

Fila

SCC0502 – Algoritmos e Estruturas de
Dados I

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?

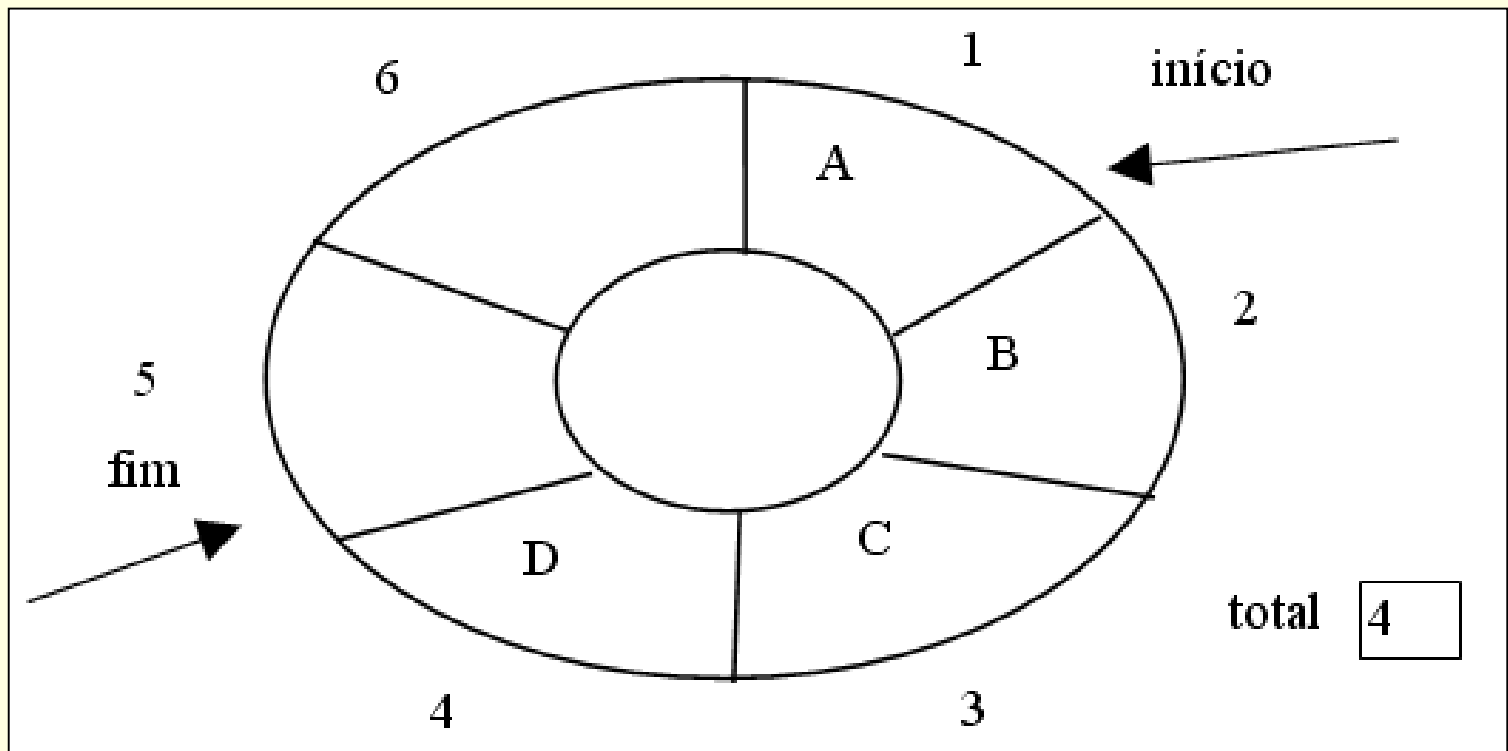
- Modelar situações em que é preciso armazenar um conjunto ordenado de elementos, no qual o primeiro elemento a entrar no conjunto será também o primeiro elemento a sair do conjunto, e assim por diante

- F.I.F.O

- *First In, First Out*

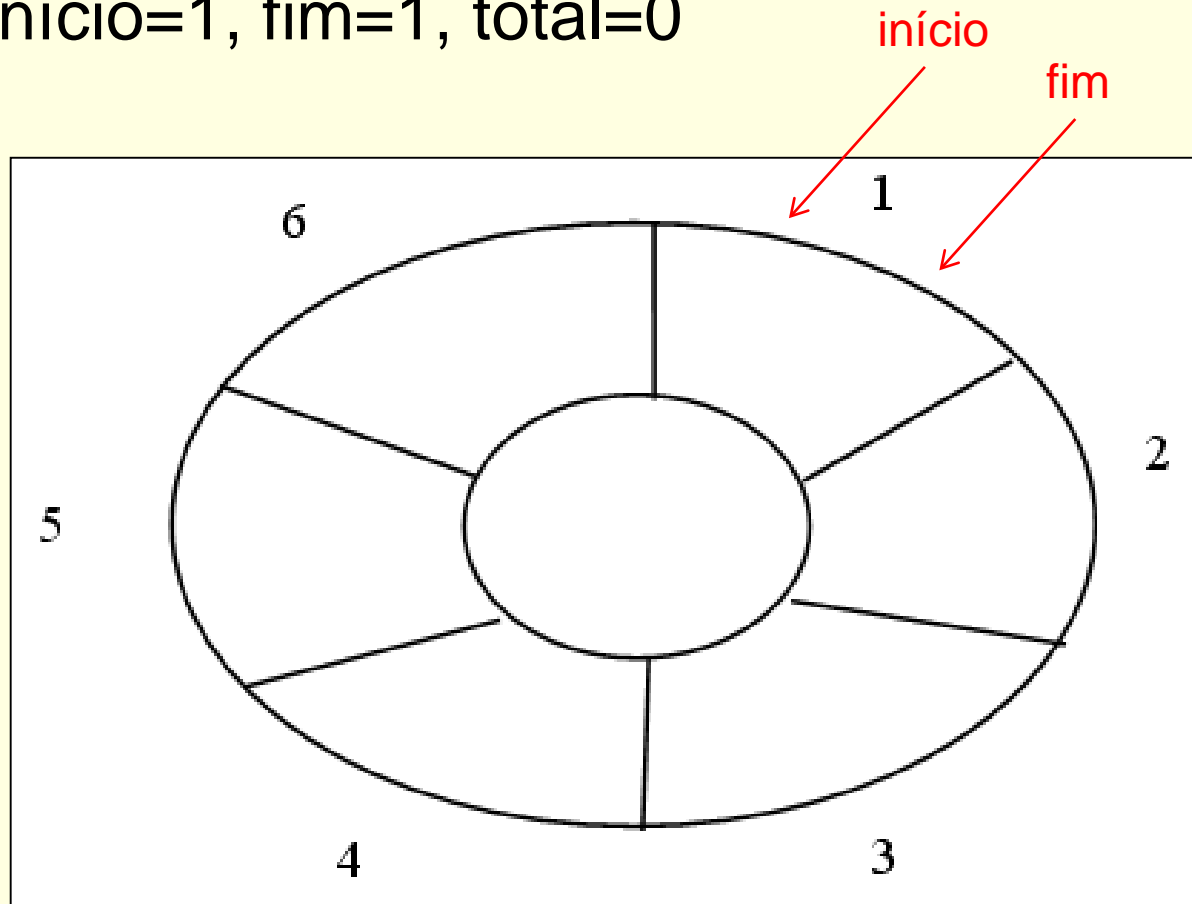
Fila em vetor circular

- Vetor circular



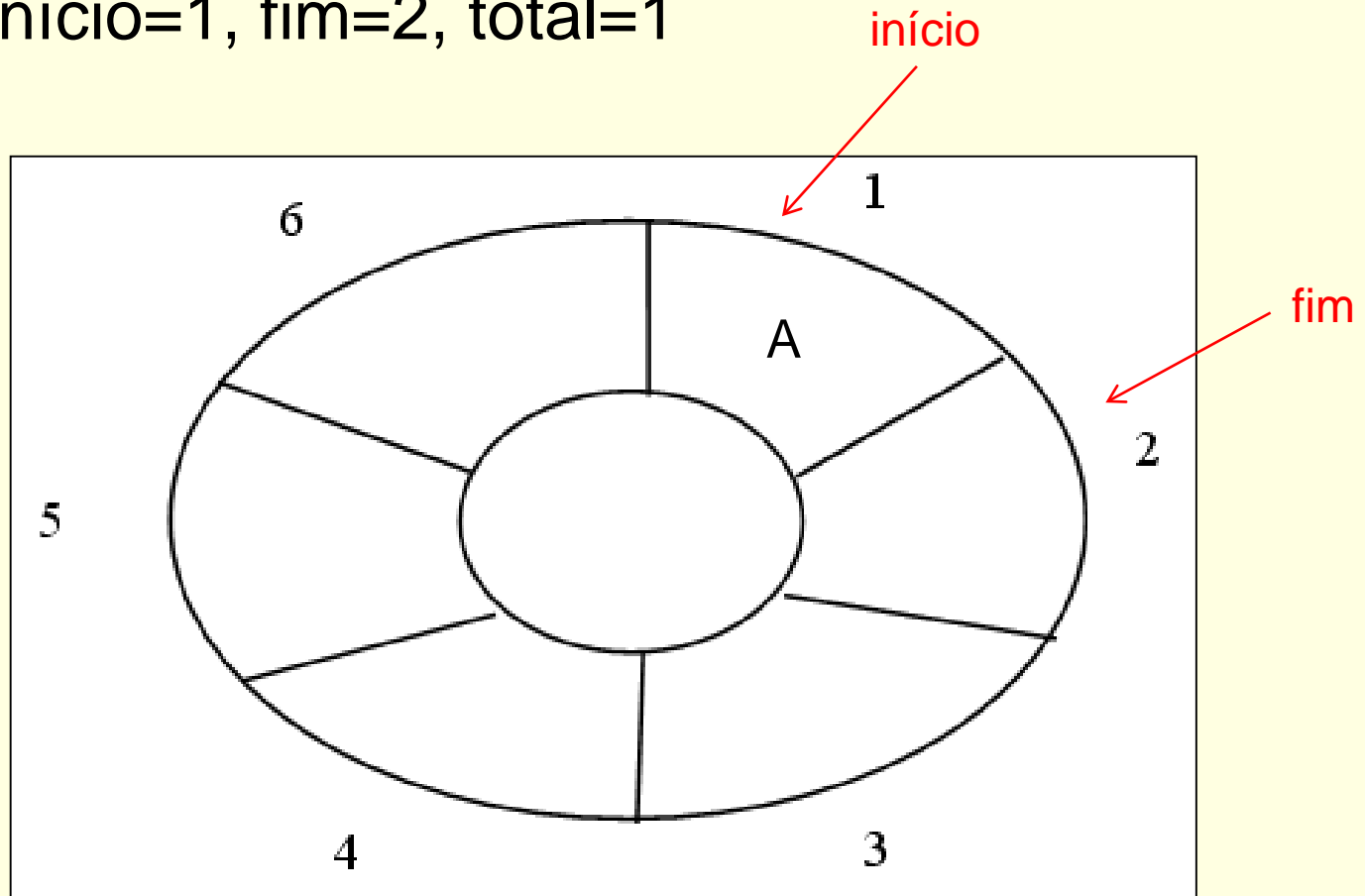
Exemplo

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



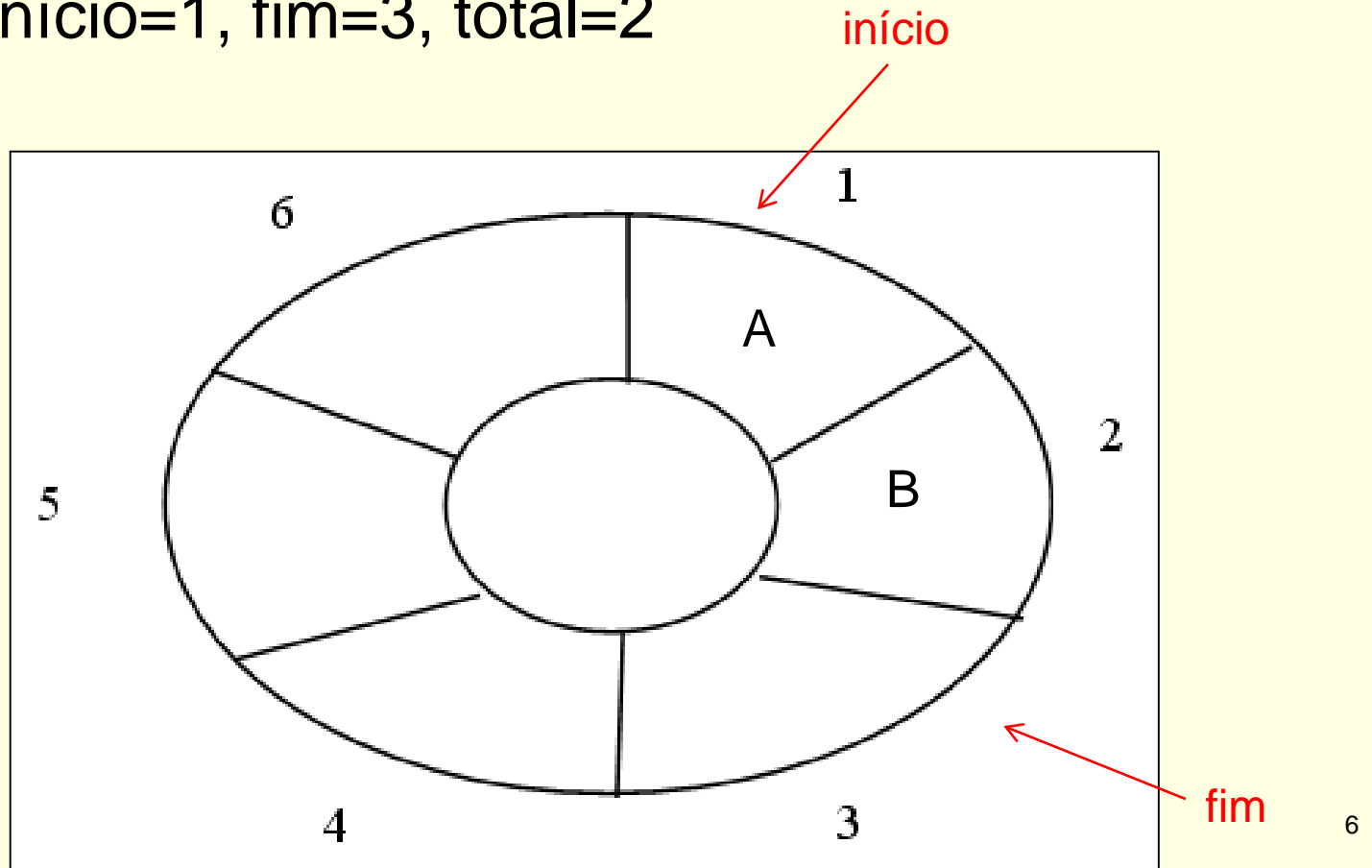
Exemplo

- Entra A
 - Início=1, fim=2, total=1



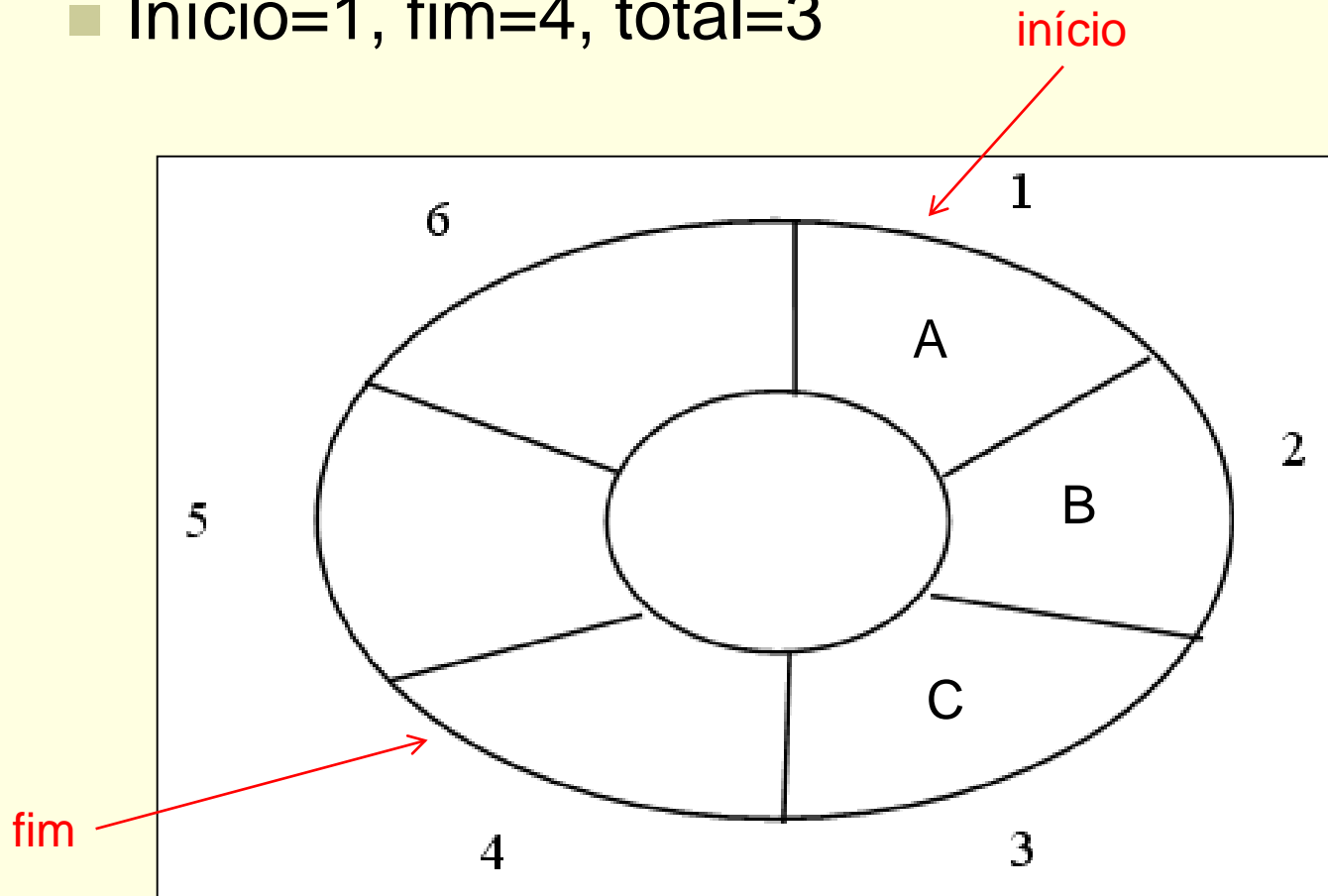
Exemplo

- Entra B
 - Início=1, fim=3, total=2



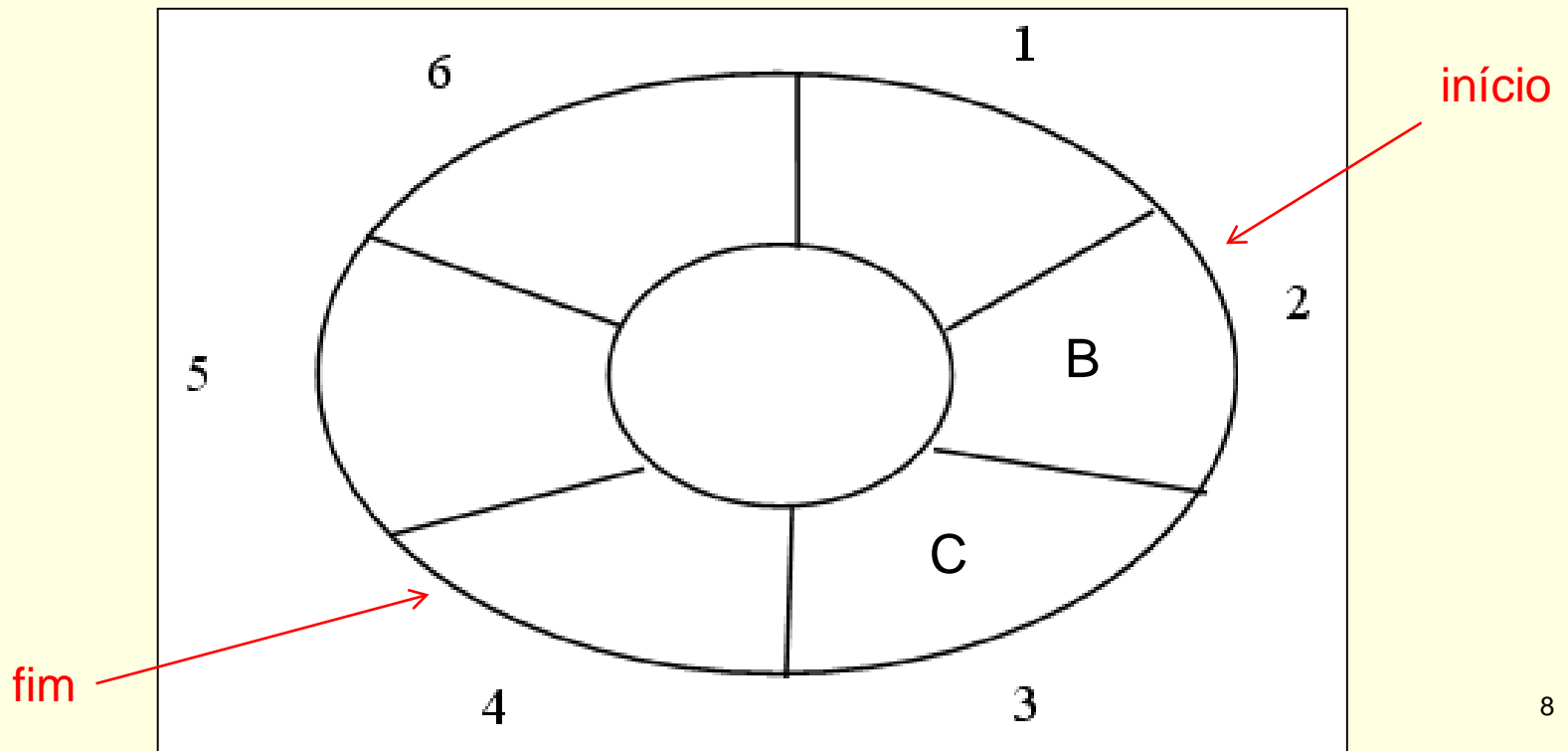
Exemplo

- Entra C
 - Início=1, fim=4, total=3



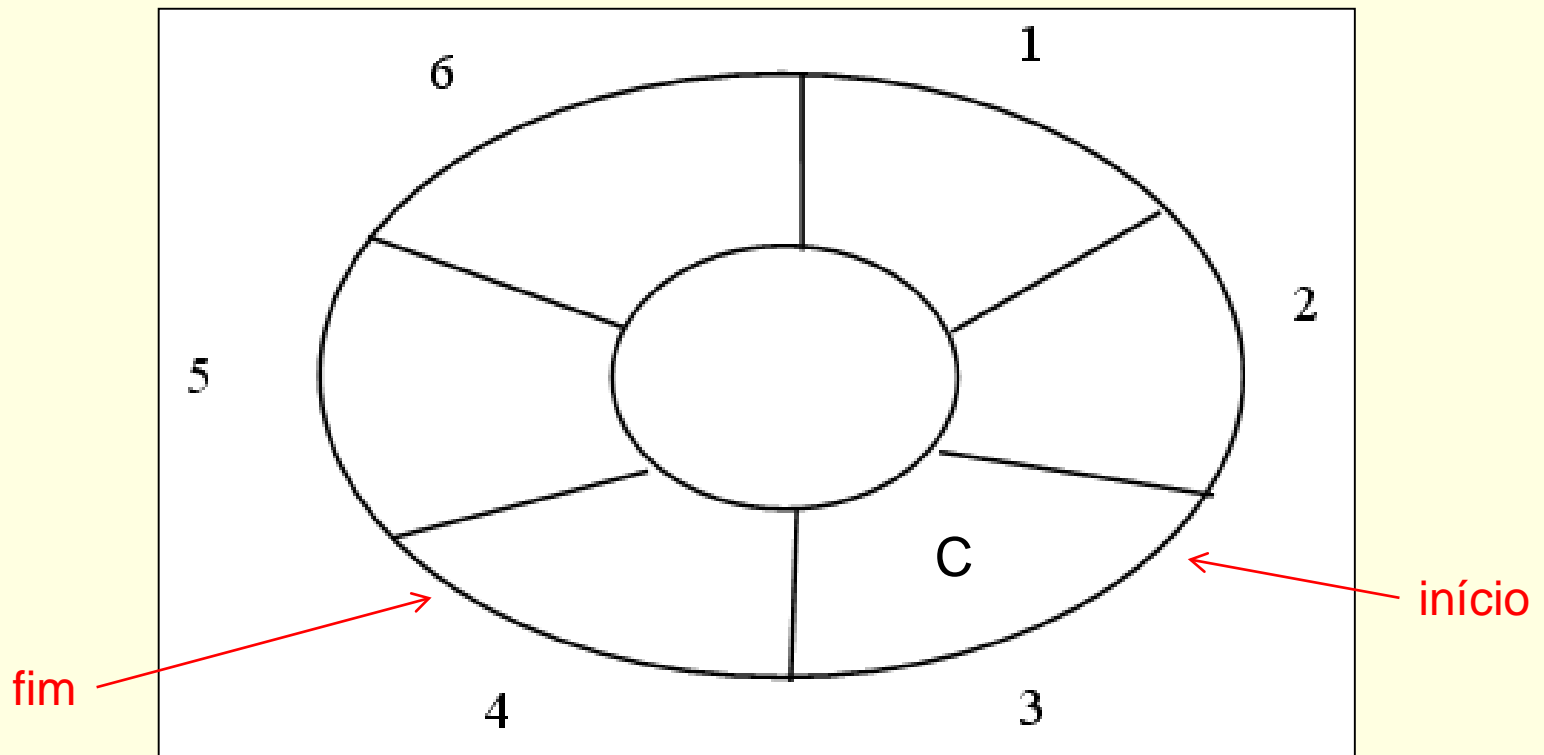
Exemplo

- Sai primeiro
 - Início=2, fim=4, total=2



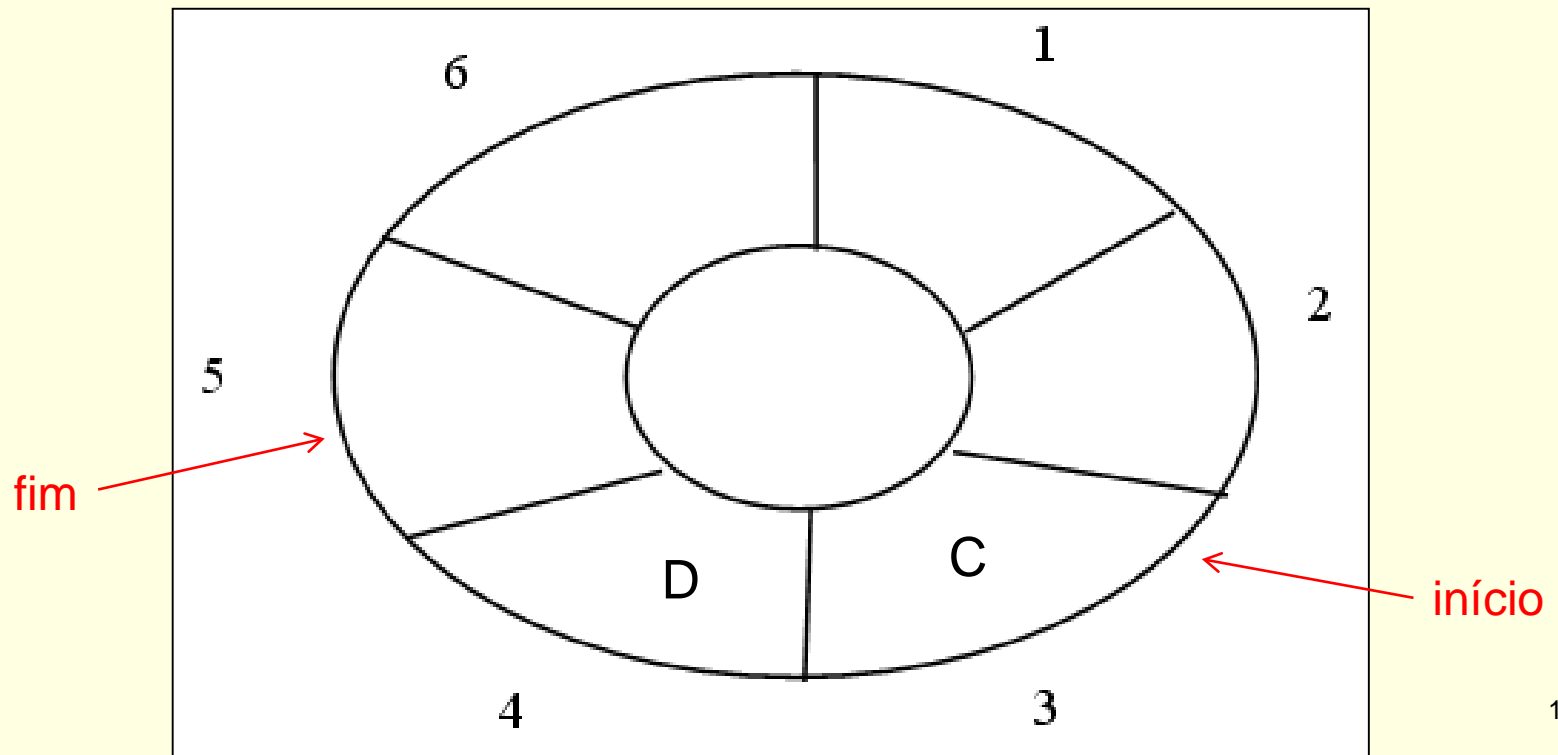
Exemplo

- Sai primeiro
 - Início=3, fim=4, total=1



Exemplo

- Entra D
 - Início=3, fim=5, total=2



Implementação da fila

■ Declaração em C

```
#define TamFila 100

typedef int elem;

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

Exercício

- Faça uma rotina para verificar se os elementos de uma fila estão **ordenados** de forma crescente

Exercício

- Faça uma rotina que **inverta** uma fila F
 - Use uma pilha auxiliar

Exercício

- Faça uma rotina que receba duas filas previamente ordenadas e, a partir da união delas, produza uma **terceira fila ordenada**

Exercício

- Desafio: como criar uma fila “mais genérica” que possa guardar tipos diferentes (inteiros e reais, por exemplo)?
 - TAD ainda melhor!

Exercício

- Considere a situação de uma rotina recursiva que contém uma fila de impressão local de documentos. Quando uma nova execução da rotina é acionada, os dados da execução anterior – incluindo a fila – devem ser empilhados na memória, para que sejam retomados posteriormente.
 - Declare a **estrutura de dados “pilha de filas”** que seria utilizada em situações como essa

Exercício

- Considere a situação de uma rotina recursiva que contém uma fila de impressão local de documentos. Quando uma nova execução da rotina é acionada, os dados da execução anterior – incluindo a fila – devem ser empilhados na memória, para que sejam retomados posteriormente.
 - Declare a estrutura de dados “pilha de filas” que seria utilizada em situações como essa
 - Implemente as funcionalidades básicas de **empilhar uma fila** e **desempilhar uma fila**

Exercício... para pensar

- E como seria uma “**fila de pilhas**”?

Exercício... para pensar

- E como seria uma “fila de pilhas”?
- E uma “**fila de filas**”?

Exercício para casa

- 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

Um começo...

- 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
 - Como é a declaração dessa estrutura de dados?

Um começo...

■ Possível declaração

```
#define NroLivros 1000
#define TamFila 100

typedef struct {
    char nome_pessoa[50];
    int telefone[20];
} elem;

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

```
typedef struct {
    char nome_livro[100];
    int disponivel;
    Fila fila_espera;
} Livro

Livro Biblioteca[NroLivros];
```