Filas

Claudia Akemi Izeki

Filas

- **Filas** (*queues*) são listas lineares nas quais a movimentação dos elementos é realizada apenas em posições especiais: a **primeira** e a **última**.

- Exemplos: filas de banco, filas de caixa de supermercado, filas de cinema, etc.

- A fila, como a pilha, é conceitualmente uma **estrutura dinâmica** que está continuamente mudando, pois itens são adicionados/retirados.

Política de Acesso aos Dados

- A fila é a aplicação de uma técnica de programação denominada de **FIFO** (*First In First Out*), ou seja,

"**os primeiros elementos da fila são os primeiros a sair**".

Algumas aplicações

- Listas de espera em geral:
  - Reservas em sistemas de bibliotecas, etc.
  - Gerenciamento de fila de impressão

TAD Fila

- $Cria(F)$
  - Cria uma fila F vazia
- $Enfileira(x, F)$: *enqueue*
  - Insere x no final de F
- $Desenfileira(F)$: *dequeue*
  - Remove e retorna o elemento do início da fila
- $Vazia(F)$
  - Testa se a fila F está vazia
- $Primeiro(F)$
  - Acessa o elemento do início da fila
- $Destrói(F)$
  - Destrói a estrutura e libera o espaço ocupado por ela

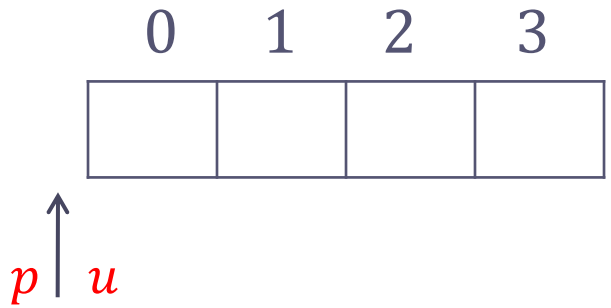
Fila estática

- Existe uma forma de se utilizar um *array* (vetor) na implementação de uma fila?
- Sim, se houver dimensionamento do *array* que dê para acomodar o tamanho máximo da fila e, além disso, precisa-se dos ponteiros início (primeiro) e fim (último).

- Exemplo:

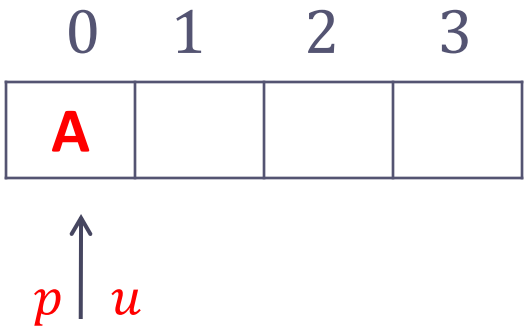
Situação 1

Inicial: fila vazia



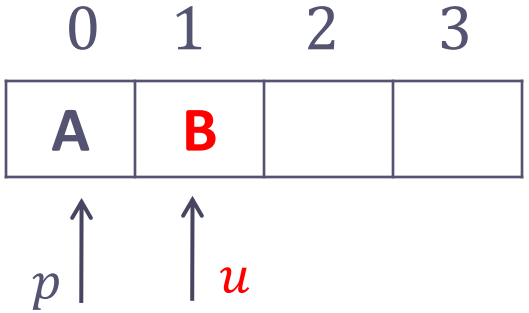
Situação 2

Inserir informação A



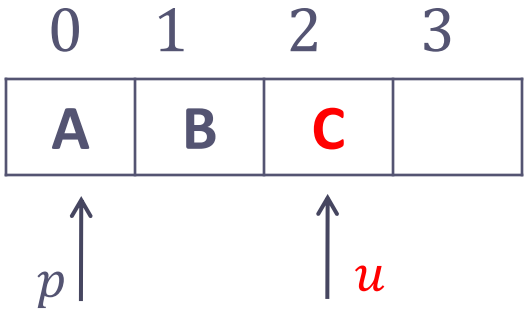
Situação 3

Inserir informação B



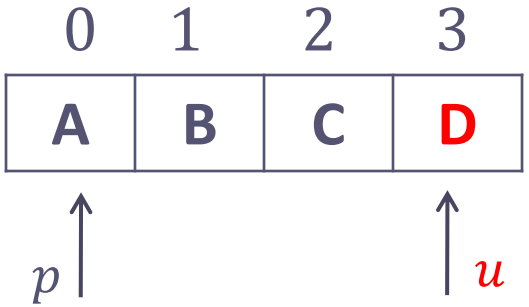
Situação 4

Inserir informação C



Situação 5

Inserir informação D



Situação 6

Inserir informação E

Overflow!

Situação 7

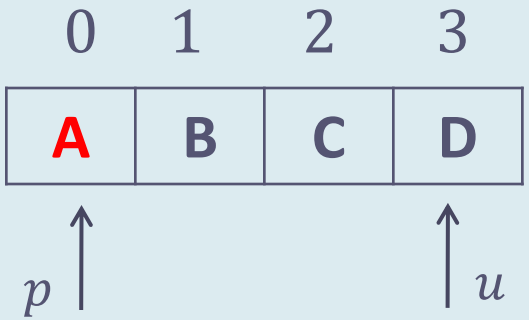
Retirar informação (A)

?

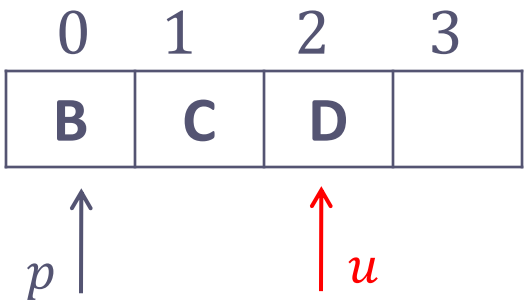
- Desloco os $n - 1$ elementos uma posição à esquerda? *ou*
- Movo o ponteiro do início para frente?

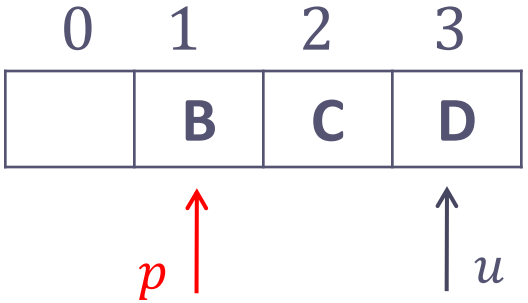
Situação 7

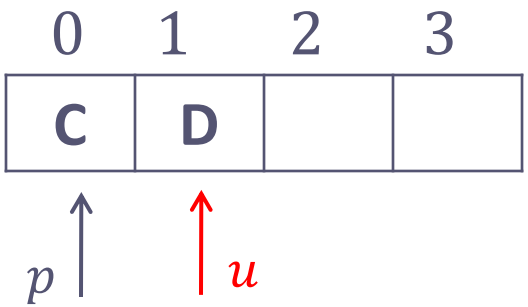
Retirar informação (A)



2 formas

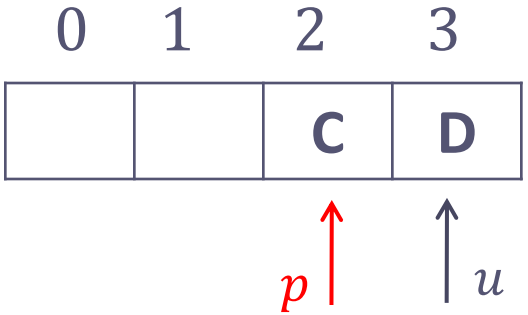


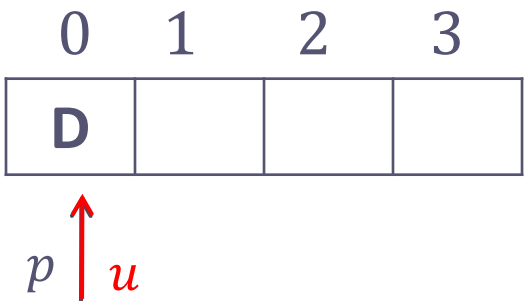




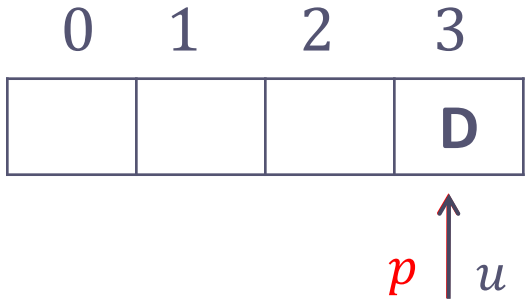
Situação 8

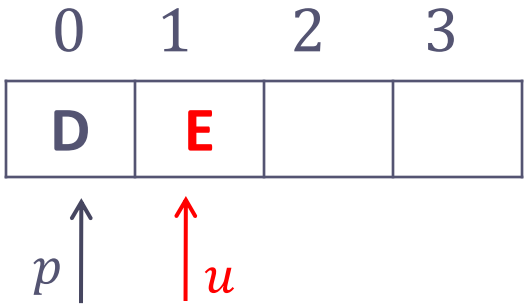
(*Removendo B*)



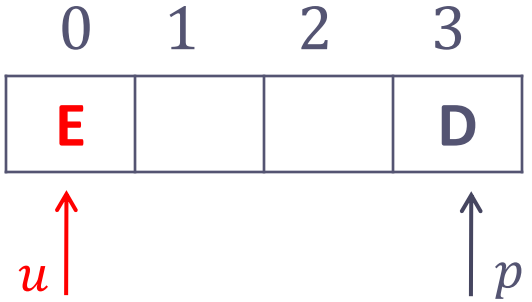


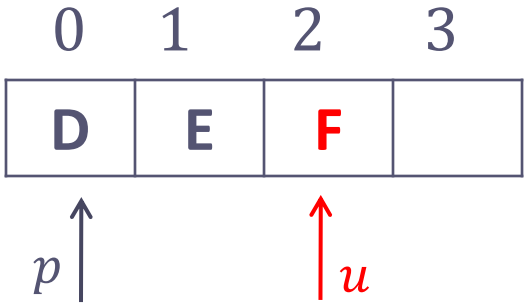
Situação 9
(*Removendo C*)



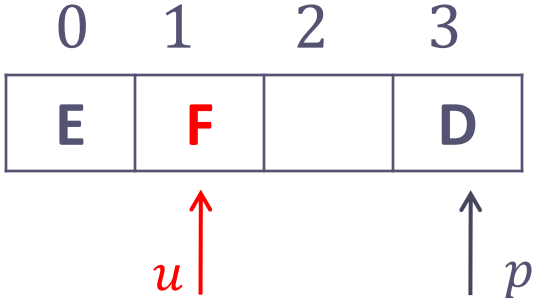


Situação 10
(Inserindo E)





Situação 11
(Inserindo F)



Fila circular

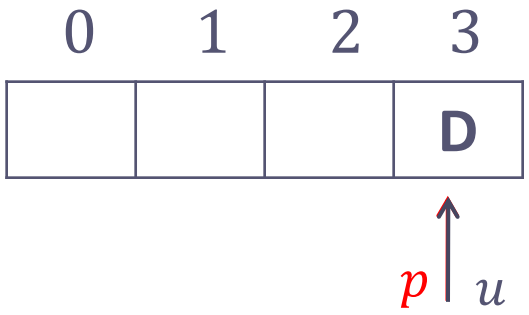
Como podemos implementar uma fila como uma lista linear estática?

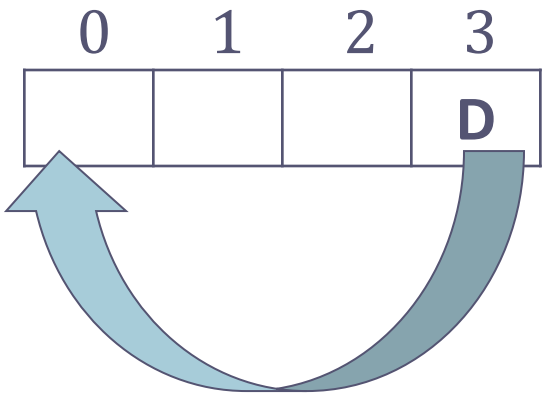
1. O ponteiro **início** sempre na posição 0 para $N > 0$. Daí, na remoção, devem-se deslocar todos os elementos uma posição à esquerda. **Isso não é nada eficiente!!!**

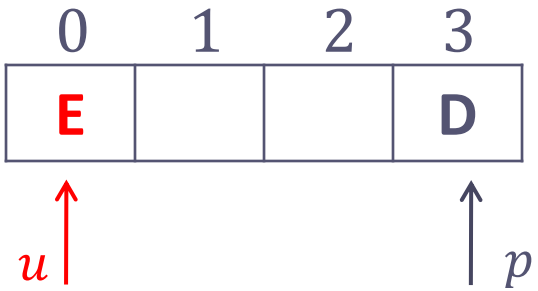
2. Como uma fila circular, não necessitando de quaisquer deslocamentos na operação de remoção.

Observação: em qualquer uma das implementações acima, supor inicialmente ponteiros início e fim com valor -1.

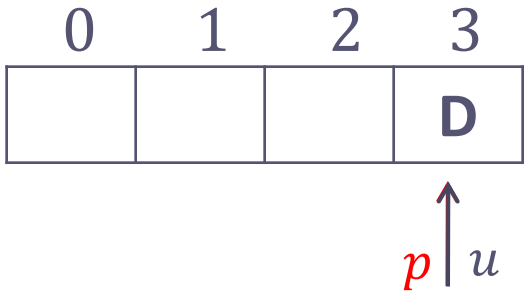
Fila circular estática

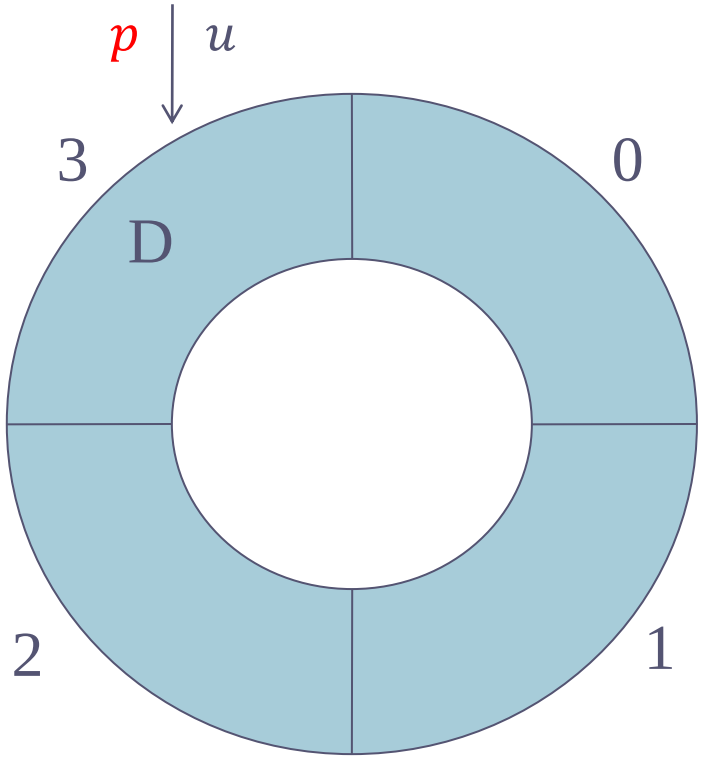




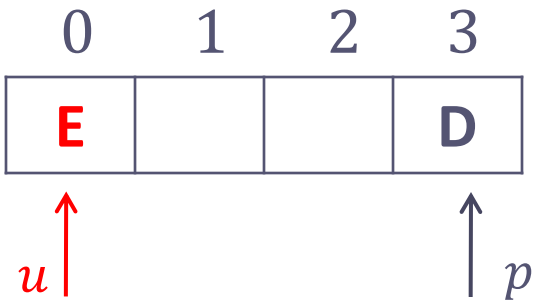


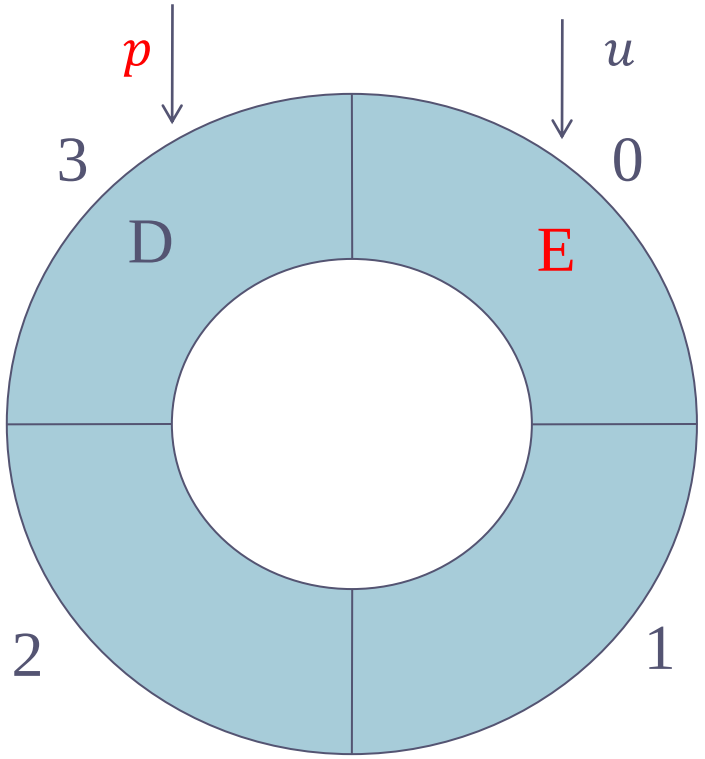
Fila circular estática





Fila circular estática





Implementação estática: Fila circular

```
/* Definição da Estrutura de Dados */
class Fila{
  private:
     int inicio;
     int fim;
     int vet[N];
      // outros campos...
};
```

Atividade 1: implementar cada uma das operações do TAD Fila.

```
/* Algoritmo para enfileirar o elemento */

   nova_posição = (fim+1) mod MAX

   SE nova_posição igual inicio ENTÃO
       OVERFLOW!
   SENÃO
       fim = nova_posição
       fila[fim] = x

       SE inicio igual -1 ENTÃO
             inicio=0
```

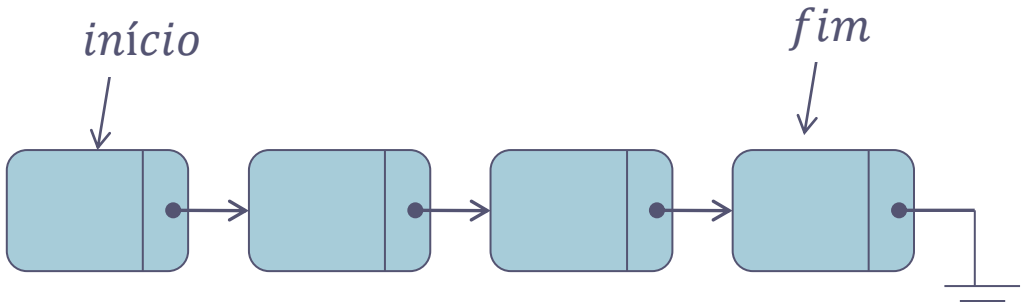
mod é o operador resto (%) em C++!

```
/* Algoritmo que desenfileira (remove) o elemento */

Se a fila está vazia então // inicio igual a -1
  UNDERFLOW!
Senão
  obtém o elemento do início da fila
  Se os ponteiros inicio e fim são iguais então
     a fila fica vazia // inicio e fim recebem -1
  Senão
     // calcula o ponteiro inicio, respeitando a circularidade
     inicio = (inicio+1) mod MAX
   retorna o elemento removido
```

Implementação Dinâmica

- Ponteiros para início e fim da lista
- Uso de uma lista simplesmente encadeada



Implementação dinâmica: Lista simplesmente encadeada com ponteiros para o início e fim

```
/* Definição da Estrutura de Dados */
class No{
  private:
     int chave;
     No* prox;
     // outros campos...
};
class Fila{
  private:
     No* inicio;
     No* fim;
     // outros campos...
};
```

Atividade 2: implementar cada uma das operações do TAD Fila.

Análise dos dois tipos de representação

- Vantagens da Fila estática (anel):
  - Não envolve custos da alocação dinâmica
- Desvantagens da Fila Estática:
  - Previsão de tamanho máximo
- Vantagens da Fila Dinâmica:
  - Ocupa espaço estritamente necessário
- Desvantagens da Fila Dinâmica:
  - Custos usuais da alocação dinâmica (tempo de alocação, campos de ligação)

Quando usar

- Representação estática:
  - Quando fila tiver tamanho pequeno ou seu comportamento for previsível
- Representação dinâmica:
  - Nos demais casos

Bibliografia

- NUNES, M. G, SCC-0202 - Algoritmos e Estruturas de Dados I do ICMC-USP, 2009.

- SZWARCFITER, J. L.; MARKENZON, L. Estruturas de Dados e Seus Algoritmos. Rio de Janeiro: LTC, 1994.