

Fila

Algoritmos e Estruturas de Dados I

Fila

- O que é?
- Para que serve?



Biblioteca

- Problema: **automação de uma biblioteca**
 - Todos os livros devem ser cadastrados
 - O sistema deve informar se um livro está disponível ou não nas estantes
 - Caso o livro não esteja disponível, o usuário pode aguardar em uma fila de espera
 - Quando o livro for devolvido, o primeiro da fila de espera pode retirá-lo

- Sua tarefa: desenvolver esse sistema

Biblioteca

- 1^o passo: abstração
 - Identificar os elementos do mundo real que são relevantes para a solução do problema

Biblioteca

■ Quais são eles?

fila de espera para o livro	livros do acervo	disponível?
 <p>último ---> <--- 1º</p>	 <p>trigonometria</p>	não
 <p>último ---> <--- 1º</p>	 <p>química inorgânica</p>	não
fila vazia!	 <p>estruturas de dados</p>	sim

Biblioteca

- Elementos relevantes
 - Um cadastro de livros
 - Indicação da disponibilidade dos livros
 - Uma fila de espera para cada livro, com indicação da ordem das pessoas
 - Primeiro e último da fila
 - Cadastro de pessoas: nome, endereço e telefone

Biblioteca

- 2º passo: quais são as operações possíveis nas filas?
 - Entrar na fila
 - Quem entra, entra onde?
 - Sair da fila
 - Quem sai, sai de onde?
 - Outras?

Fila (*queue*)

- O que é?

- *É uma estrutura para armazenar um conjunto de elementos, que funciona da seguinte forma*

- Novos elementos sempre entram no fim da fila
 - O único elemento que se pode retirar da fila em um dado momento é seu primeiro elemento

- Para que serve?

- Modelar situações em que é preciso armazenar um conjunto ordenado de elementos, no qual o primeiro elemento a entrar no conjunto será também o primeiro elemento a sair do conjunto, e assim por diante

- F.I.F.O

- *First In, First Out*

Aplicações de fila

- Biblioteca
 - Lista de espera para livros
- Impressão
 - Documentos a serem impressos
- Aeroporto
 - Lista de espera para vôos
- Tarefas (Jobs) do Sistema Operacional
- Outras?

Operações

- $\text{Cria}(F)$: cria uma fila F vazia
- $\text{Entra}(F, X)$: X entra no fim da fila F
- $\text{Sai}(F, X)$: o primeiro elemento da fila F é retirado da fila e atribuído a X
- $\text{IsEmpty}(F)$: verdade se a fila estiver vazia; caso contrário, falso
- $\text{IsFull}(F)$: verdade se a fila estiver cheia; caso contrário, falso

Exemplo

operação	fila	resultado
cria(F)	1º DA FILA -->	
entra(F, a)	1º DA FILA --> a	
entra(F, b)	1º DA FILA --> a, b	
entra(F, c)	1º DA FILA --> a, b, c	
sai(F, X)	1º DA FILA --> b, c	X = a
entra(F, d)	1º DA FILA --> b, c, d	
sai(F, X)	1º DA FILA --> c, d	X = b

Implementação da fila

■ Alocação seqüencial

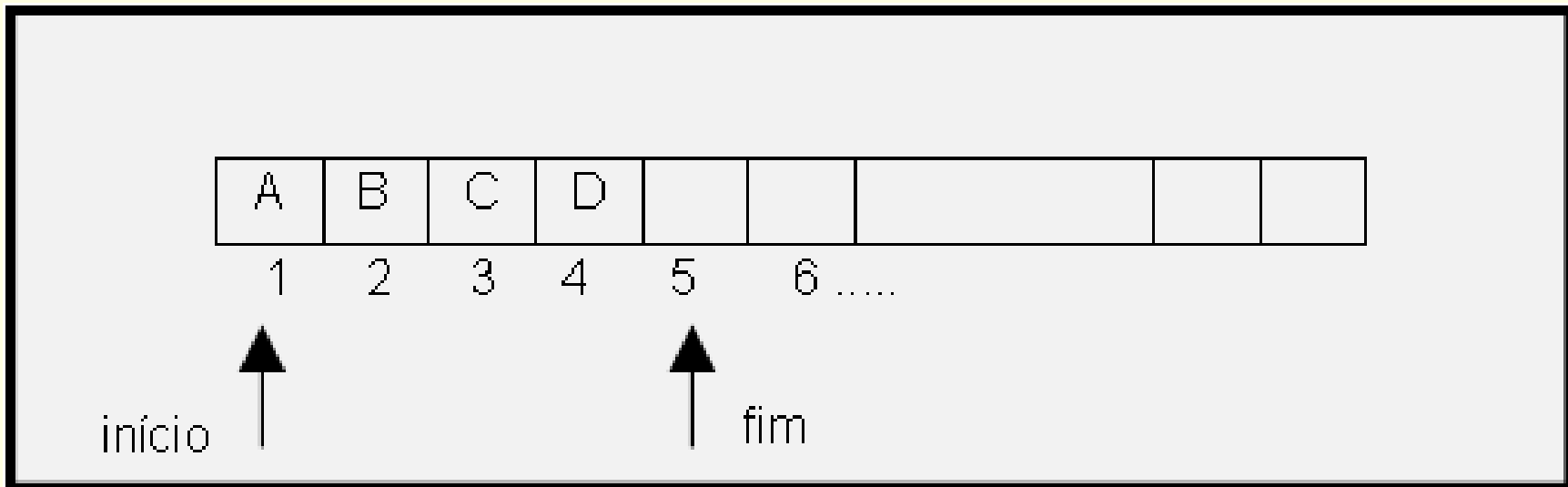
- Os elementos da fila ficam, necessariamente, em seqüência (um ao lado do outro) na memória

■ Alocação estática

- Todo o espaço de memória a ser utilizado pela fila é reservado (alocado) em tempo de compilação
- Todo o espaço reservado permanece reservado durante todo o tempo de execução do programa, independentemente de estar sendo efetivamente usado ou não

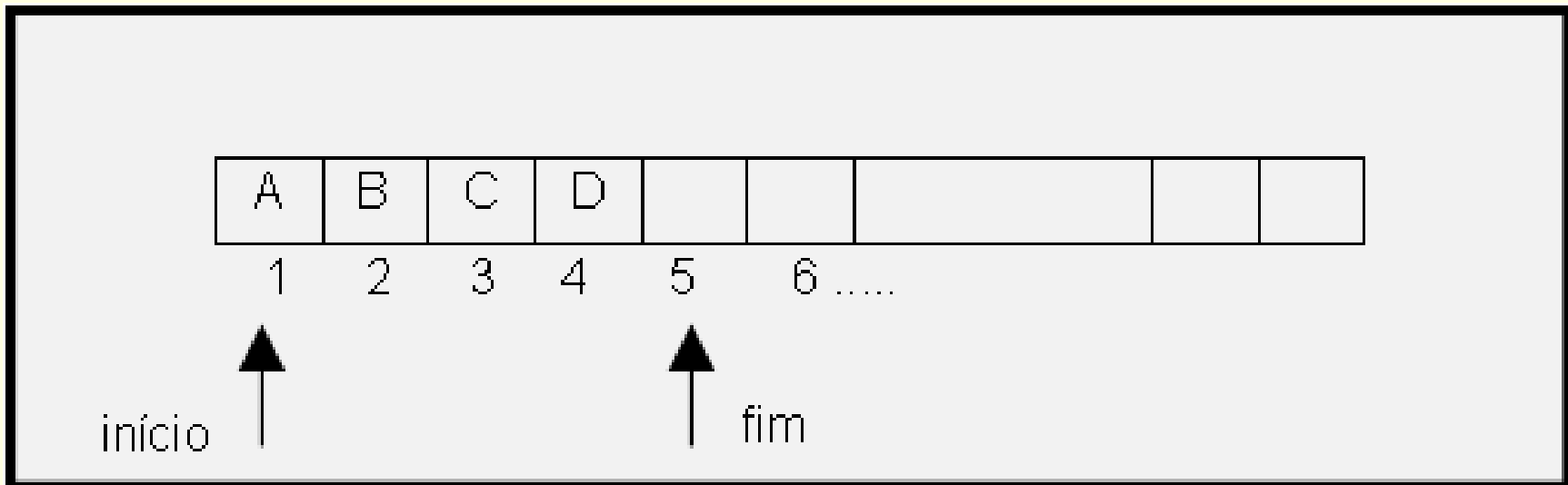
Implementação da fila

- **Início** aponta para/indica o primeiro da fila, ou seja, o primeiro elemento a sair
- **Fim** aponta para/indica o fim da fila, ou seja, onde o próximo elemento entrará



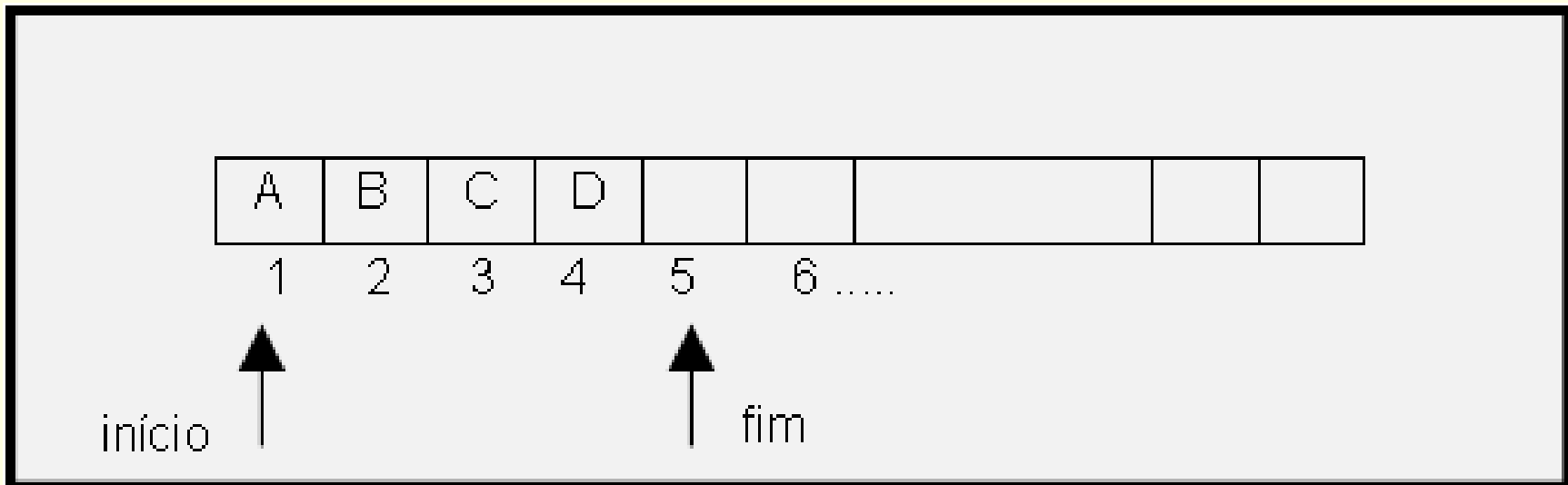
Implementação da fila

- Qual a condição inicial, quando a fila é criada?
- Qual a condição para fila vazia?
- Qual a condição para fila cheia?



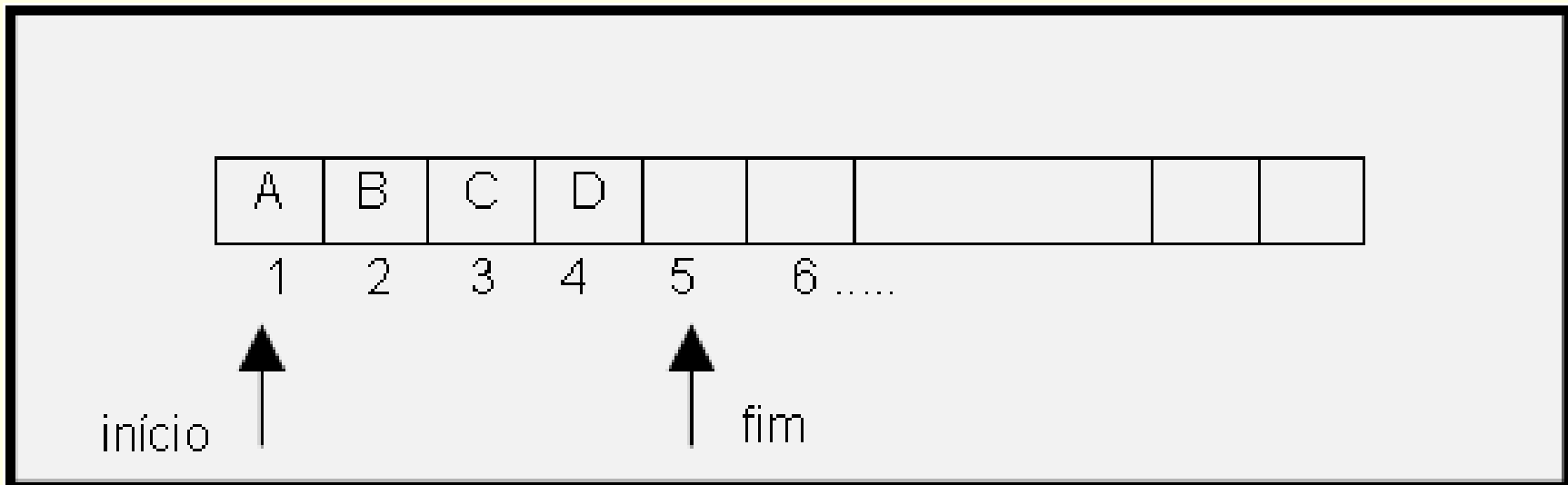
Implementação da fila

- Qual a condição inicial, quando a fila é criada?
 - Início=0, fim=1 (fim= índice adiantado do (próximo) lugar para inserção)
- Qual a condição para fila vazia?
- Qual a condição para fila cheia?



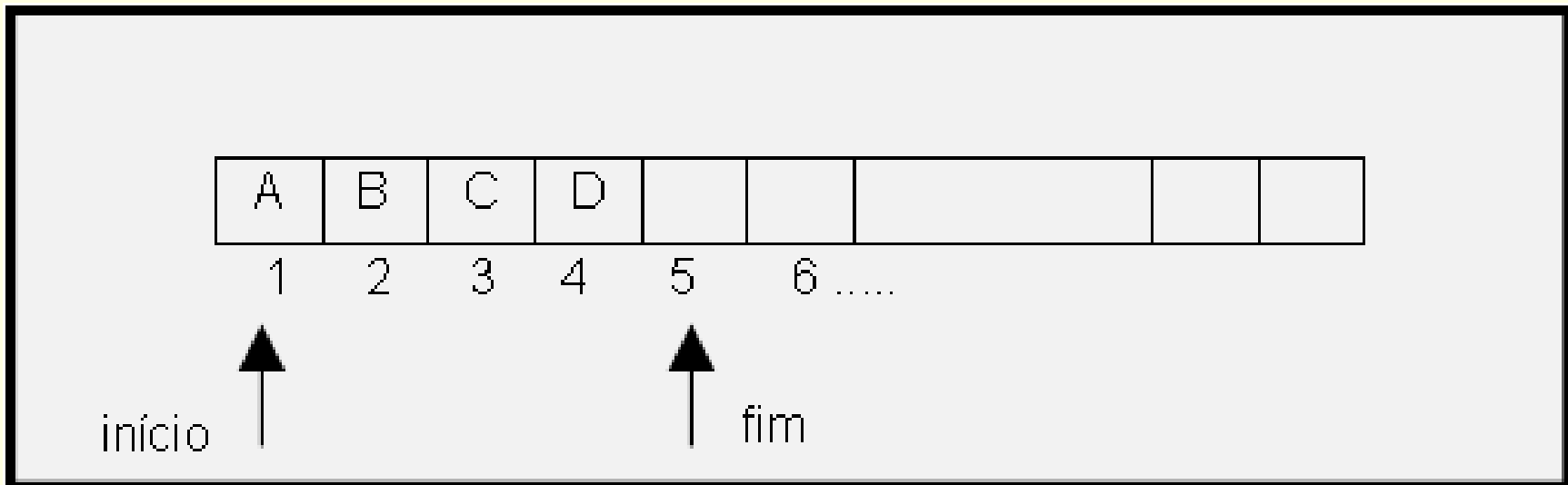
Implementação da fila

- Qual a condição inicial, quando a fila é criada?
 - Início=0, fim=1
- Qual a condição para fila vazia?
 - Início=0, fim=1 ?
- Qual a condição para fila cheia?



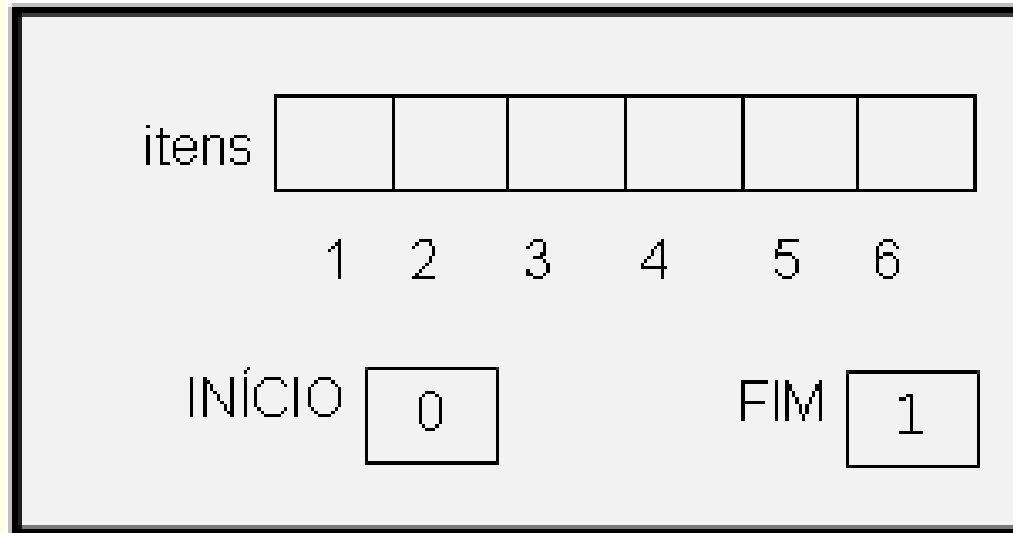
Implementação da fila

- Qual a condição inicial, quando a fila é criada?
 - Início=0, fim=1
- Qual a condição para fila vazia?
 - Início=0, fim=1 ?
- Qual a condição para fila cheia?
 - fim=tamanho do array+1 ?



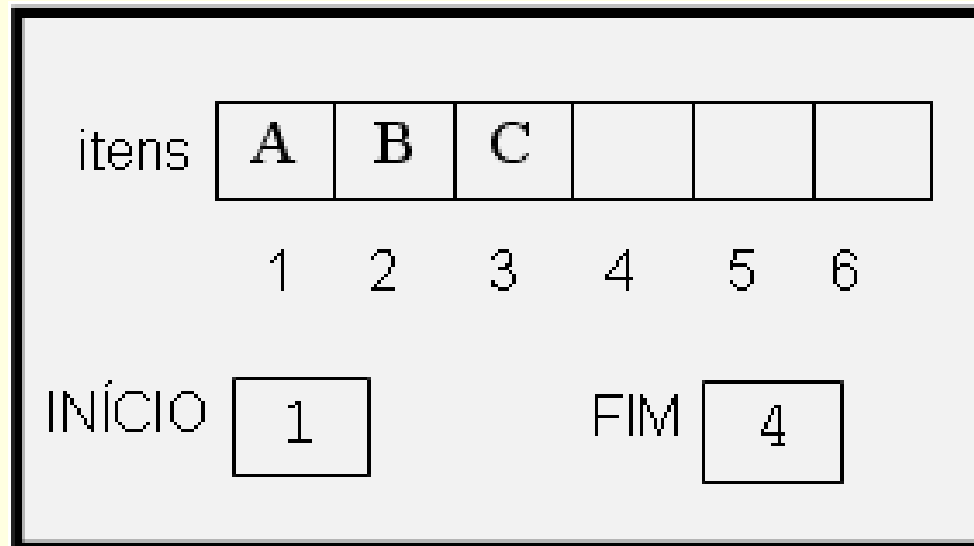
Exemplo de uso da fila

- Criação da fila



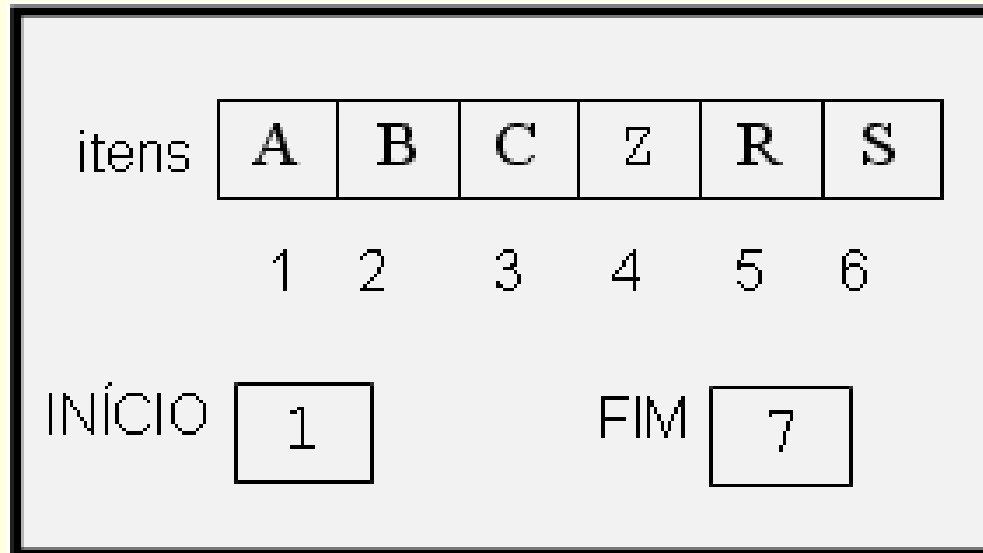
Exemplo de uso da fila

- `entra(F,A)`, `entra(F,B)`, `entra(F,C)`



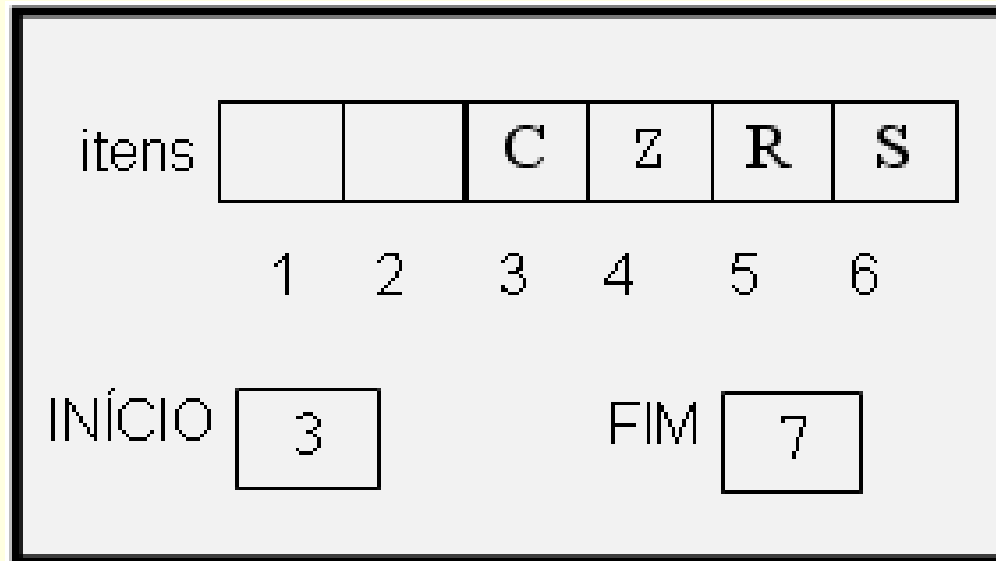
Exemplo de uso da fila

- entra(F,Z), entra(F,R), entra(F,S)
 - IsFull=TRUE



Exemplo de uso da fila

- sai(F,X), sai(F,X)
 - IsFull=TRUE !!!!



Como inserir mais elementos?

Qual o problema com a fila?

Fila

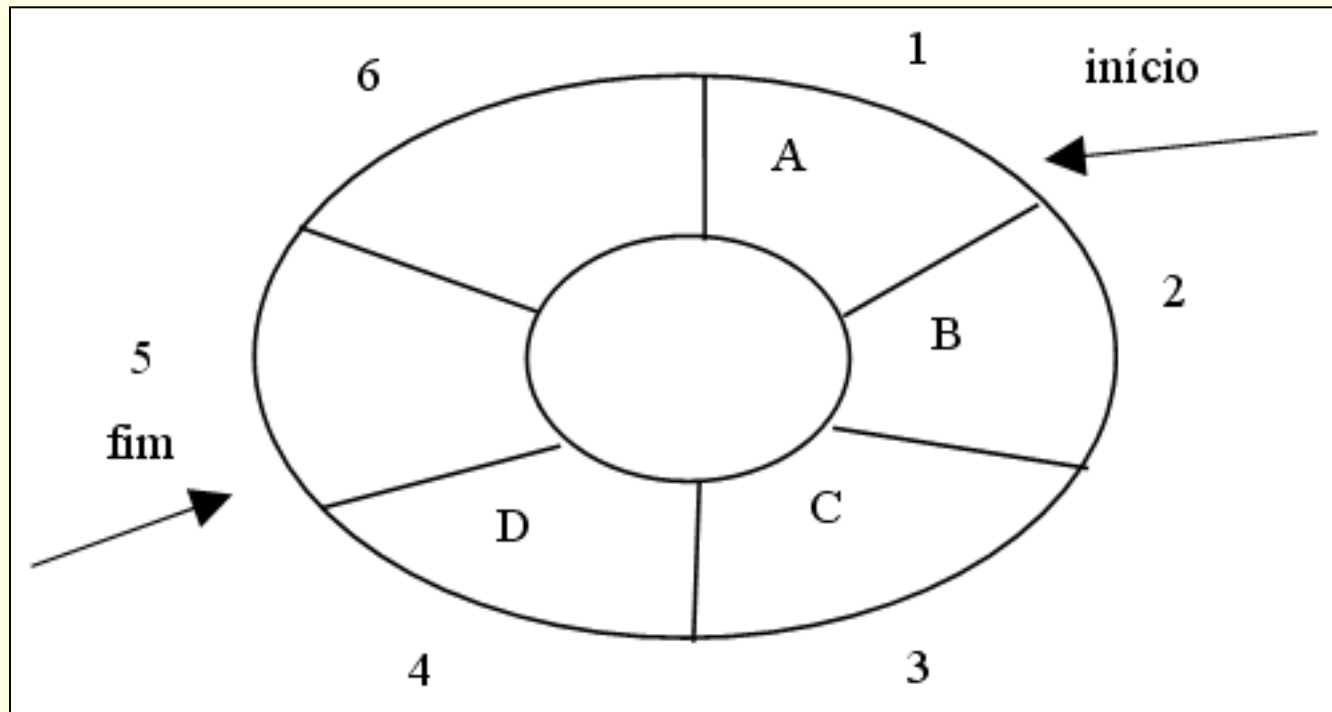
- Como reutilizar os espaços do início da fila?

Fila

- Como reutilizar os espaços do início da fila?
 - Outra forma de implementação
 - Melhor aproveitamento da representação utilizada

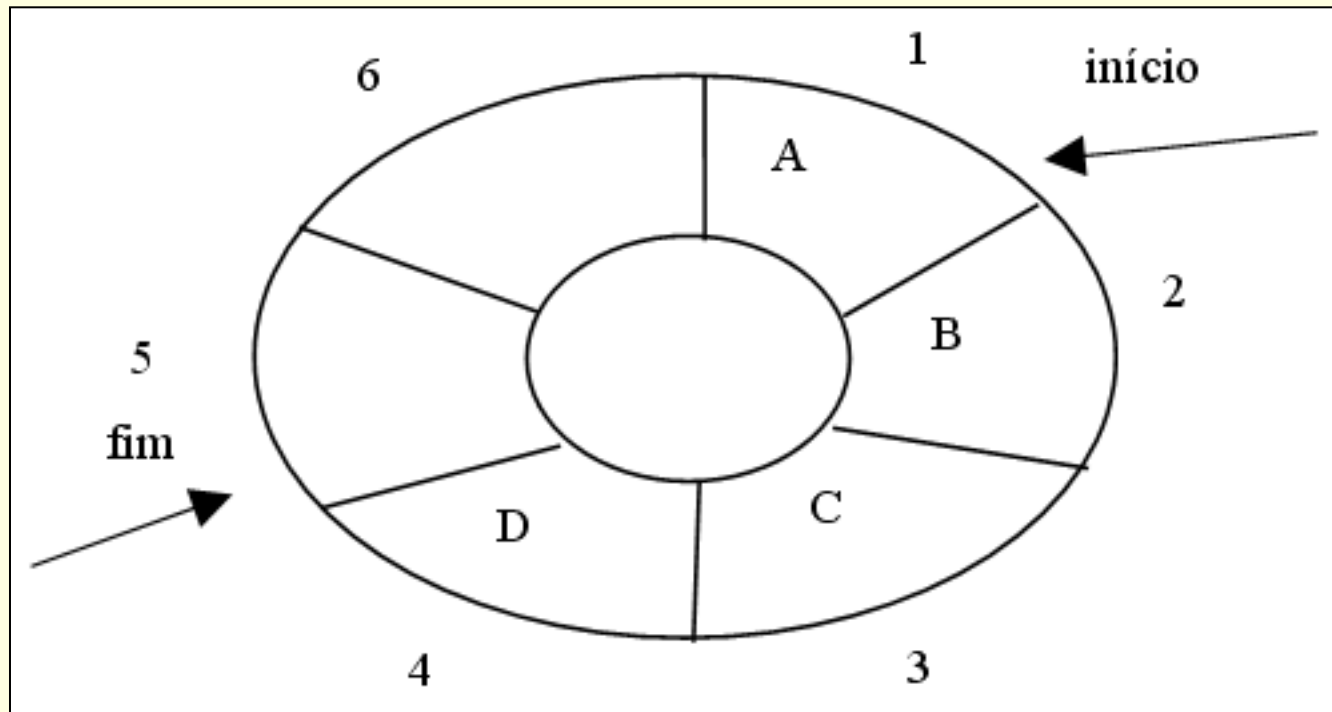
Fila vista como um ANEL

Fila como ANEL



- Qual a condição para fila vazia?
- Qual a condição para fila cheia?
- Qual a condição inicial (quando a fila é criada)?

Fila como ANEL

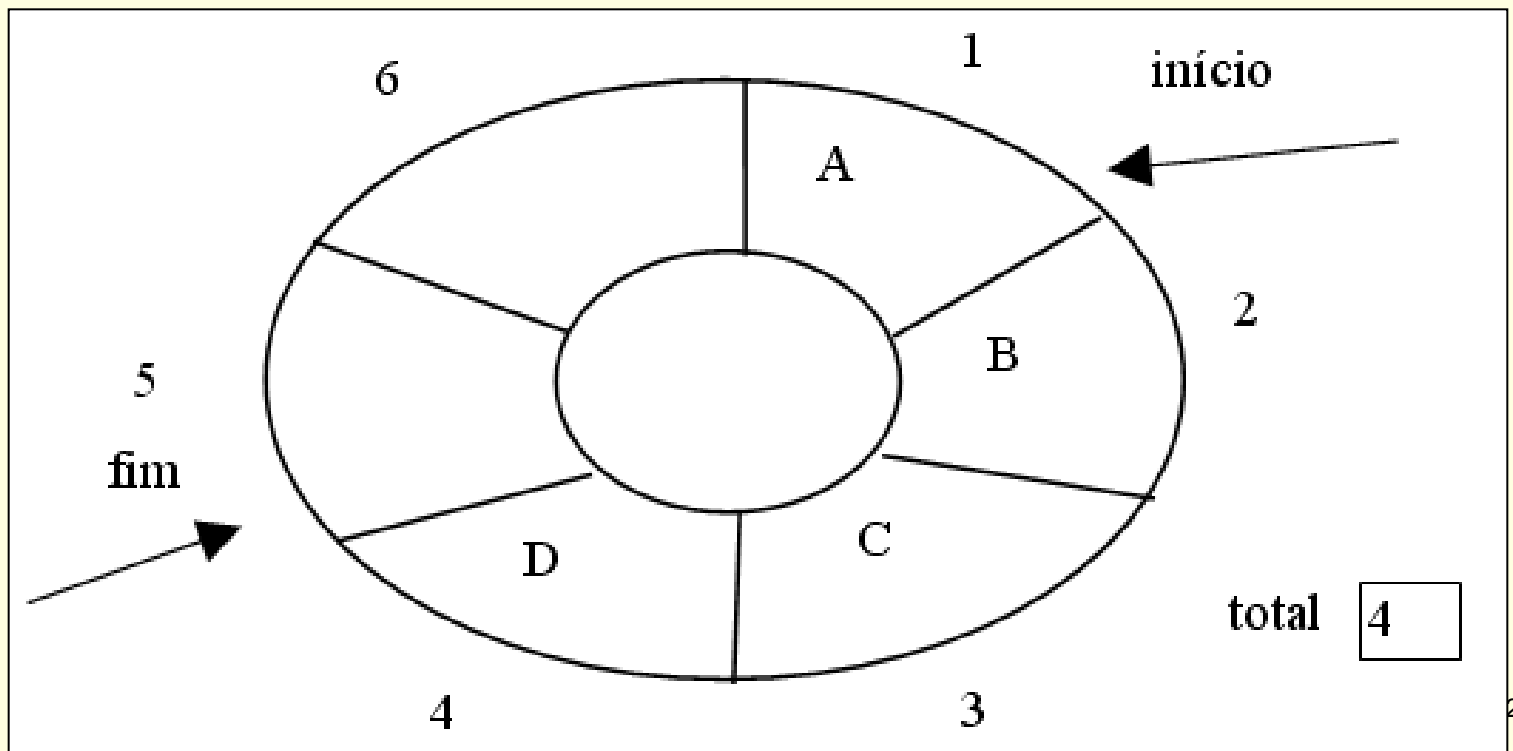


- Qual a condição para fila vazia?
- Qual a condição para fila cheia?
- Qual a condição inicial (quando a fila é criada)?

Difícil! Perde-se um pouco do sentido com essa representação

Fila como ANEL

- **Solução:** campo extra para guardar número de elementos



Fila como ANEL

- Qual a condição para fila vazia?
- Qual a condição para fila cheia?
- Qual a condição inicial (quando a fila é criada)?

Fila como ANEL

- Qual a condição para fila vazia?
 - Total=0
- Qual a condição para fila cheia?
- Qual a condição inicial (quando a fila é criada)?

Fila como ANEL

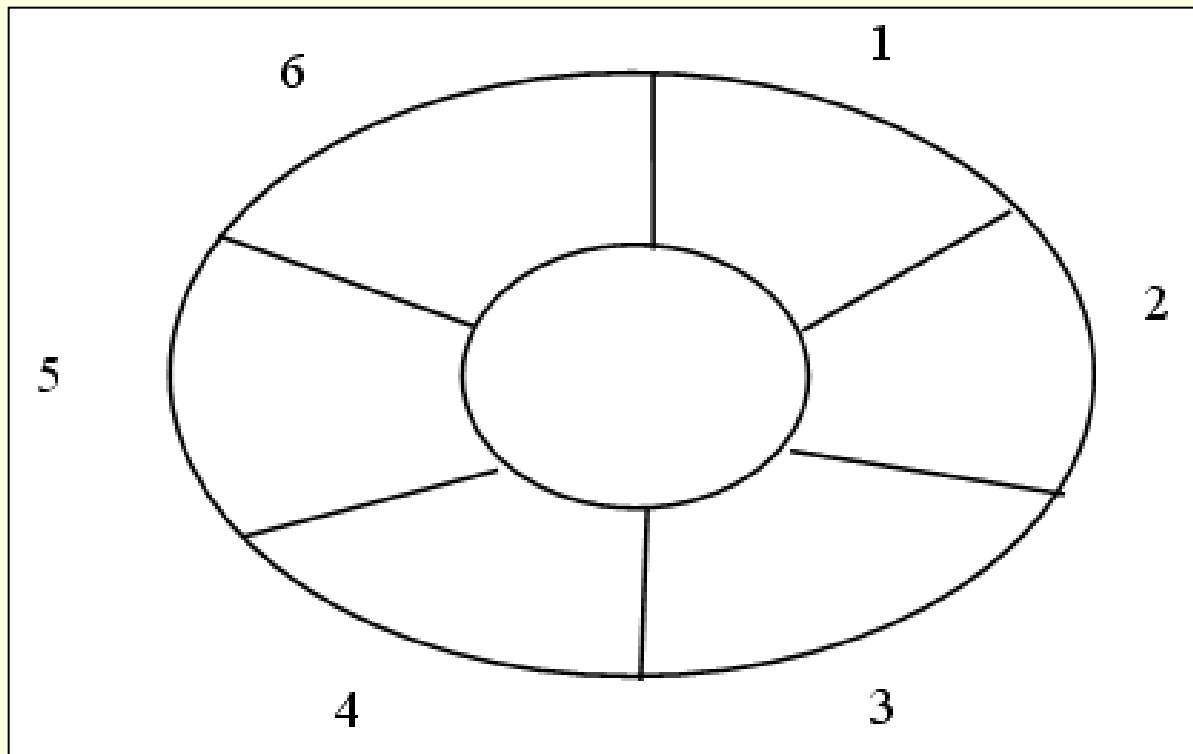
- Qual a condição para fila vazia?
 - Total=0
- Qual a condição para fila cheia?
 - Total=tamanho do array
- Qual a condição inicial (quando a fila é criada)?

Fila como ANEL

- Qual a condição para fila vazia?
 - Total=0
- Qual a condição para fila cheia?
 - Total=tamanho da fila
- Qual a condição inicial (quando a fila é criada)?
 - Total=0, início=1, fim=1

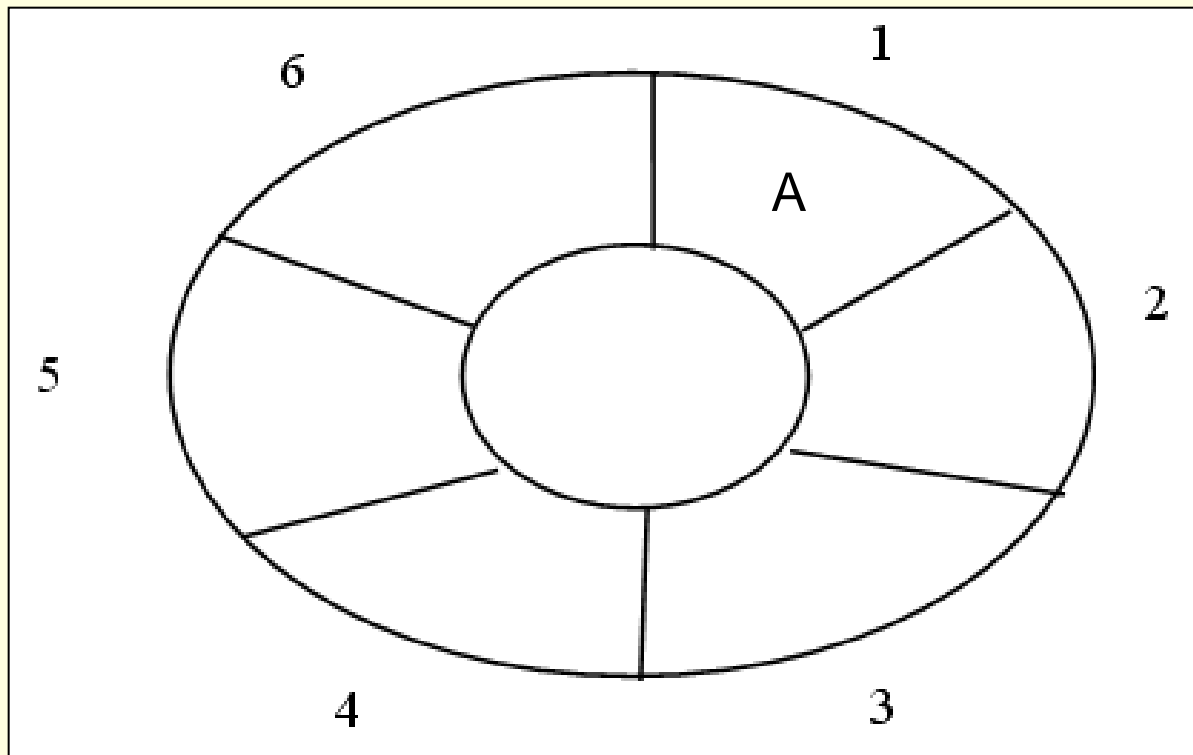
Exemplo

- Fila criada
 - início=1, fim=1, total=0



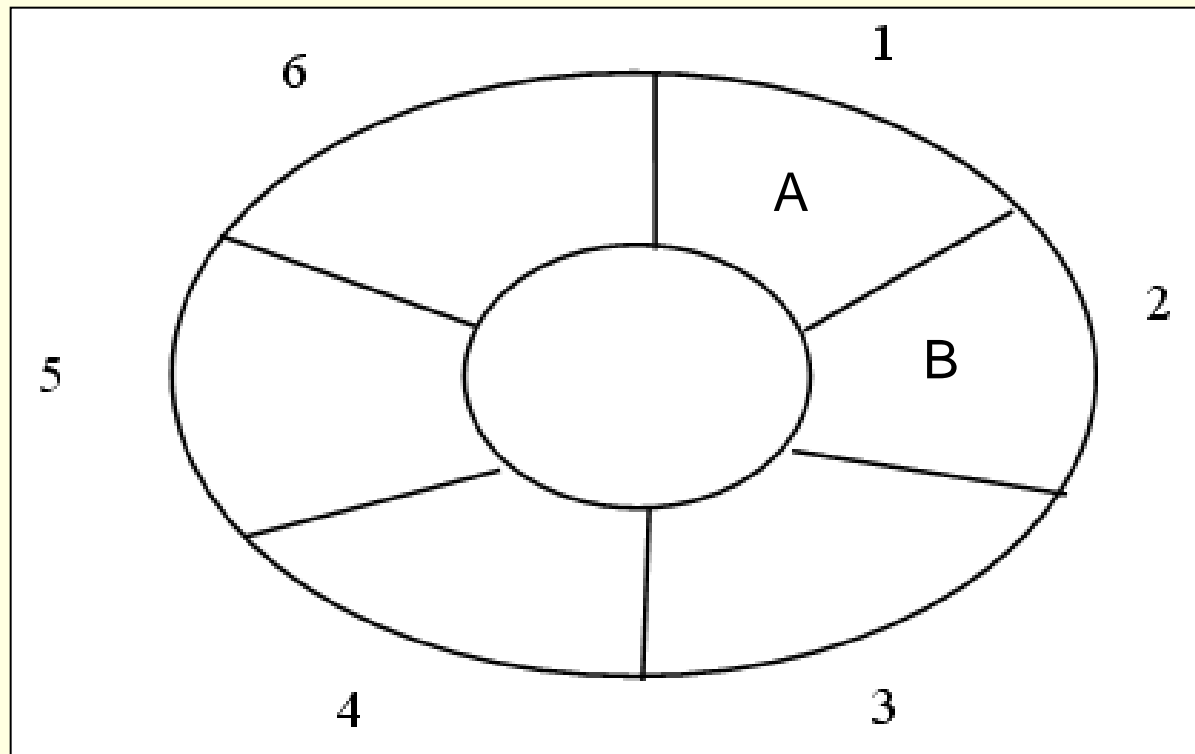
Exemplo

- Entra A
 - início=1, fim= fim+1=2, total=1



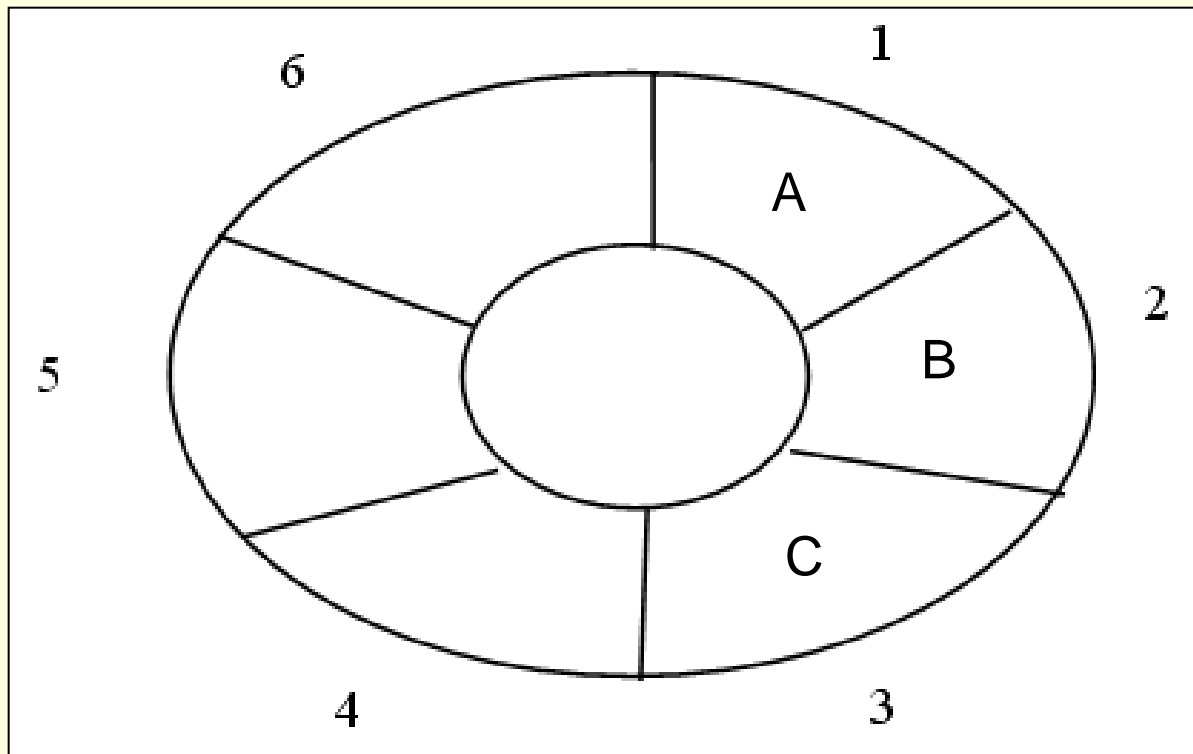
Exemplo

- Entra B
 - início=1, fim=fim+1=3, total=2



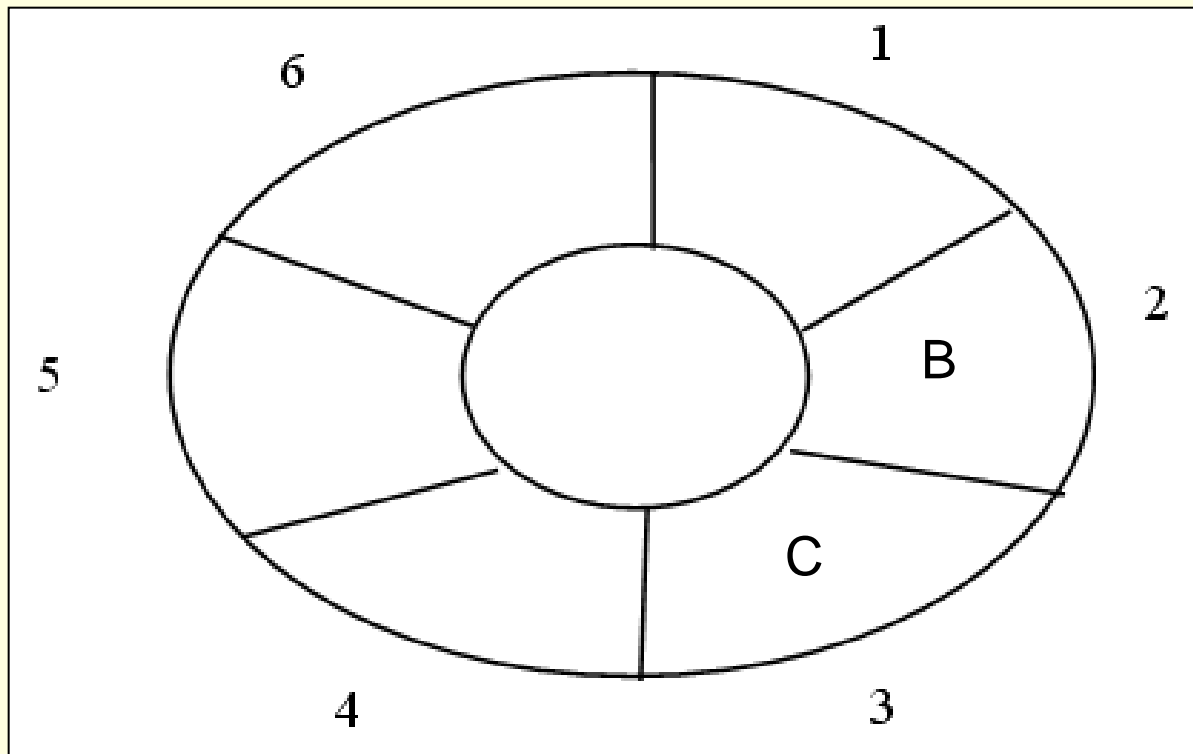
Exemplo

- Entra C
 - início=1, fim=fim+1=4, total=3



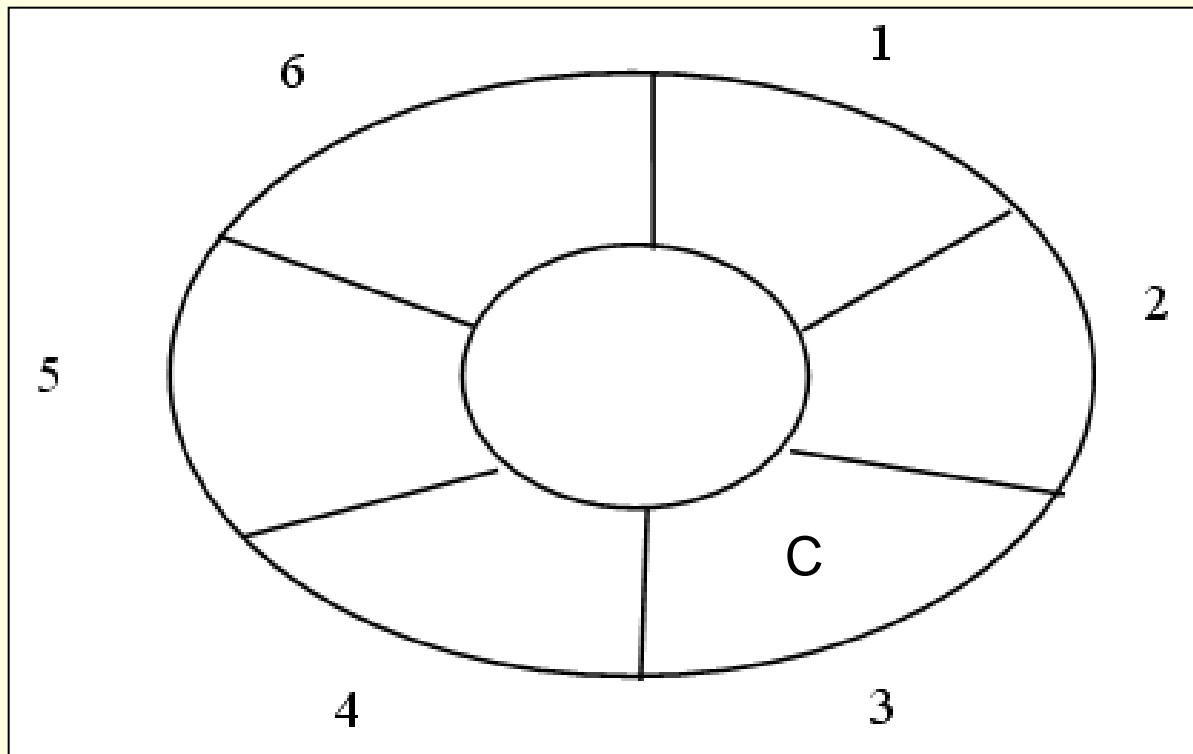
Exemplo

- Sai primeiro
 - início= início+1=2, fim=4, total=2



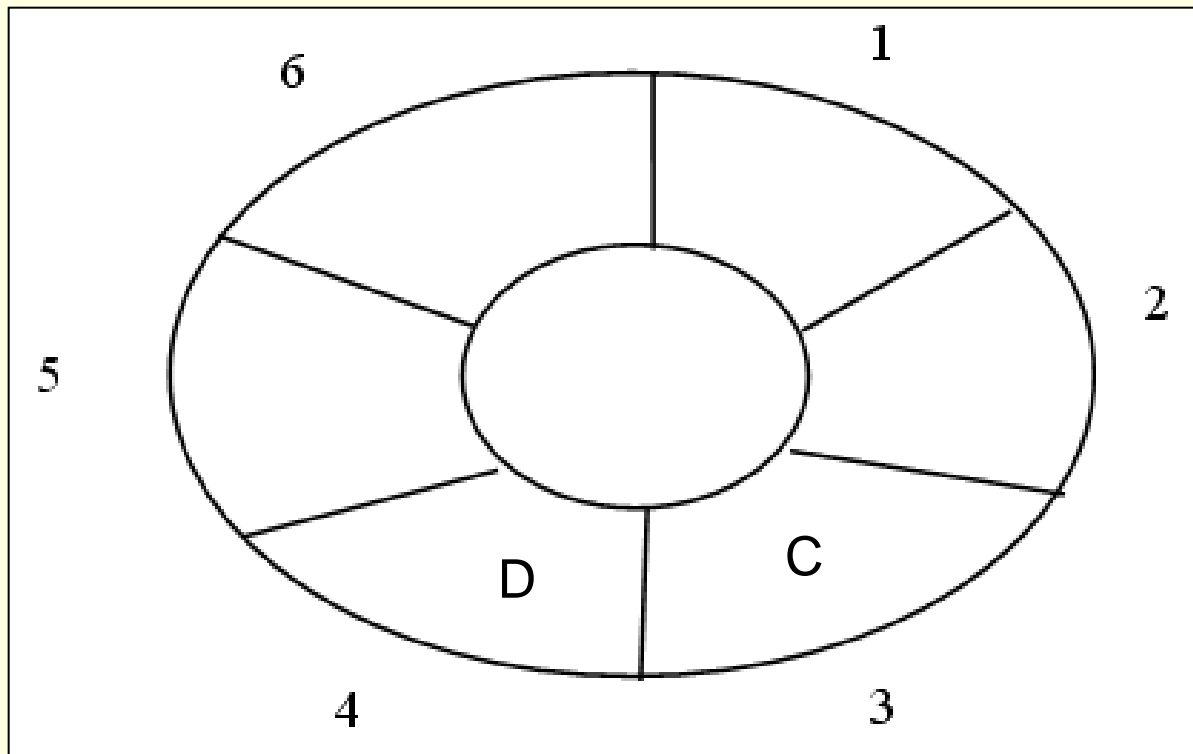
Exemplo

- Sai primeiro
 - início=início+1=3, fim=4, total=1



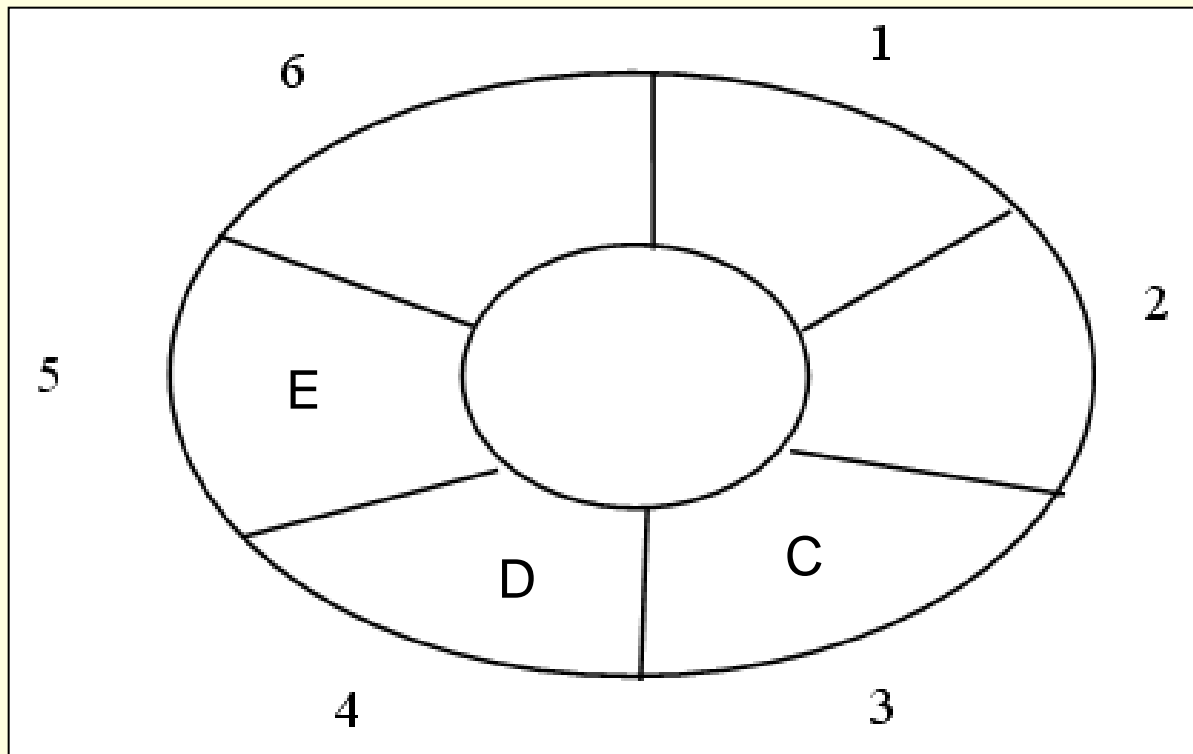
Exemplo

- Entra D
 - início=3, fim=fim+1=5, total=2



Exemplo

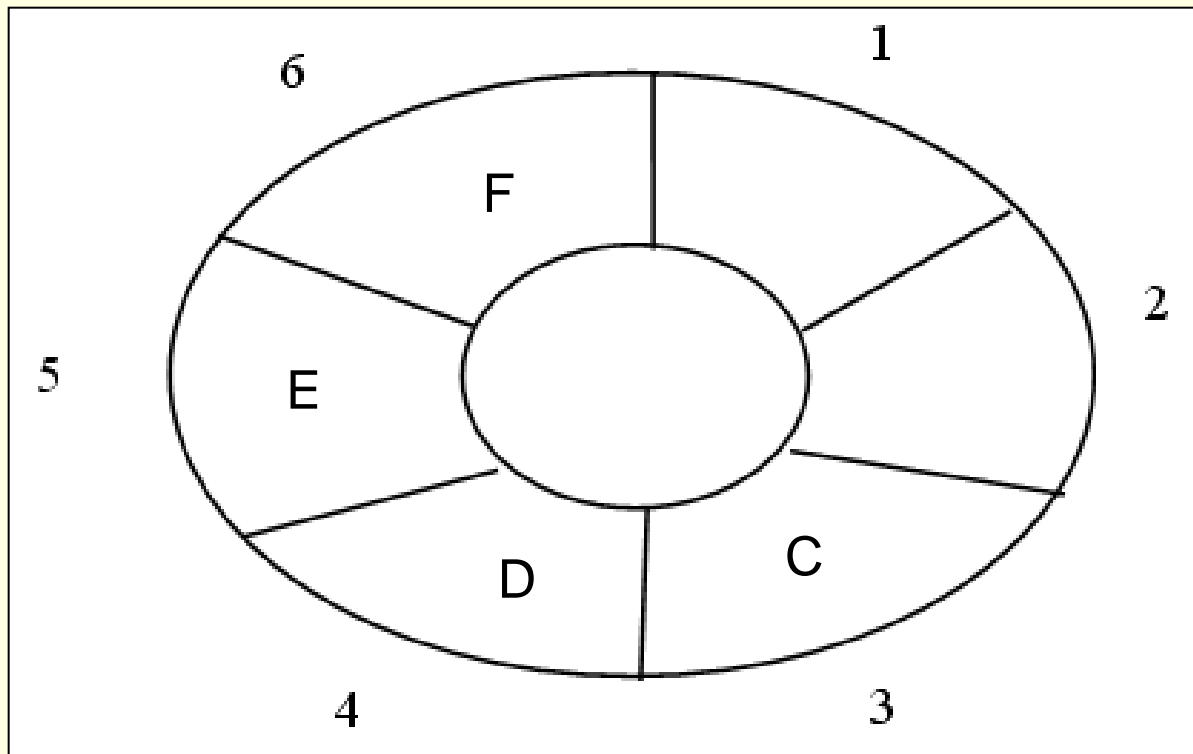
- Entra E
 - início=3, fim=fim+1=6, total=3



Exemplo

- Entra F

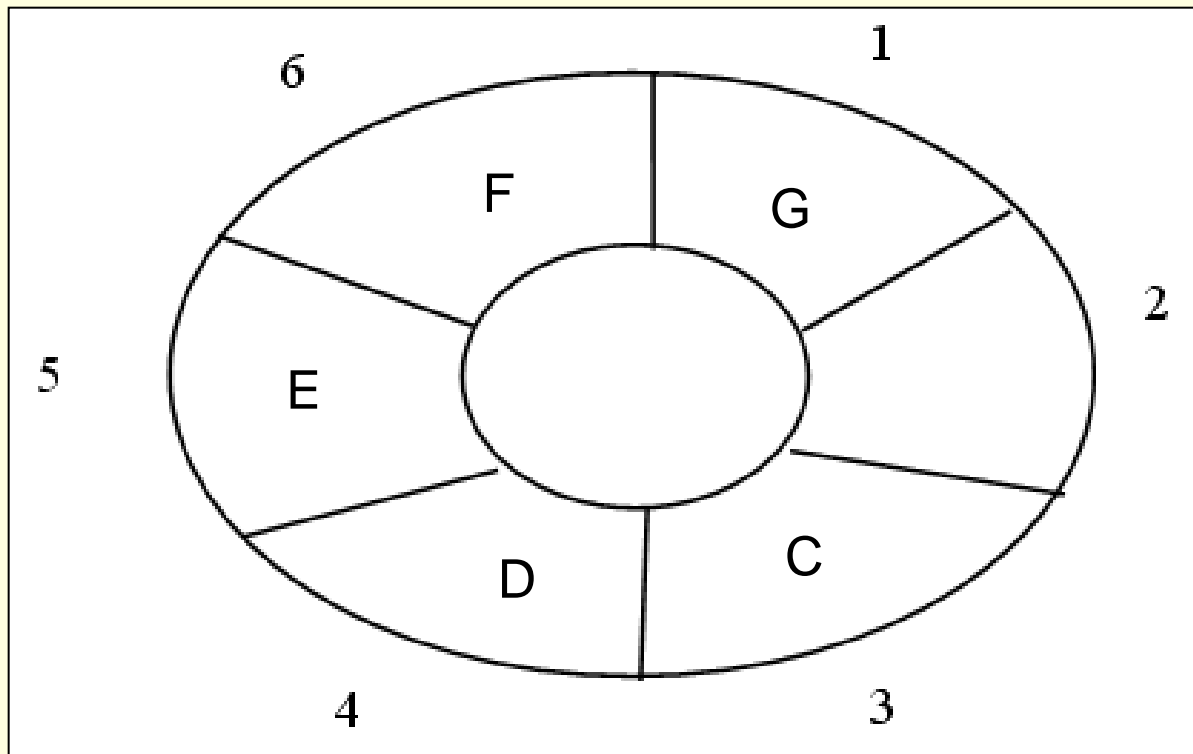
- início=3, fim= (fim+1) % 6 =1, total=4



Exemplo

- Entra G

- Início=3, fim=fim+1=(fim+1) % 6 =2, total=5



Passo a passo para *Entra e Sai*

- **Entra elemento no fim da fila**
 - Se não estiver cheia (Total = tamanho array):
 - `vetor[fim]=elemento`
 - avança fim (“módulo tamanho do array” para “fazer a curva”, se preciso)
 - atualiza total
- **Sai primeiro elemento**
 - Se não estiver vazia (Total \neq 0):
 - `elemento=vetor[início]`
 - avança início (“módulo tamanho do array” para “fazer a curva”, se preciso)
 - atualiza total

Implementação da fila

■ Declaração em C

```
#define TamFila 100

typedef int elem;

typedef struct {
    int inicio, fim, total;
    elem itens[TamFila];
} Fila;

Fila F;
```

Operações sobre a fila

- Implementar as operações
 - Create
 - Empty
 - IsEmpty
 - IsFull
 - Entra
 - Sai

- Atenção: considerações sobre TAD
 - Arquivos .c e .h, parâmetros

Interface do TAD Fila (.h)

```
#define TRUE 1 /*define tipo booleano*/
#define FALSE 0
#define boolean int
#define TamFila 100

typedef char elem;
typedef struct {
    int inicio, fim, total;
    elem A[TamFila];
} Fila;
void Create(Fila*);
void Empty(Fila*);
boolean IsEmpty(Fila*);
boolean IsFull(Fila*);
boolean Entra(Fila*, elem*);
boolean Sai(Fila*, elem* );
```

Implementando as operações: Fila como Anel (.c)

```
#include "fila.h"
#define TRUE 1 /*define tipo booleano*/
#define FALSE 0
#define boolean int
void Create(Fila *F) { /* inicializa fila F como vazia */
    F->inicio=0;
    F->fim=0;
    F->total=0;
    return;
}
void Empty(Fila *F) { /* esvazia logicamente a fila F */
    F->inicio=0;
    F->fim=0;
    F->total=0;
    return;
}
```

```
boolean IsEmpty(Fila *F) { /*verifica se F está vazia */
    return (F->total==0);
}
boolean IsFull(Fila *F) { /*verifica se array está cheio */
    return (F->total==TamFila-1);
}
boolean Entra(Fila *F, elem *X) { /*insere X no início da Fila F, se não estiver cheia*/
    if (!IsFull(F)) {
        F->total++;
        F->A[F->fim]=*X;
        F->fim = (F->fim +1) % TamFila;
        return TRUE;
    }
    return FALSE;
}
```

```
boolean Sai(Fila *F, elem *X) { /*elimina do início da fila e copia valor eliminado em
    X */
    if (!IsEmpty(F)) {
        F->total--;
        *X=F->A[F->inicio];
        F->inicio = (F->inicio +1) % TamFila;
        return TRUE;
    }
    return FALSE;
}
```

Análise da Complexidade – Fila Sequencial (Anel)

- **Acesso a elemento**
 - Apenas no início: tempo constante (via índice “início”) – $O(1)$
 - **Inserção**
 - Apenas no fim: tempo constante (via índice “fim”) – $O(1)$
 - **Eliminação**
 - Apenas no início: tempo constante (via índice “início”) – $O(1)$
- ➔ Portanto, toda operação tem custo **independente** do tamanho da fila
- **Espaço**: exatamente o necessário para armazenar o número de elementos da fila
 - **Desvantagem**: espaço máximo fixado pelo tamanho do array pode ser excessivo ou insuficiente

Exercício

- Faça uma função para verificar se os elementos de uma fila estão ordenados de forma crescente
- Faça uma função que inverta uma fila F1, criando-se uma nova fila F2

Exercício

- Implemente o sistema para a biblioteca usando o TAD fila
 - Cada livro deve ser representado por um registro
 - Nome do livro, disponibilidade, fila de espera
 - Ao requisitar um livro, a pessoa entra na fila de espera se o livro não estiver disponível
 - Quando um livro fica disponível, o primeiro da fila de espera do livro deve receber o livro
 - Implemente as demais funcionalidades (cadastra livro, retira livro, etc.) que julgar necessárias

Filas de Prioridade

- Filas em que a prioridade de remoção não é cronológica
 - Maior prioridade não é do elemento que ingressou primeiro
- Exemplos de Aplicações
 - Vôos lotados (*standby flyers*)
 - Listas de espera em geral (p. ex. transplantes)
 - Fila de processos para o Sistema Operacional

TAD Fila de Prioridade

- Armazena **Itens**
- **Item**: par (chave, informação) – chave determina prioridade
 - p.ex. a idade de uma pessoa na fila do banco
- Operações principais:
 1. Insere no fim e remove o de maior prioridade (tem que fazer uma busca antes da remoção), **OU**
 2. Insere de acordo com a prioridade (buscando o lugar correto onde inserir) e remove do Início