ED62A-COM2A
ESTRUTURAS DE DADOS

Aula 04A - Filas
Implementação estática

Prof. Rafael G. Mantovani

05/04/2019

UTFPR
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Apucarana - PR, Brasil
Abril, 2019

Universidade Tecnológica Federal do Paraná (UTFPR)
Engenharia de Computação

Roteiro

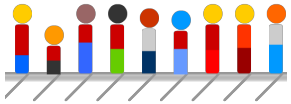
1. Introdução
2. Filas
3. Operações gerais
4. Inserção de elementos
5. Remoção de elementos
6. Referências

Roteiro

1. Introdução
2. Filas
3. Operações gerais
4. Inserção de elementos
5. Remoção de elementos
6. Referências

Introdução

Pilhas

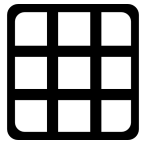


Filas

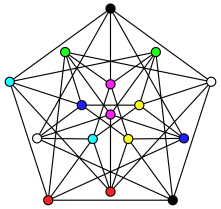
Árvores

Problema

Vetores



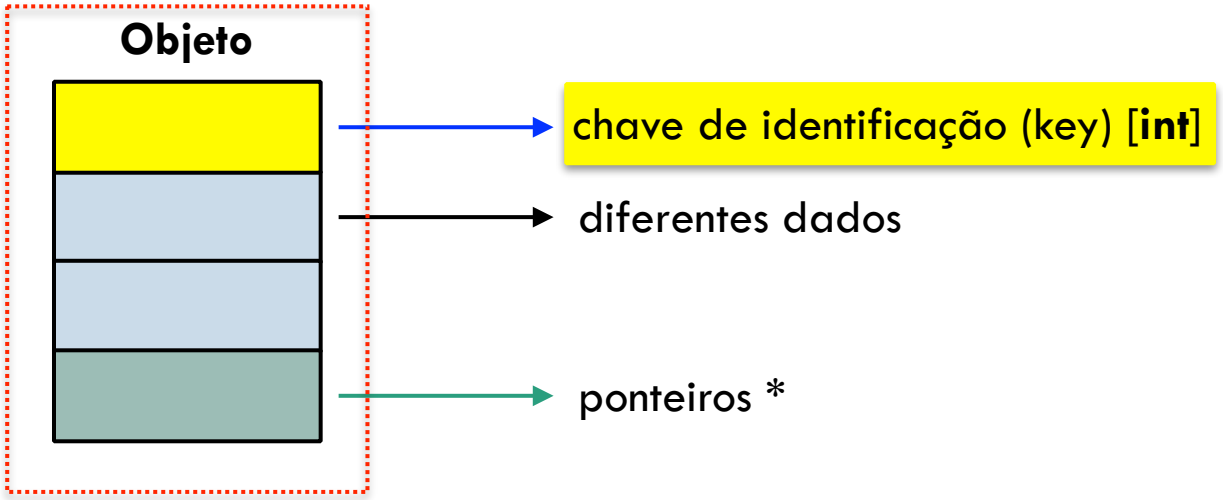
Grafos



Etc

Introdução

- Elemento (objeto) → vários atributos



Roteiro

1. Introdução
2. Filas
3. Operações gerais
4. Inserção de elementos
5. Remoção de elementos
6. Referências

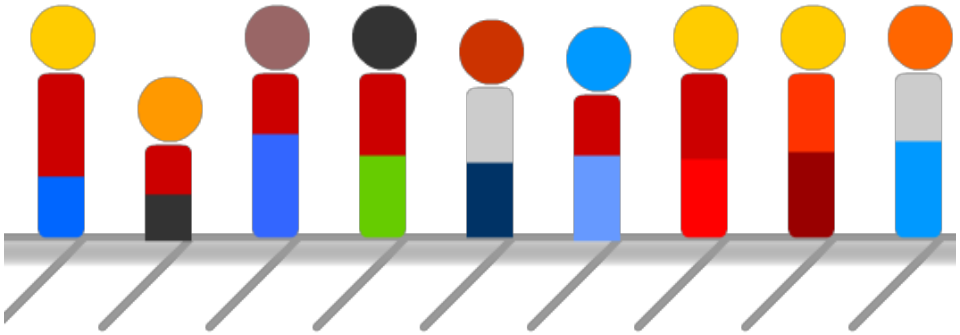
Filas

Filas

- FIFO (**F**irst **I**n, **F**irst **O**ut)

> "Primeiro elemento a entrar é o primeiro a sair"

Filas

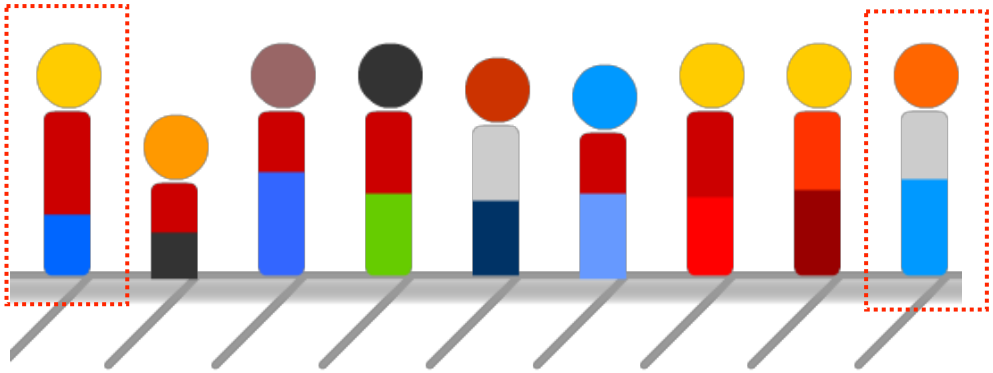


Fila de pessoas
(Queue)

Filas

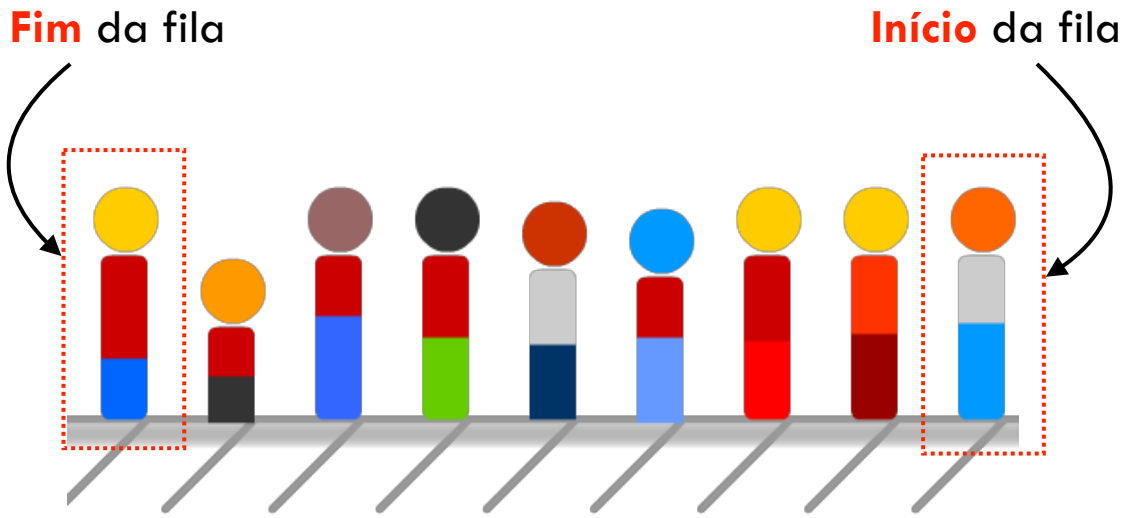
Fim da fila

Início da fila



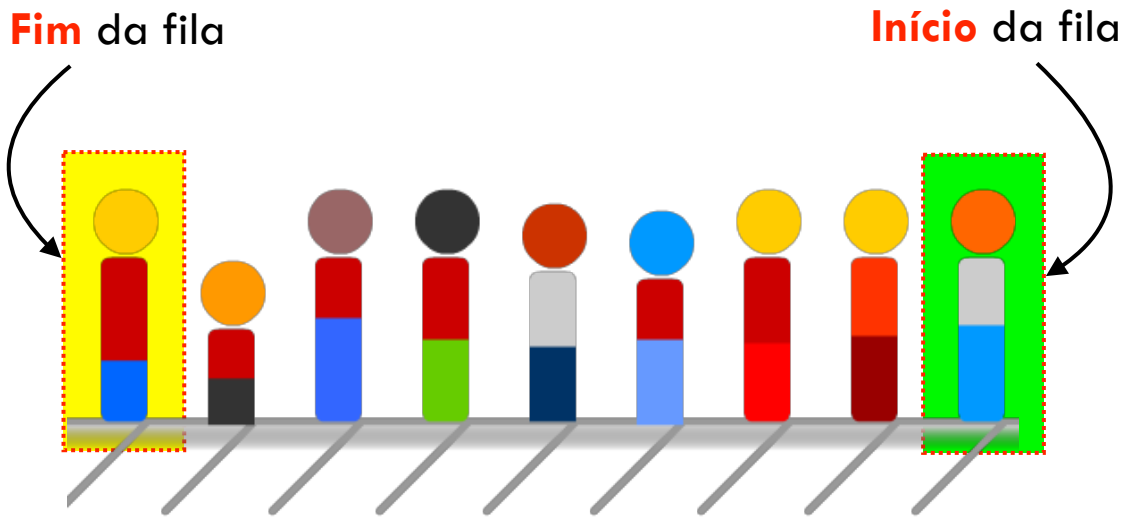
Fila de pessoas
(Queue)

Filas



Fila de pessoas
(Queue)

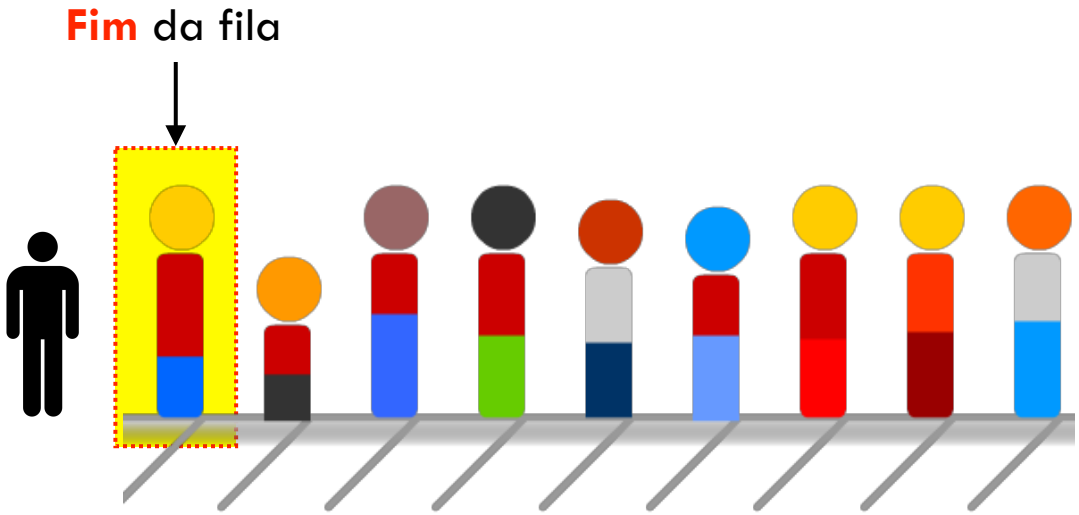
Filas



Fila de pessoas
(Queue)

Filas

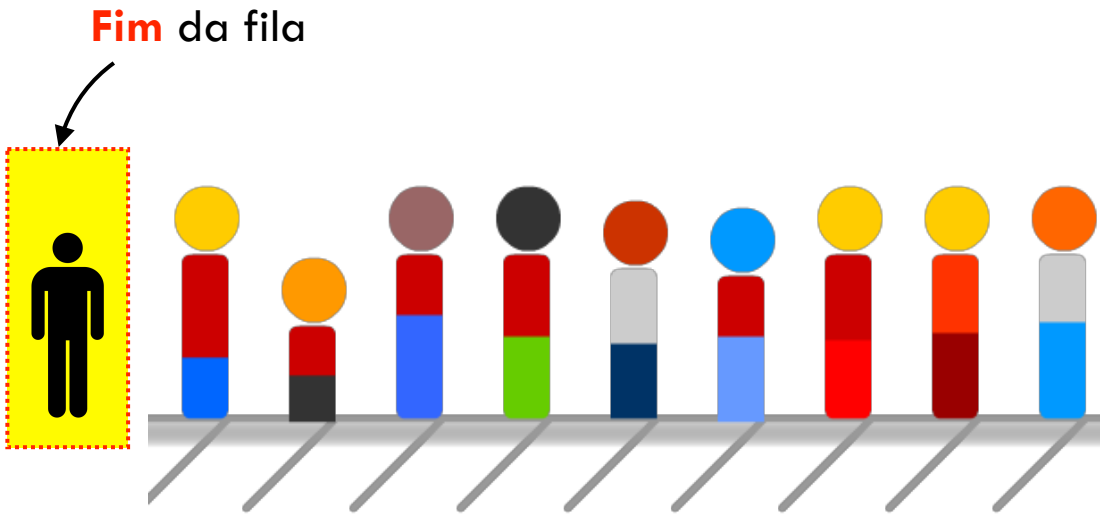
- Inserindo novo elemento



Fila de pessoas
(Queue)

Filas

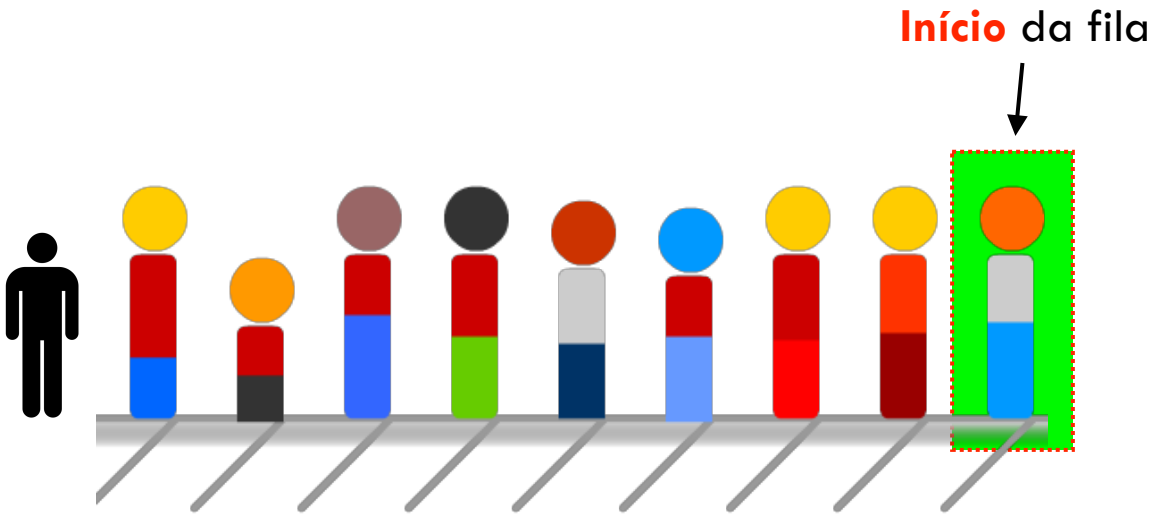
- Inserindo novo elemento



Fila de pessoas
(Queue)

Filas

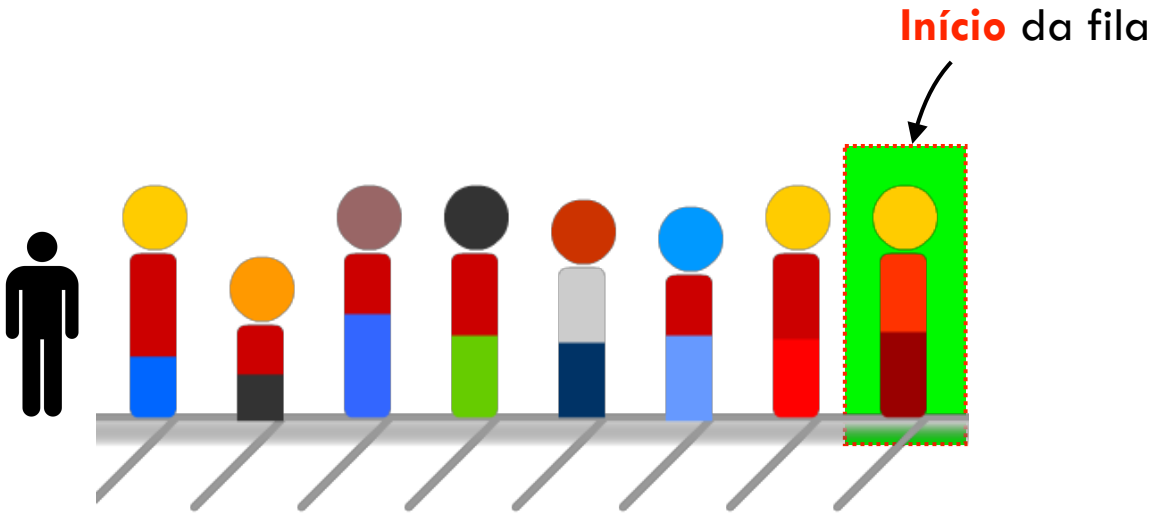
- Removendo elemento



Fila de pessoas
(Queue)

Filas

- Removendo elemento



Fila de pessoas
(Queue)

Filas

- Onde usamos?
  - buffer da analise léxica (Compiladores)
  - paginação de memória (Sistemas Operacionais)
  - fila de processos (Sistemas Operacionais)
  - algoritmos de árvores/grafos (Grafos, Inteligência Artificial)
  - quando queremos estabelecer ordem

Roteiro

1. Introdução
2. Filas
3. Operações gerais
4. Inserção de elementos
5. Remoção de elementos
6. Referências

Operações

Dada uma pilha **S**, chave **k**, elemento **x**:

- pesquisar(S, k) → procurar k em S [TRUE/FALSE]
- inserir(S, k) → inserir k em S
- remover(S, k) → remover k em S
- minimo(S) → menor valor armazenado em S
- maximo(X) → maior valor armazenado em S
- proximo(S, x) → elemento sucessor a x
- anterior(S, x) → elemento antecessor a x
- tamanho(S) → tamanho de S
- vazia(S) → S está vazia? [TRUE/FALSE]
- cheia(S) → S está cheia? [TRUE/FALSE]

Operações

- Quais operações devemos implementar para manipular um TAD de Fila?

Operações

Dada uma pilha **S**, chave **k**, elemento **x**:

- ~~pesquisar(S, k) → procurar k em S [TRUE/FALSE]~~
- **inserir(S, k) → inserir k em S**
- **remover(S, k) → remover k em S**
- ~~minimo(S) → menor valor armazenado em S~~
- ~~maximo(X) → maior valor armazenado em S~~
- ~~proximo(S, x) → elemento sucessor a x~~
- ~~anterior(S, x) → elemento antecessor a x~~
- **tamanho(S) → tamanho de S**
- **vazia(S) → S está vazia? [TRUE/FALSE]**
- **cheia(S) → S está cheia? [TRUE/FALSE]**

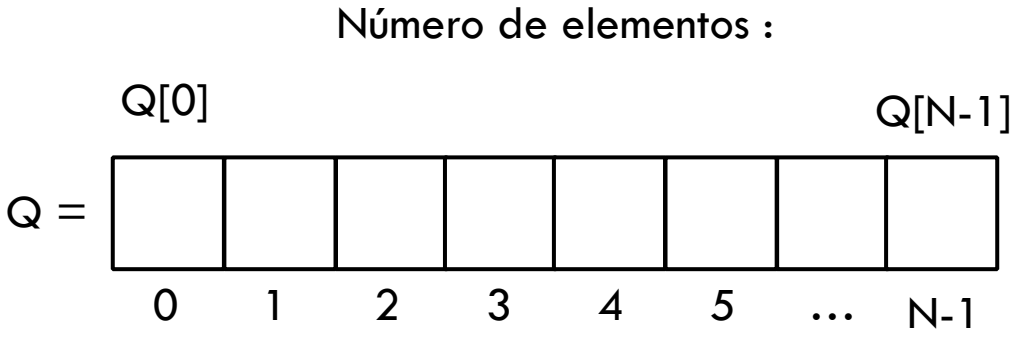
Operações

Dada uma pilha **S**, chave **k**, elemento **x**:

- ~~pesquisar(S, k) → procurar k em S [TRUE/FALSE]~~
- **inserir(S, k) → inserir k em S**
- **remover(S, k) → remover k em S**
- ~~minimo(S) → menor valor armazenado em S~~
- ~~maximo(X) → maior valor armazenado em S~~
- ~~proximo(S, x) → elemento sucessor a x~~
- ~~anterior(S, x) → elemento antecessor a x~~
- **tamanho(S) → tamanho de S**
- **vazia(S) → S está vazia? [TRUE/FALSE]**
- **cheia(S) → S está cheia? [TRUE/FALSE]**
- **primeiro(S) → elemento no início da fila**
- **último(S) → elemento no final da fila**

Fila estática

Fila (queue) Q = Arranjo de N elementos



Início

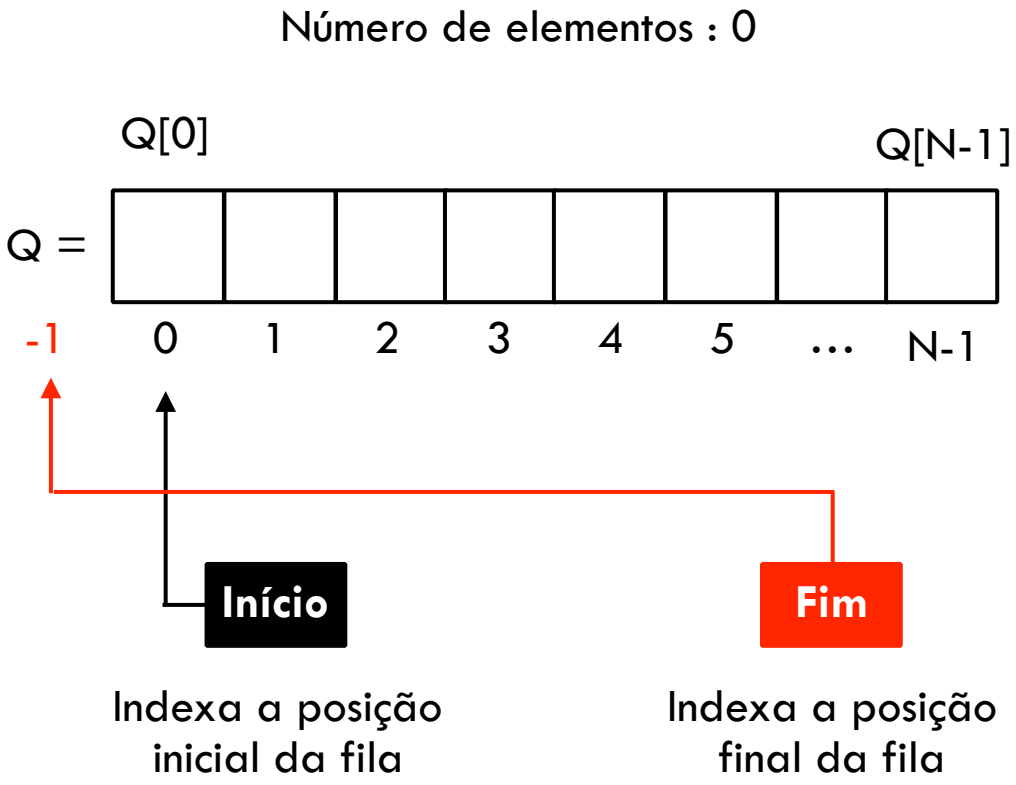
Indexa a posição inicial da fila

Fim

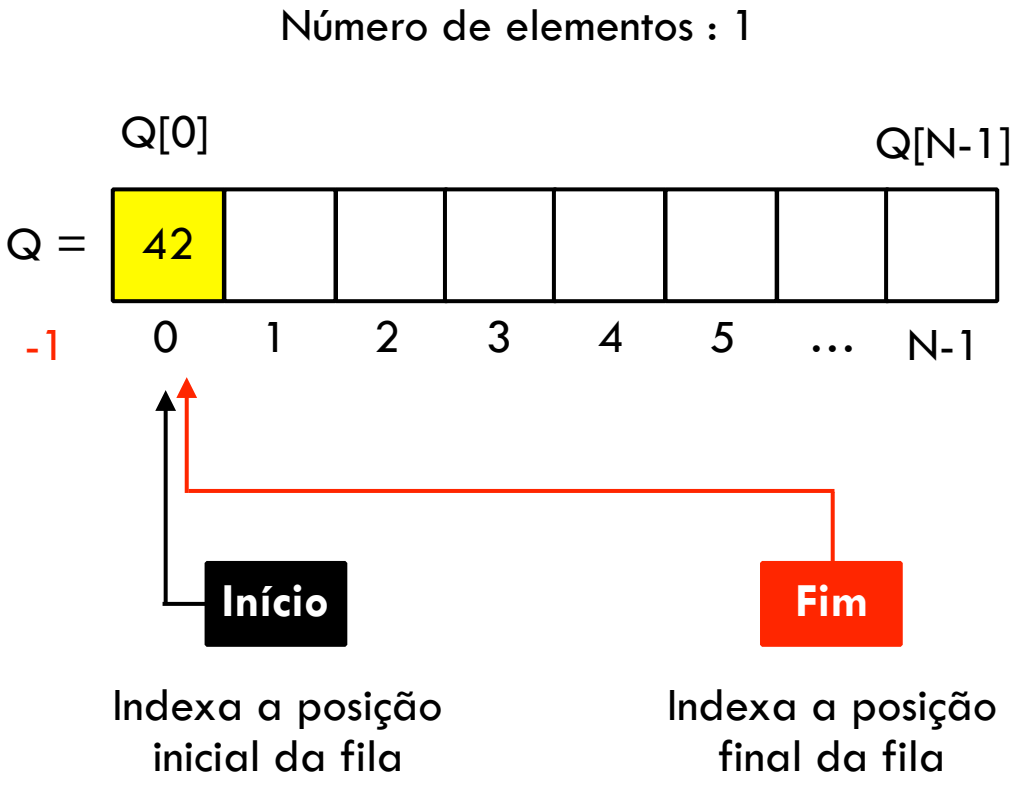
Indexa a posição final da fila

Fila estática

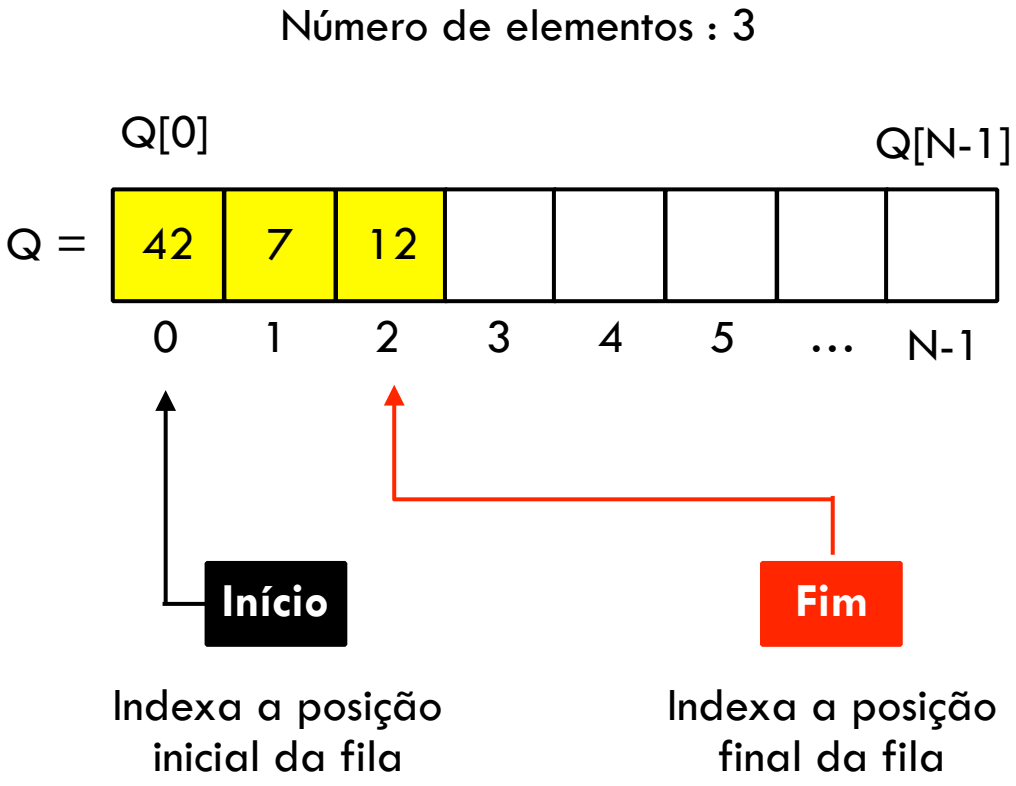
- Inicialização



Fila estática

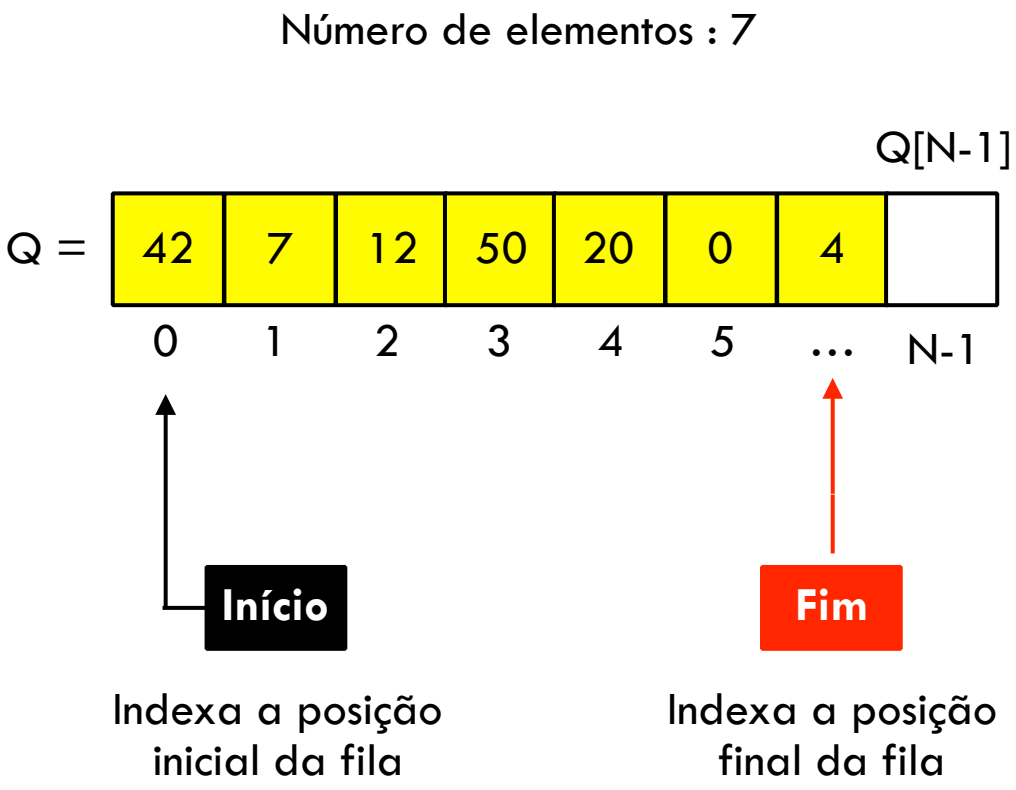


Fila estática



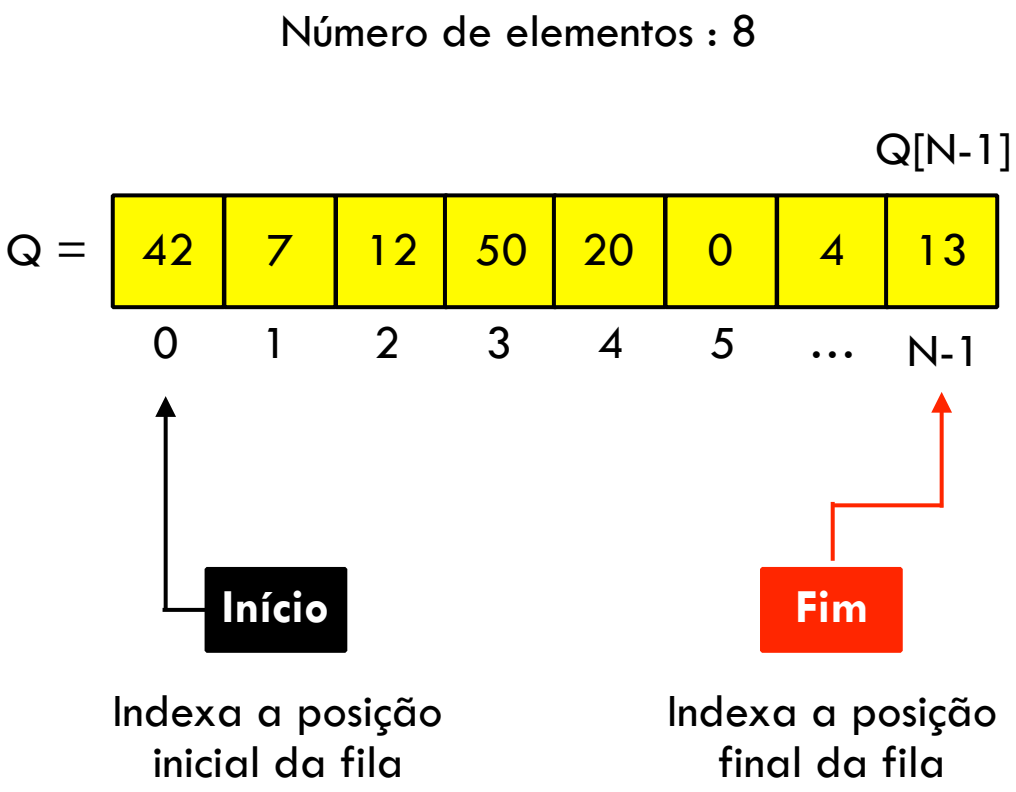
Fila estática

- Inserindo os elementos: 50, 20, 0, 4, 7



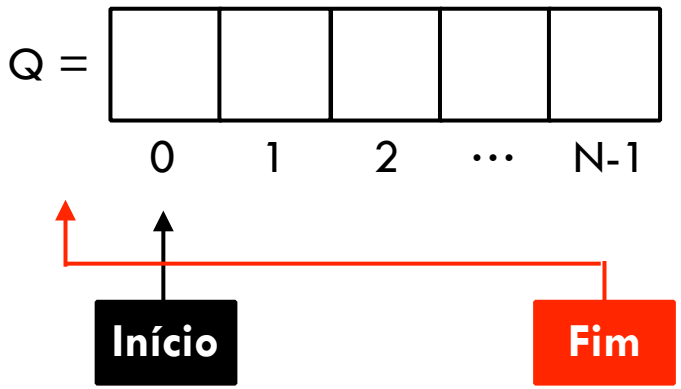
Fila estática

- Inserindo o elemento: 13



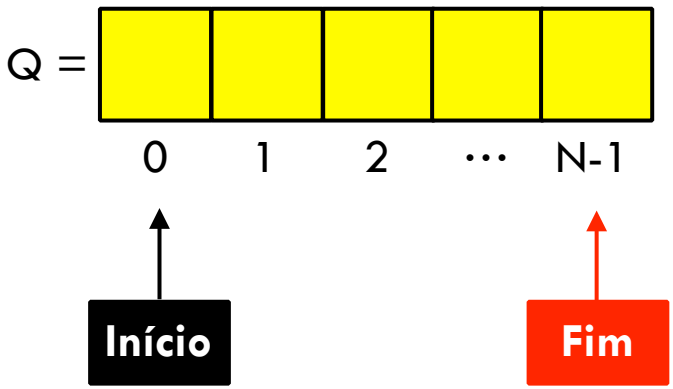
Fila estática

- Q[Q.contador] == 0 → fila vazia




```
isEmpty (Q)
1. return(Q.contador == 0)
```

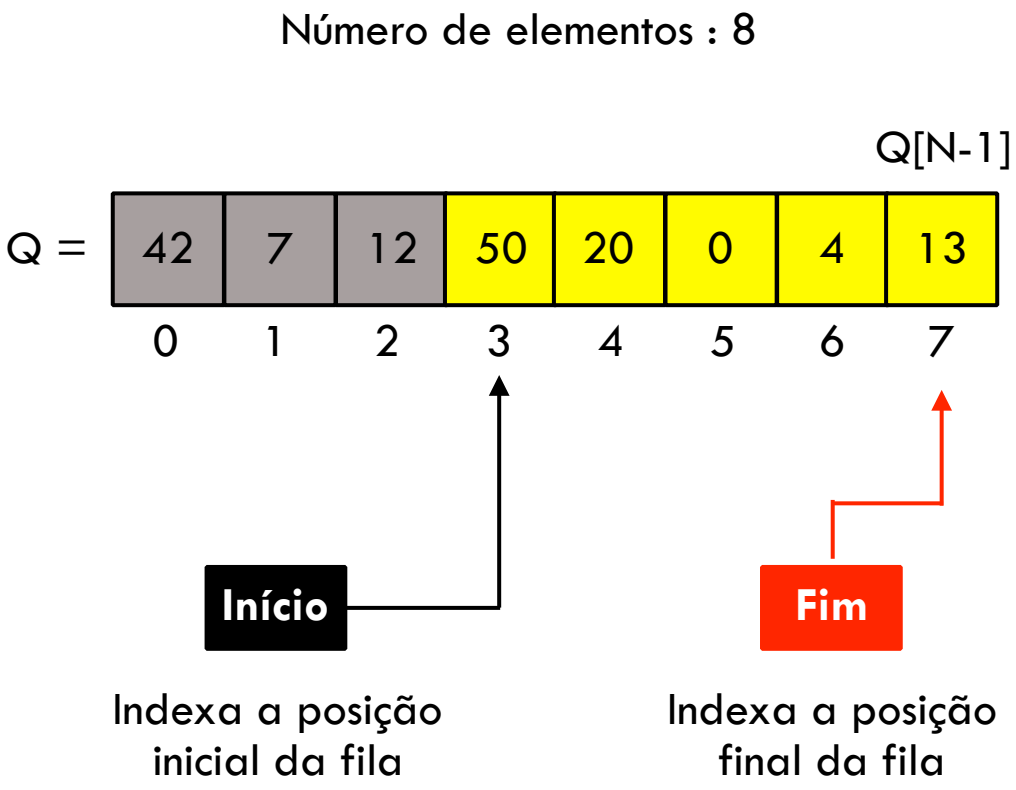
- Q [Q.contador] == N → fila cheia




```
isFull (Q)
1. return(Q.contador == N)
```

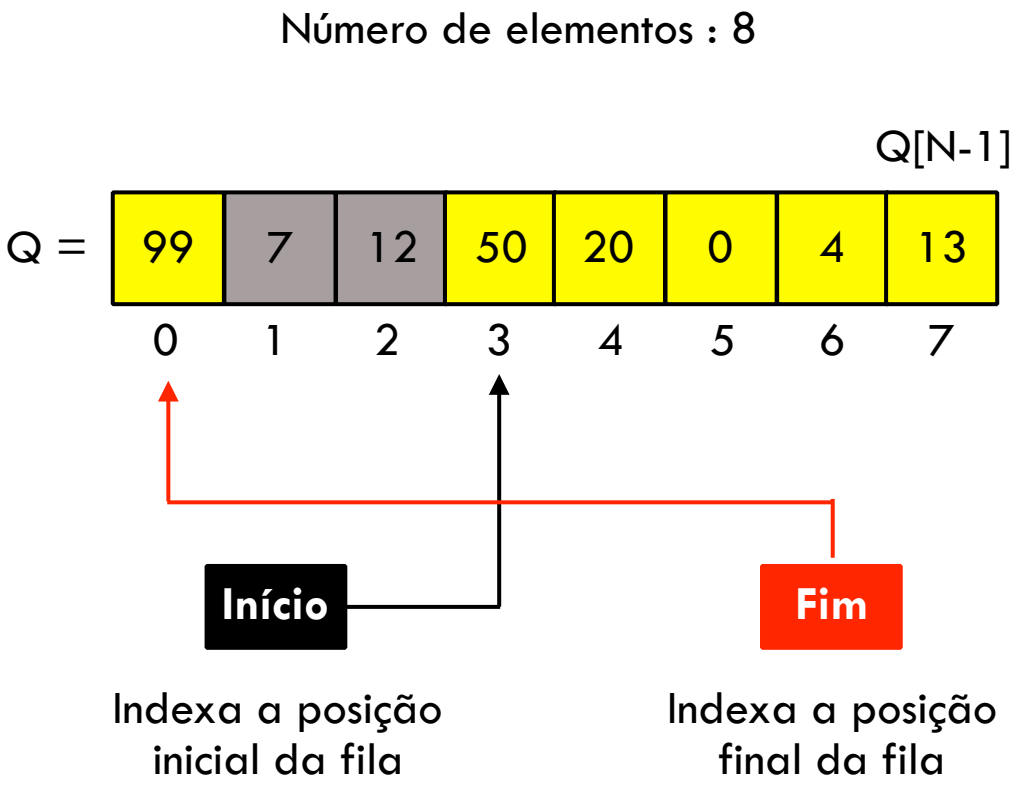
Fila estática

- Removendo 3 elementos

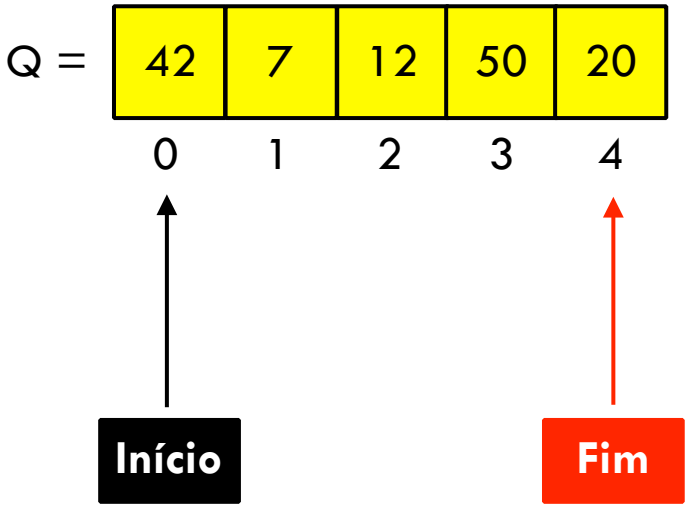


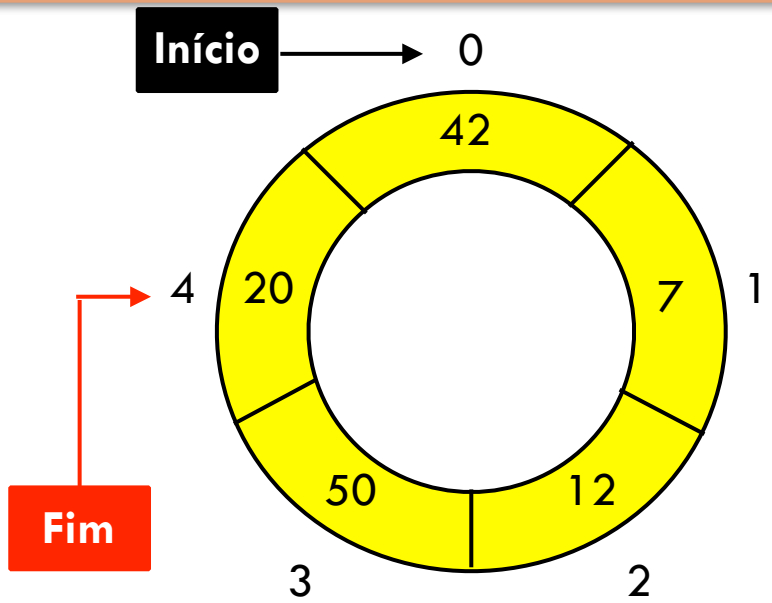
Fila estática

- Inserindo elemento 99



Fila estática





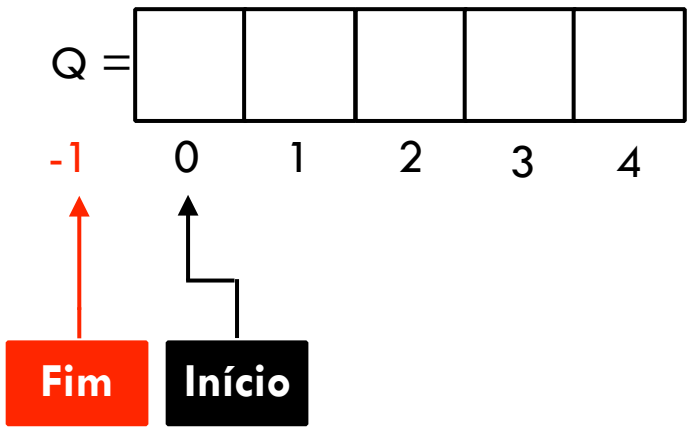
buffer circular

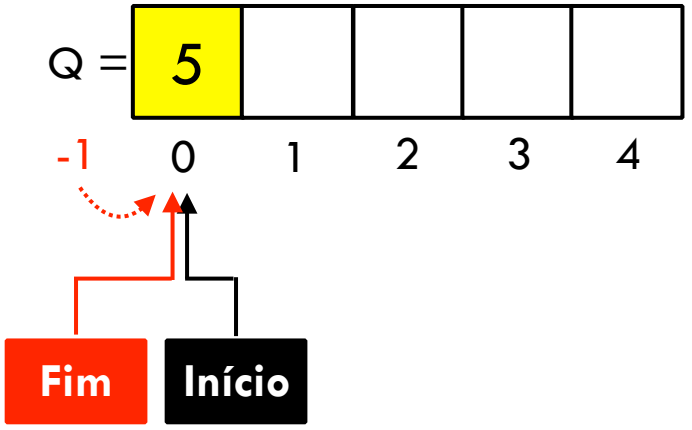
Roteiro

1. Introdução
2. Filas
3. Operações gerais
4. Inserção de elementos
5. Remoção de elementos
6. Referências

Enfileirar (enqueue)

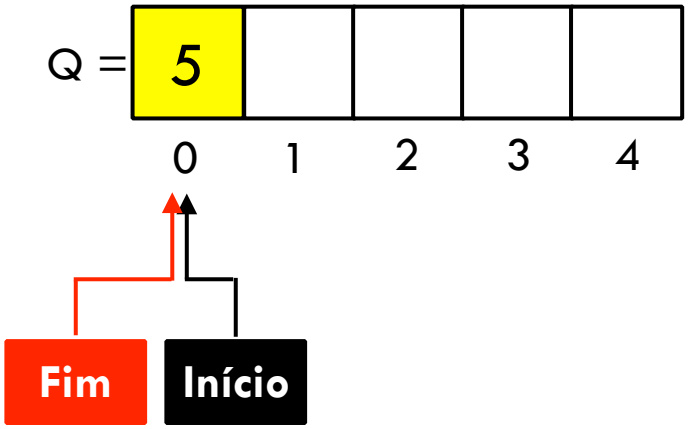
- Inserir elemento x = 5

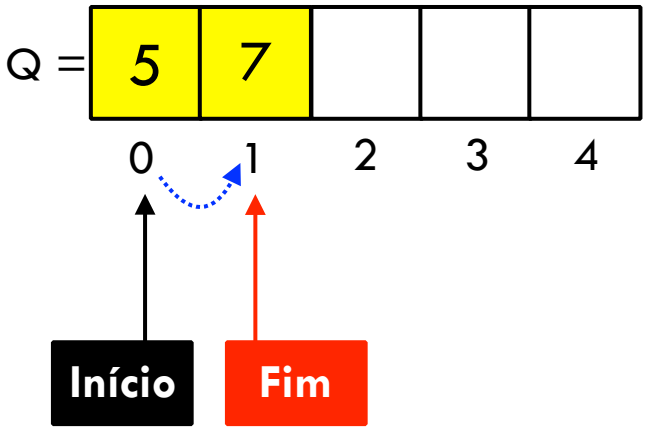




Enfileirar (enqueue)

- Inserir elemento x = 20

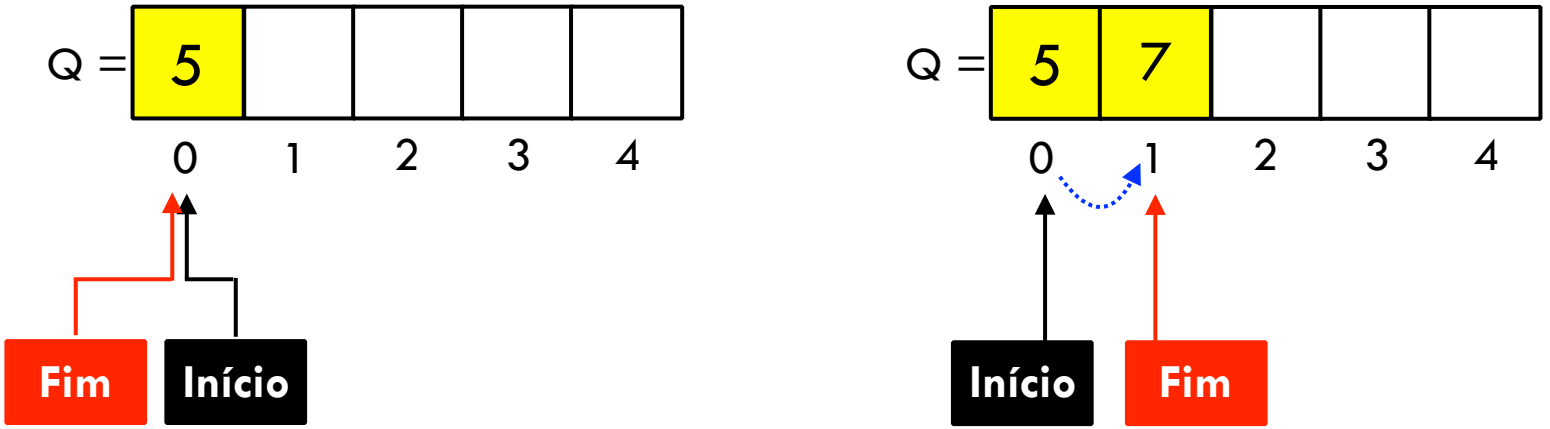




O que aconteceu?

Enfileirar (enqueue)

- Inserir elemento x = 20



O que aconteceu?

1. Incrementamos o **Fim**
2. Atribuímos o novo elemento na posição **Q[Fim]**
3. Incrementamos o contador de elementos

Enfileirar (enqueue)

```
Enqueue (Q, x)
1. se Q não está cheia:
2.    Incrementar a variável Fim
3.    Q[Fim] recebe x
4.    Incrementa o contador
```

ou

```
Enqueue (Q, x)
1. if(estaCheia(Q)==0)
2.     Q.fim = incrementaIndice(Q.fim);
3.     Q.array[Q.fim] = x
4.     Q.contador++;
```

Enfileirar (enqueue)

```
Enqueue (Q, x)
1. se Q não está cheia:
2.    Incrementar a variável Fim
3.    Q[Fim] recebe x
4.    Incrementa o contador
```

ou

Função auxiliar
comportamento circular

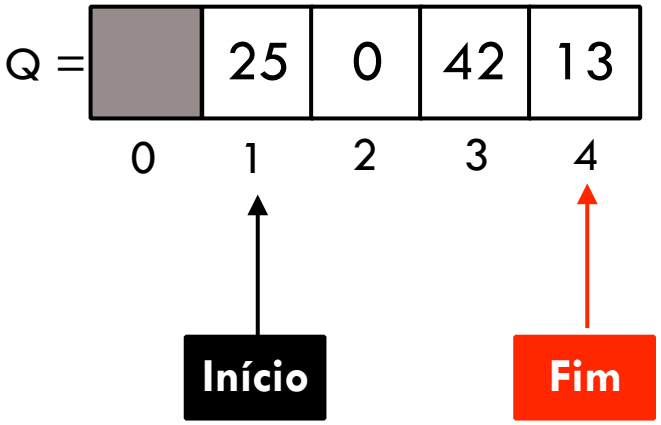
```
Enqueue (Q, x)
1. if(estaCheia(Q)==0)
2.     Q.fim = incrementaIndice(Q.fim);
3.     Q.array[Q.fim] = x
4.     Q.contador++;
```

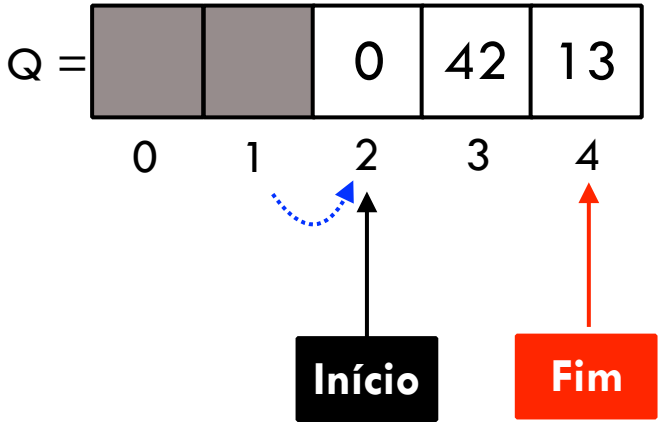
Roteiro

1. Introdução
2. Filas
3. Operações gerais
4. Inserção de elementos
5. Remoção de elementos
6. Referências

Desenfileirar (dequeue)

- Remover elemento

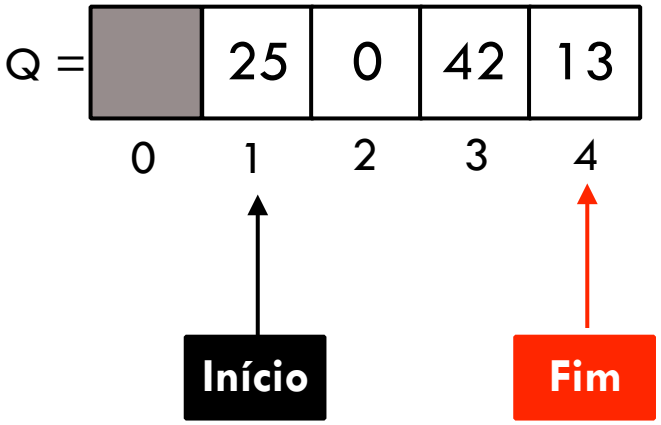


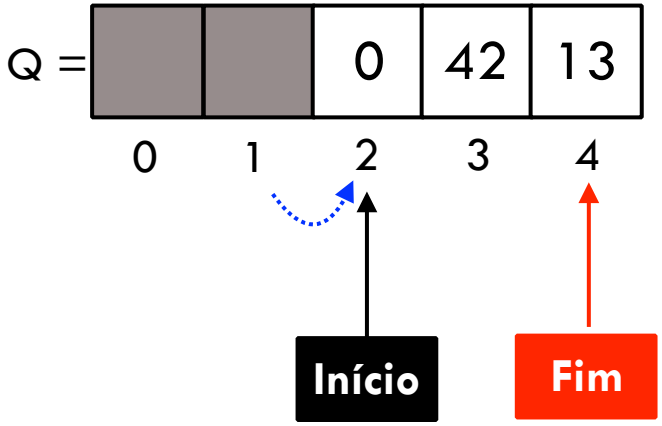


elemento retornado **25**

Desenfileirar (dequeue)

- Remover elemento



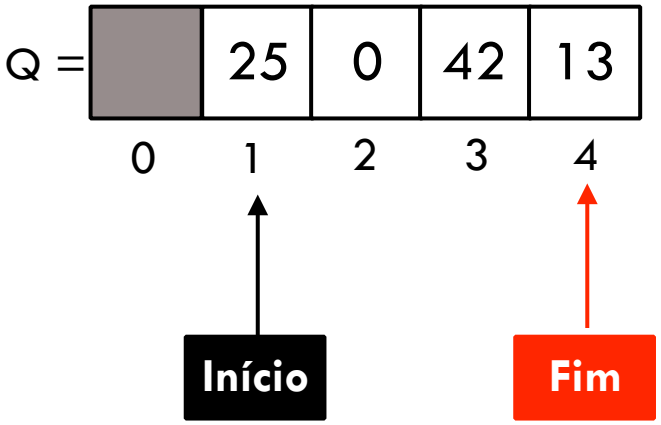


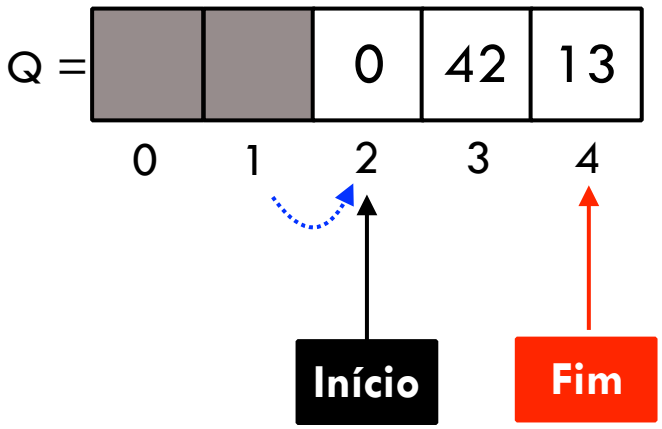
O que aconteceu?

elemento retornado **25**

Desenfileirar (enqueue)

- Remover elemento





O que aconteceu?

1. Salvamos o valor da posição Q[**Inicio**] em uma variável auxiliar
2. Incrementamos o valor da variável **Início**
3. Decrementamos o contador de elementos
4. Retornamos o valor da variável auxiliar

elemento retornado **25**

Desenfileirar (dequeue)

```
Dequeue (Q)
1. se Q não está vazia:
2.    variável auxiliar aux recebe Q[Q.Inicio]
3.    Incrementa o valor de Q.Inicio
4.    Decrementa o contador de elementos
5.    Retorna aux
```

ou

```
Dequeue (Q)
1. if(estaVazia(Q) == 0):
2.    aux = Q.array[Q.Inicio];
3.    Q.Inicio = incrementaIndice(Q.Inicio);
4.    Q.contador = Q.contador - 1;
5.    return(aux);
```

Desenfileirar (dequeue)

```
Dequeue (Q)
1. se Q não está vazia:
2.    variável auxiliar aux recebe Q[Q.Inicio]
3.    Incrementa o valor de Q.Inicio
4.    Decrementa o contador de elementos
5.    Retorna aux
```

OU

Função auxiliar comportamento circular

```
Dequeue (Q)
1. if(estaVazia(Q) == 0):
2.    aux = Q.array[Q.Inicio];
3.    Q.Inicio = incrementaIndice(Q.Inicio);
4.    Q.contador = Q.contador - 1;
5.    return(aux);
```

Funções adicionais?

- Quais outras funções podem ser úteis para o tipo Fila?

Exercício 01

- Ilustre cada estado de uma fila após realizar as seguintes operações (em ordem)
  - Enqueue(Q, 4)
  - Enqueue(Q, 1)
  - Enqueue(Q, 3)
  - Dequeue(Q)
  - Enqueue(Q, 8)
  - Dequeue(Q)

- Considere que a pilha está inicialmente vazia e é armazenada em um arranjo Q[1 .. 6]

Exercício 02

- Mãos a obra: implemente um TDA para Fila com alocação estática, e as funções de manipulação.

- Quais TDAs serão necessários?

Tipo Abstrato para Fila Estática

```
typedef struct {
    int key;
} Item;

typedef struct {
  Item array[MAXTAM];
  int inicio;
  int fim;
  int tamanho;
} filaEstatica;
```

```
void iniciaFila(filaEstatica *fila);
void enfileira(Item item, filaEstatica *fila);
void desenfileira(filaEstatica *fila, Item *item);
void imprimeFila(filaEstatica *fila);
int incrementaIndice(int i);
int estaVazia(filaEstatica *fila);
int estaCheia(filaEstatica *fila);
int tamanhoFila(filaEstatica *fila);
Item primeiro(filaEstatica *fila);
Item ultimo(filaEstatica *fila);
```

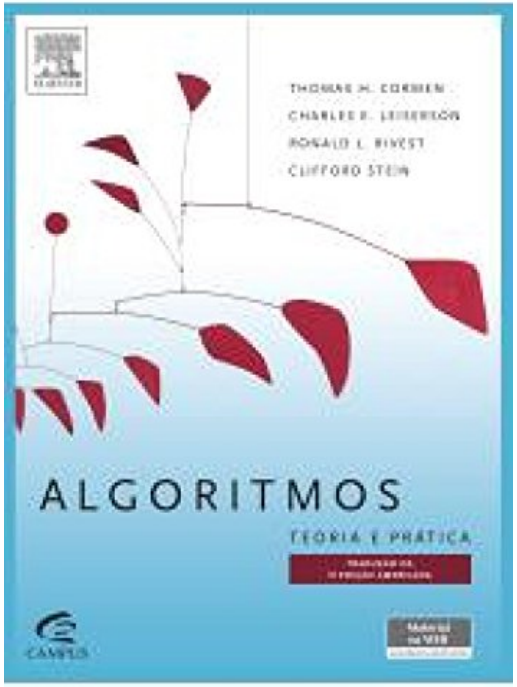
Próximas Aulas

- Filas → implementação dinâmica
- Listas Lineares
  - single-linked
  - double-linked

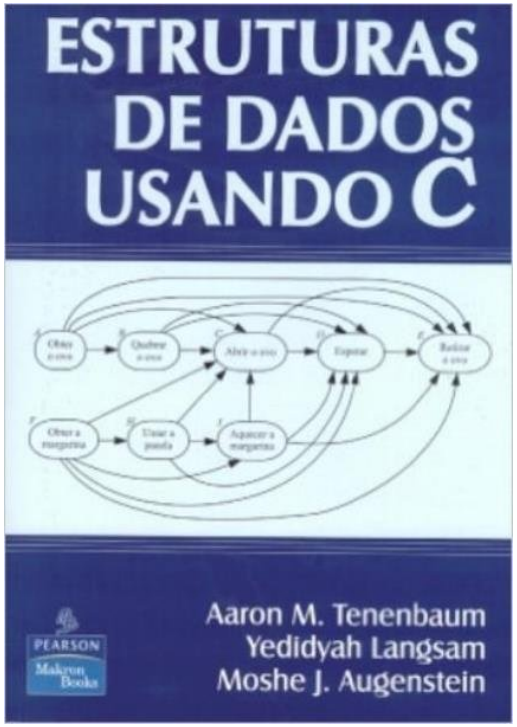
Roteiro

1. Introdução
2. Filas
3. Operações gerais
4. Inserção de elementos
5. Remoção de elementos
6. Referências

Referências sugeridas

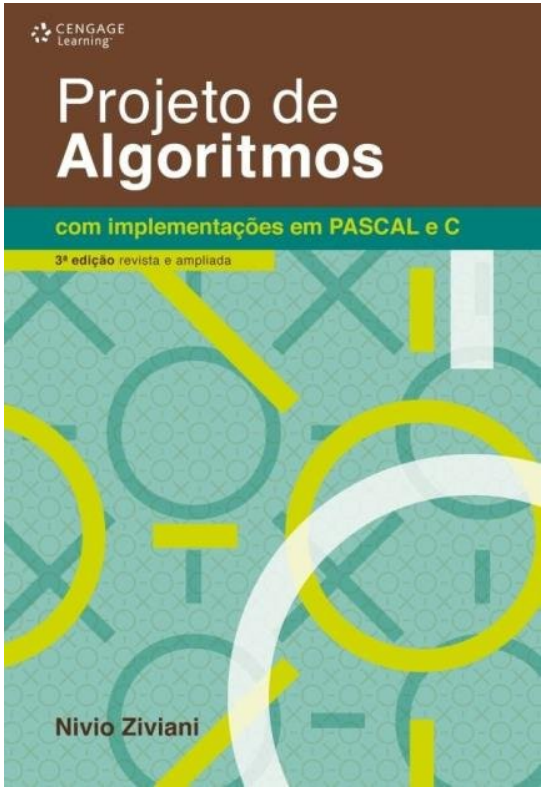


[Cormen et al, 2018]



[Tenenbaum et al, 1995]

Referências sugeridas



[Ziviani, 2010]

[Drozdek, 2017]

Perguntas?

Prof. Rafael G. **Mantovani**

rafaelmantovani@utfpr.edu.br