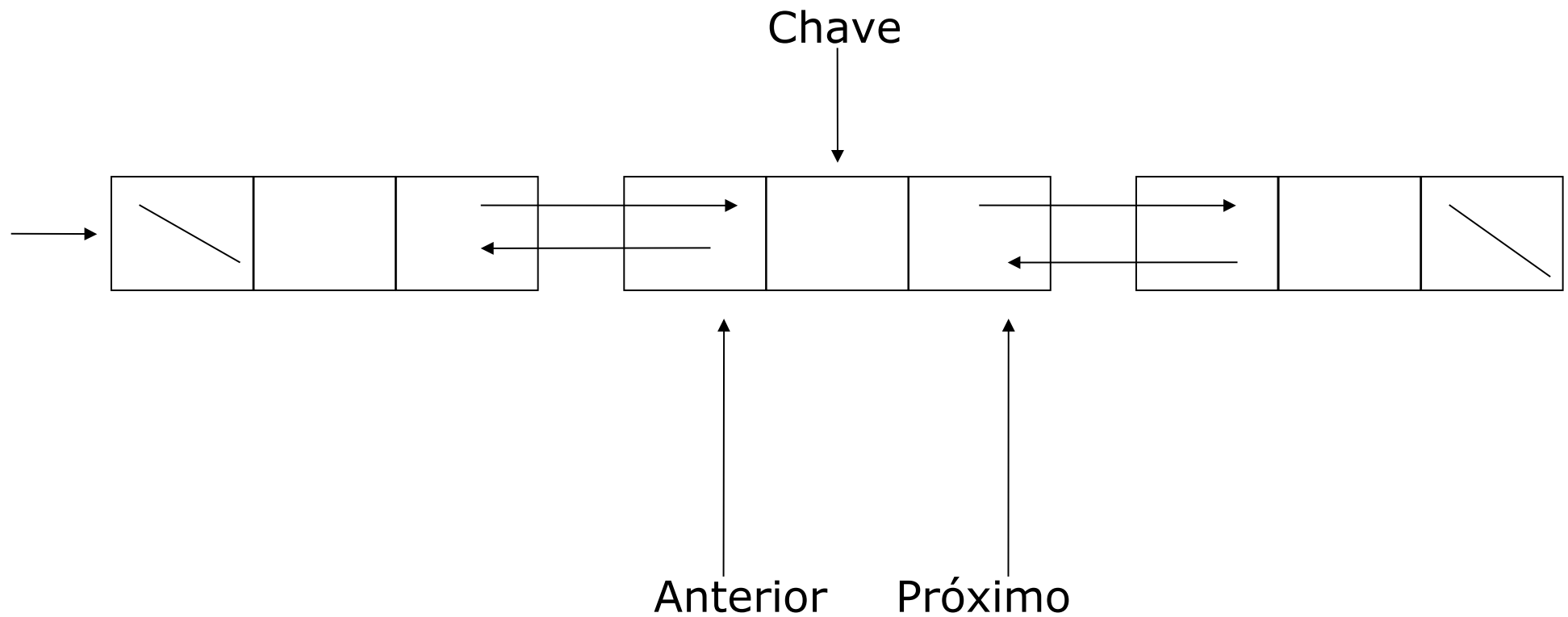
Lista duplamente encadeada

- Lista duplamente ligada
- Encadeamento duplo
- Cada nó da lista contém
 - Dado (elemento)
 - Endereço do nó anterior
 - Endereço do próximo nó

Lista duplamente encadeada

- Se o anterior de um nó é nil
 - O nó é o primeiro elemento da lista
- Se o próximo é nil
 - O nó é o último da lista

Lista duplamente encadeada



Lista duplamente encadeada

- Insert(L,x)
 proximo[x] ← início[L]
 if início[L] <> NIL then
 anterior[início[L]] ← x
 início[L] ← x
 anterior[x] ← NIL
(*insere na “frente” da lista*)