



Algoritmos e Estruturas de Dados I

Prof^a Priscilla Abreu
priscilla.abreu@ime.uerj.br
2022.1

Algoritmos e Estruturas de Dados I



Roteiro da aula

- Listas lineares
 - Listas Sequenciais Particulares
 - Filas



Revisando...

Algoritmos e Estruturas de Dados I



Listas Linear

Listas lineares

Listas lineares gerais

SEM restrição de inserção e remoção de elementos

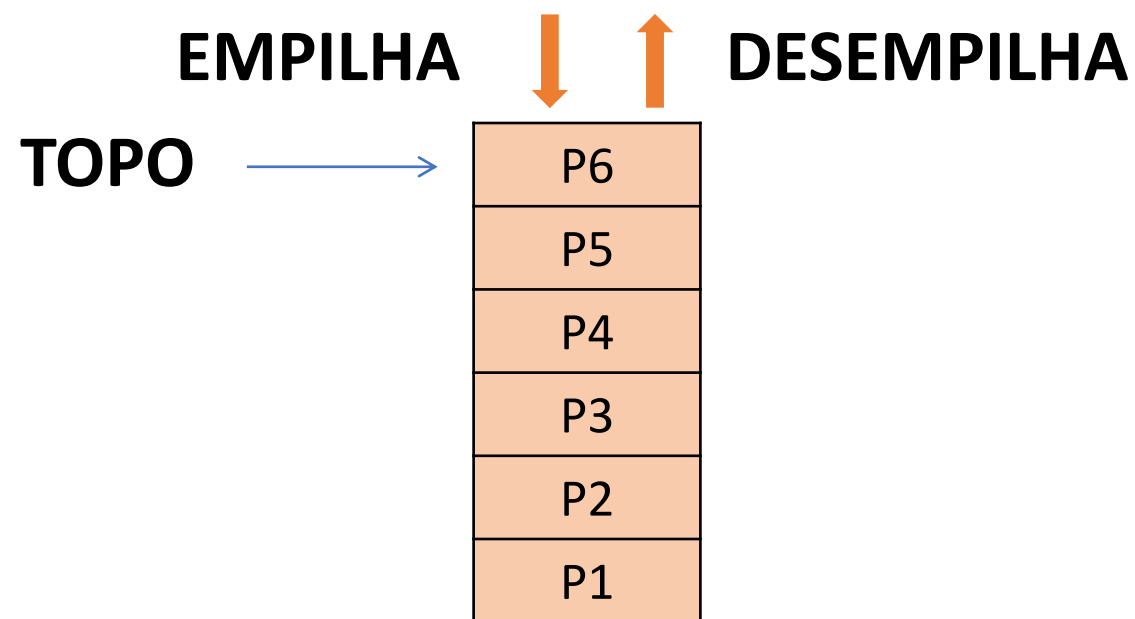
Listas particulares

COM restrição de inserção e remoção de elementos

Algoritmos e Estruturas de Dados I



Pilha





Filas

Algoritmos e Estruturas de Dados I



Fila

- São listas em que todas as inserções ocorrem em uma extremidade e as remoções em outra extremidade;
- Estruturas de dados do tipo FIFO (first-in first-out): o primeiro elemento a ser inserido, será o primeiro a ser removido.
- Exemplos: filas de banco, supermercado, fila de impressão de arquivos , etc.



Nome do documento	Estado	Proprietário	Páginas	Tamanho	Submetido	Porta
Sem título - Bloco de notas	Erro - Impressão	XGuest	1	824 bytes/932 bytes	21:18:05 19-11-2009	LPT1:
Sem título - Bloco de notas		XGuest	1	932 bytes	21:18:32 19-11-2009	
Sem título - Bloco de notas		XGuest	1	932 bytes	21:18:42 19-11-2009	
Sem título - Bloco de notas		XGuest	1	932 bytes	21:18:43 19-11-2009	
Sem título - Bloco de notas		XGuest	1	932 bytes	21:18:44 19-11-2009	
Sem título - Bloco de notas		XGuest	1	932 bytes	21:18:45 19-11-2009	
Sem título - Bloco de notas		XGuest	1	932 bytes	21:18:46 19-11-2009	
Sem título - Bloco de notas		XGuest	1	932 bytes	21:18:47 19-11-2009	

8 documento(s) em fila de espera



Algoritmos e Estruturas de Dados I



Fila



↑
INÍCIO

↑
FIM

Início da fila: extremidade onde ocorrem as remoções.

Final da fila: extremidade onde ocorrem as inserções.

Algoritmos e Estruturas de Dados I



Fila – aplicações

- Fila de arquivos para impressão;
- Atendimento de processos requisitados ao um sistema operacional;
- Processos de reserva e compra online;
- Buffer para gravação de dados em mídia;
- Processos de comunicação em redes de computadores.

Algoritmos e Estruturas de Dados I



Fila – operações

- Alocação sequencial:
 - Uso de vetores;
 - Variáveis controladoras para início e fim da fila:
 - Início e fim.
- Operações básicas:
 - Inserção
 - Remoção
- Situações extremas:
 - Fila cheia
 - Fila vazia

Algoritmos e Estruturas de Dados I



Fila – informações iniciais

i: posição do elemento que está no início da fila;

f: posição do último elemento inserido na fila;

M: tamanho máximo do vetor que representa a fila.

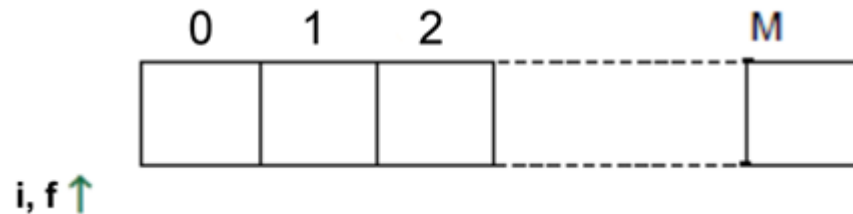
fila: estrutura que representa a fila.

Algoritmos e Estruturas de Dados I



Fila – funcionamento

Situação 1: Fila vazia



Fila: vetor de tamanho M

i : início da fila

f : final da fila

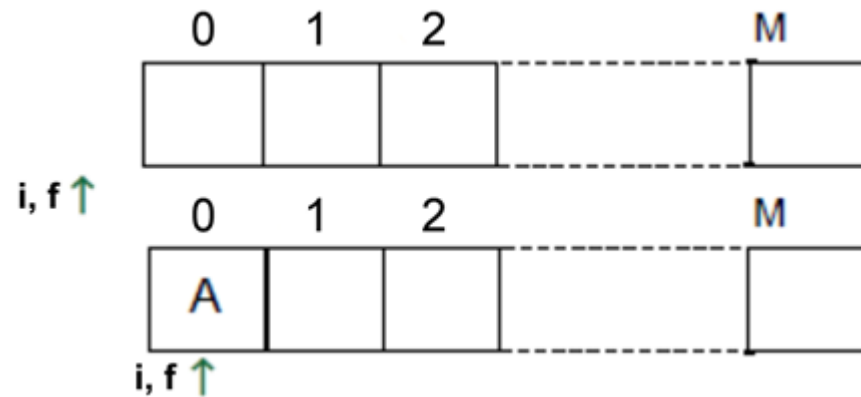
$i = f = -1$

Algoritmos e Estruturas de Dados I



Fila – funcionamento

Situação 2: Insere(A)



Fila: vetor de tamanho M

i: início da fila

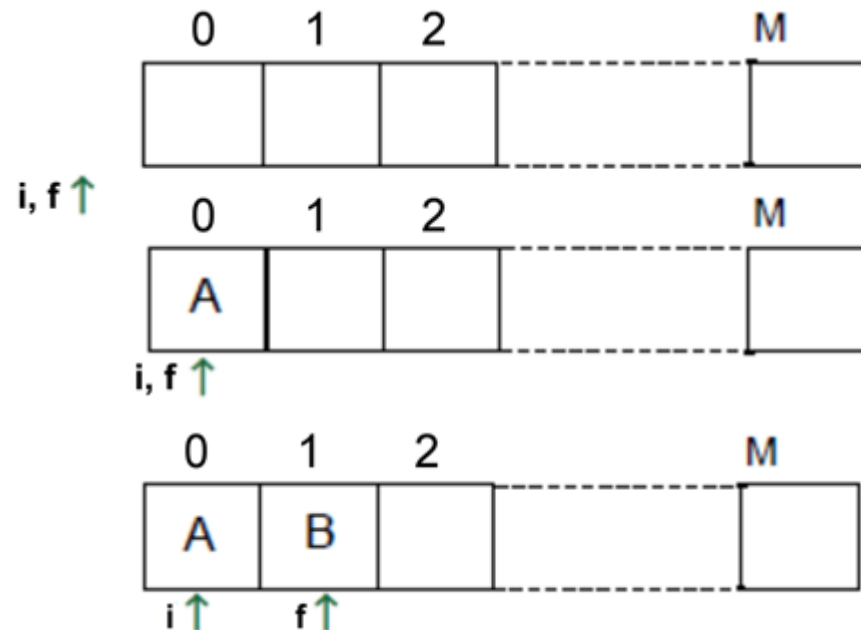
f: final da fila

Algoritmos e Estruturas de Dados I



Fila – funcionamento

Situação 3: Insere(B)



Fila: vetor de tamanho M

i: início da fila

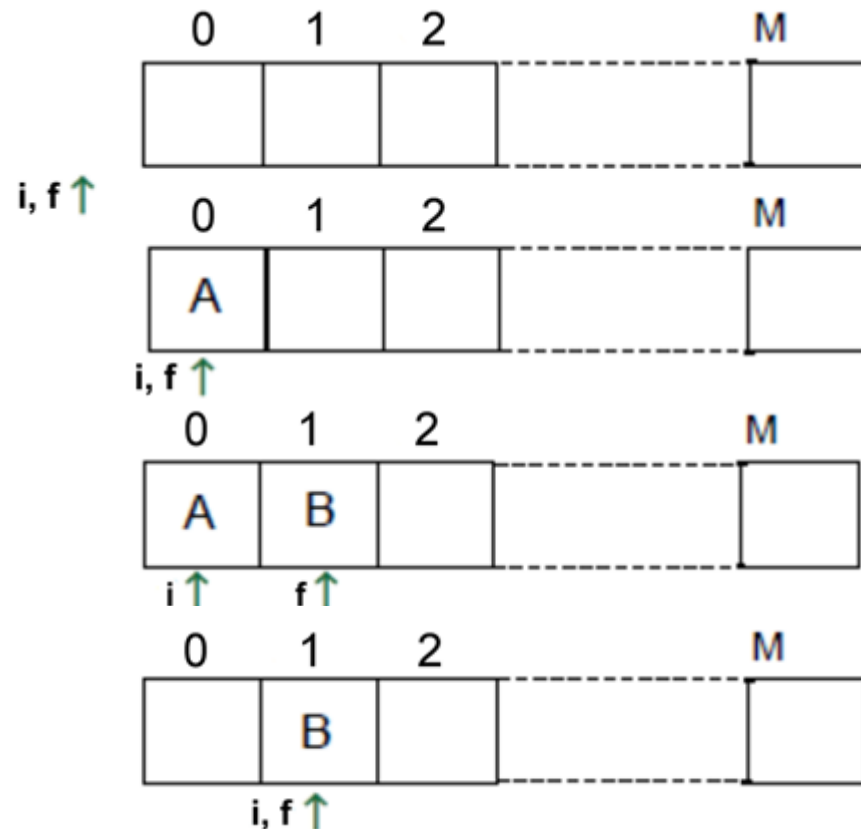
f: final da fila

Algoritmos e Estruturas de Dados I



Fila – funcionamento

Situação 4: Remove()



Fila: vetor de tamanho M

i: início da fila

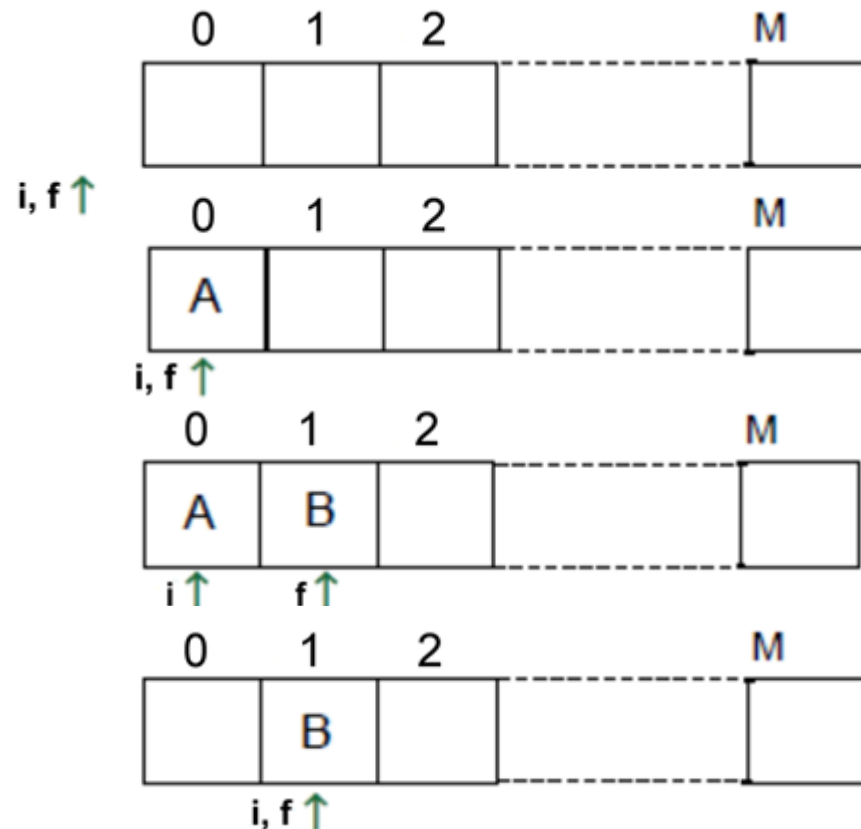
f: final da fila

Algoritmos e Estruturas de Dados I



Fila – funcionamento

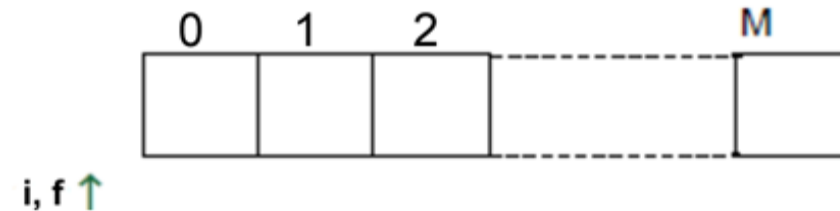
Situação 5: Remove()



Fila: vetor de tamanho M

i : início da fila

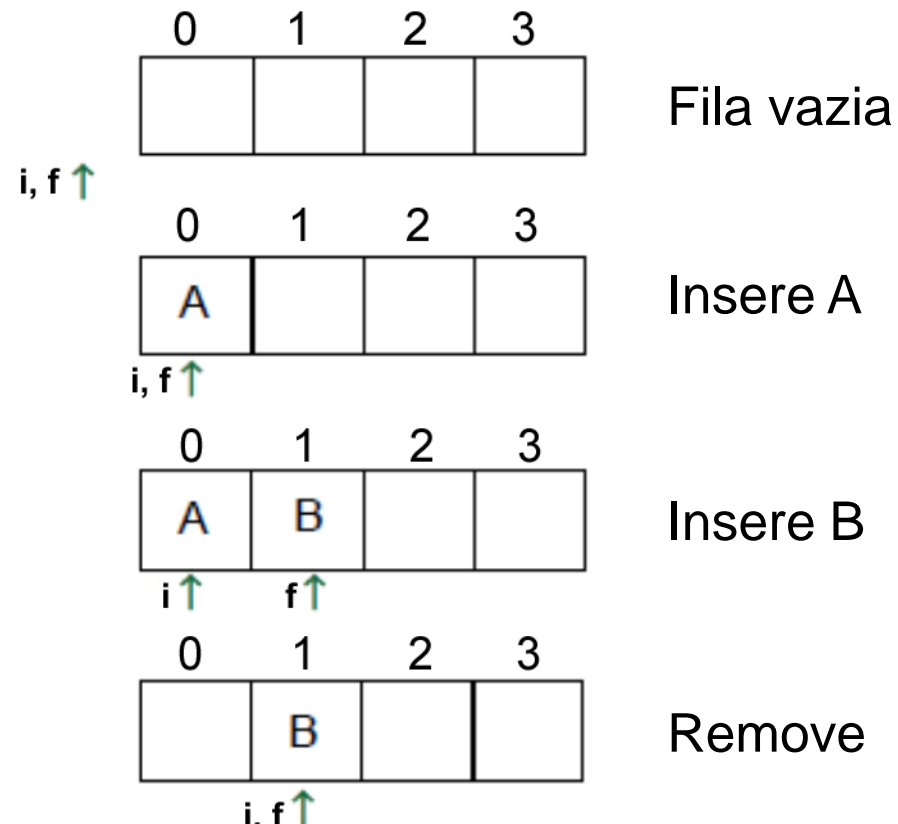
f : final da fila



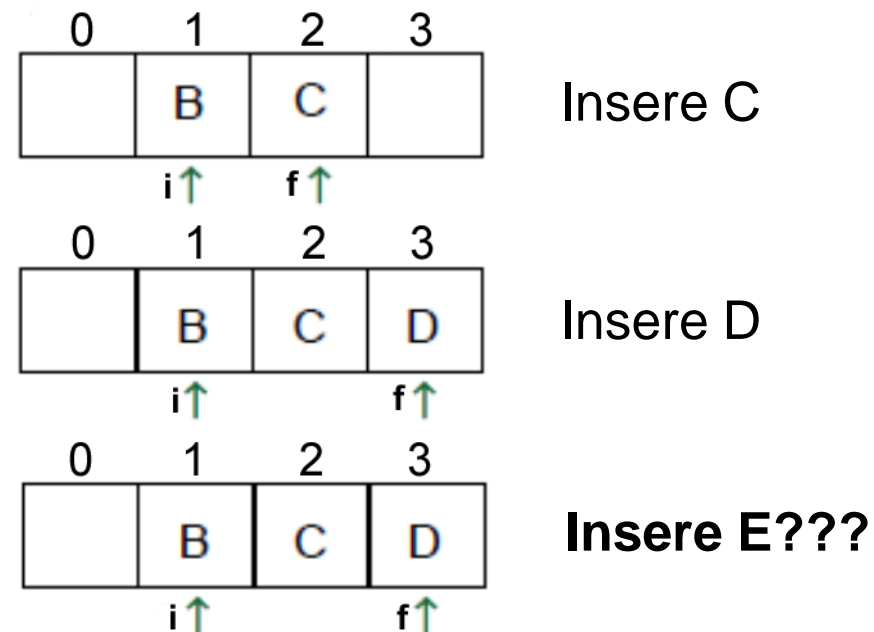
Algoritmos e Estruturas de Dados I



Fila – funcionamento



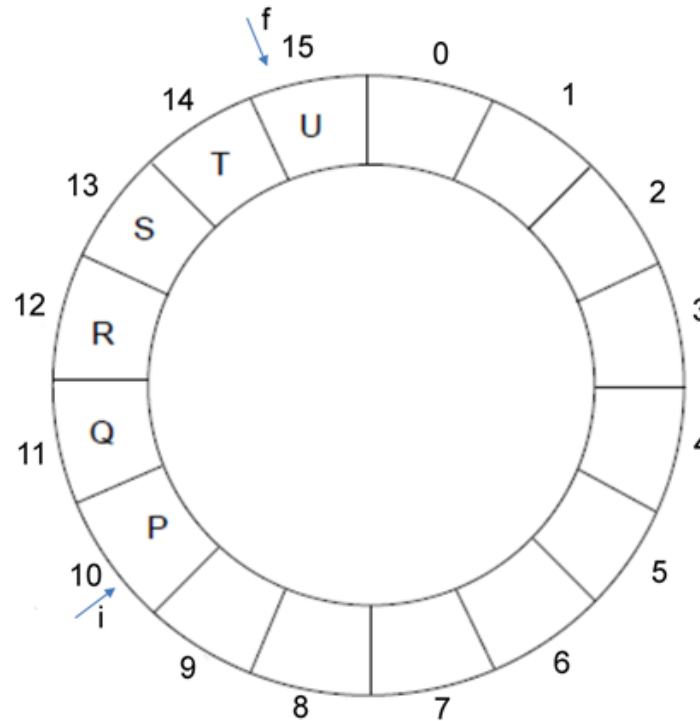
Fila: vetor de tamanho M
i: início da fila
f: final da fila



Algoritmos e Estruturas de Dados I



Fila – Uso da fila circular!



Algoritmos e Estruturas de Dados I



Fila – circular

i: posição do elemento que está no início da fila;

f: posição do último elemento inserido na fila;

n: número de elementos na fila.

M: tamanho máximo do vetor que representa a fila.

fila: estrutura que representa a fila.

Algoritmos e Estruturas de Dados I



Pilha – implementação

estrutura fila:

valores [1..M]: inteiro

i, f, n: inteiro

```
const int M = 10;
```

```
typedef struct filaSeq{
```

```
    int valores[MAX];
```

```
    int i, f, n;
```

```
} filaSeq;
```

Algoritmos e Estruturas de Dados I



Fila – Uso da fila circular!

O que fazer para
enfileirar???

$n = 0$

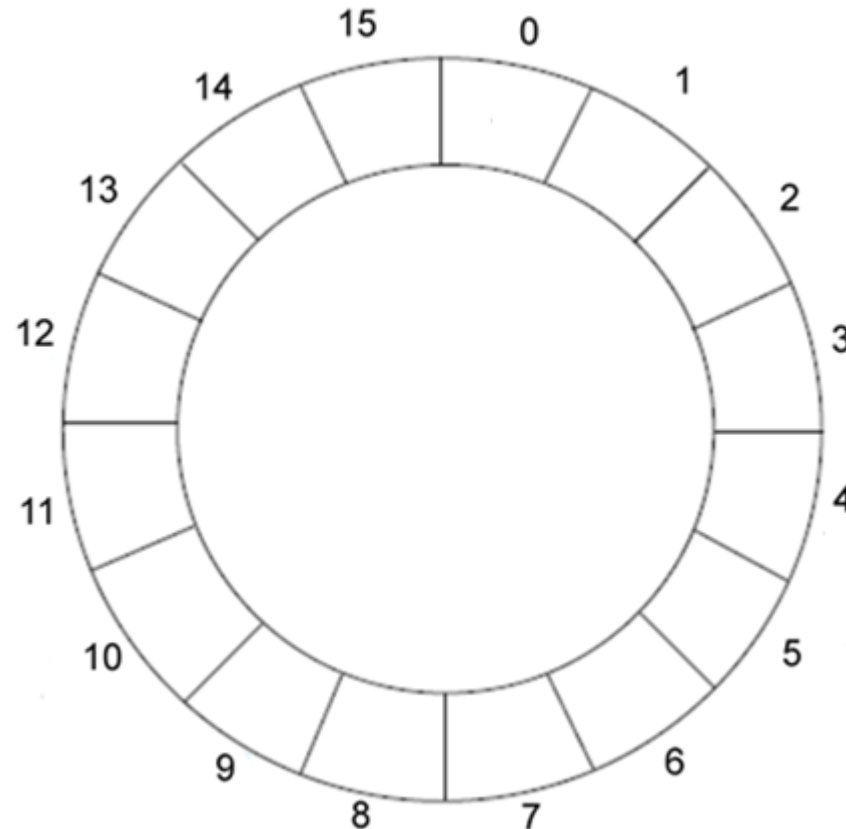
$i = -1$

$f = -1$

$M = 16$

$f++;$

$n++;$



Algoritmos e Estruturas de Dados I



Fila – Uso da fila circular!

$n = 1$

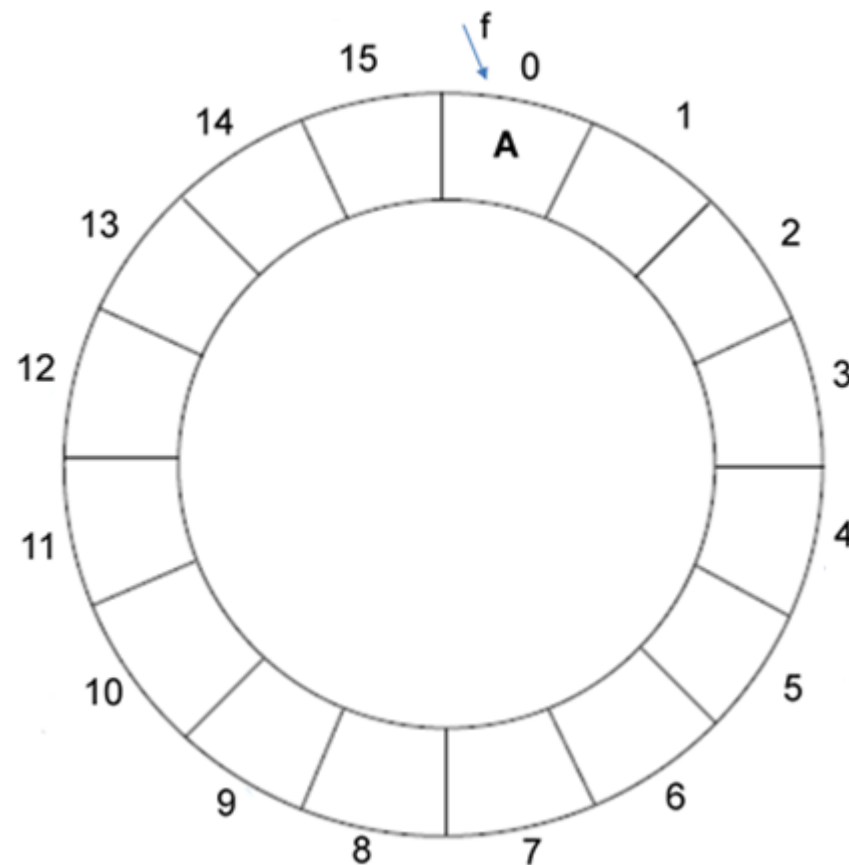
$i = -1$

$f = 0$

$M = 16$

$\text{fila}[f] = \text{valor};$

$i++$



Algoritmos e Estruturas de Dados I



Fila – Uso da fila circular!

$n = 1$

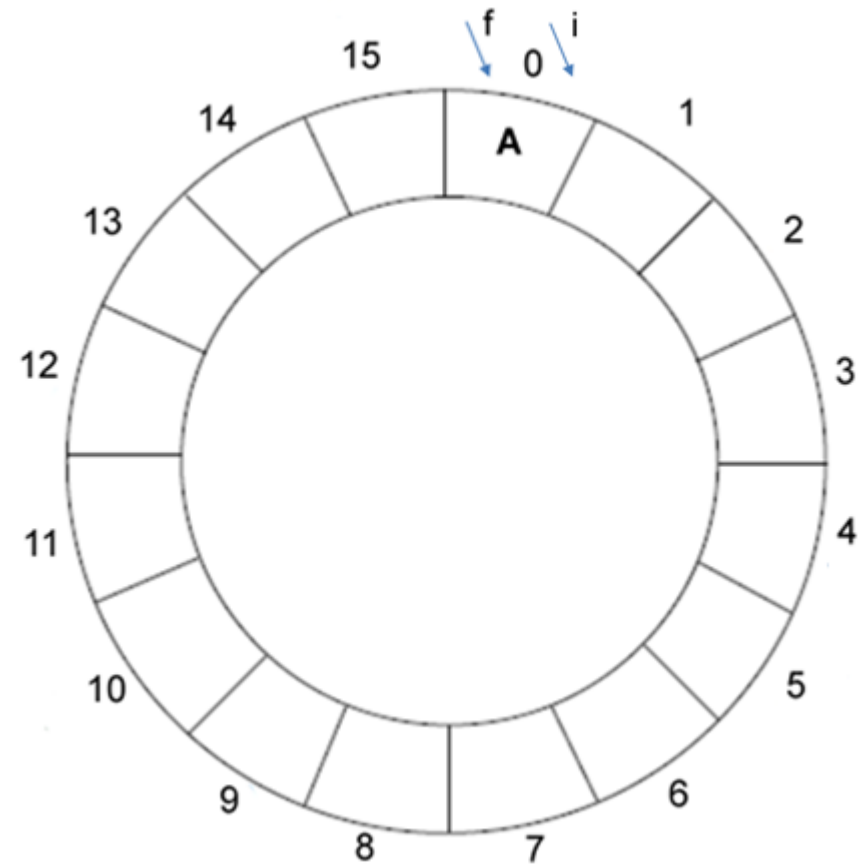
$i = 0$

$f = 0$

$M = 16$

**Inserindo
novo valor**

$f++;$



Algoritmos e Estruturas de Dados I



Fila – Uso da fila circular!

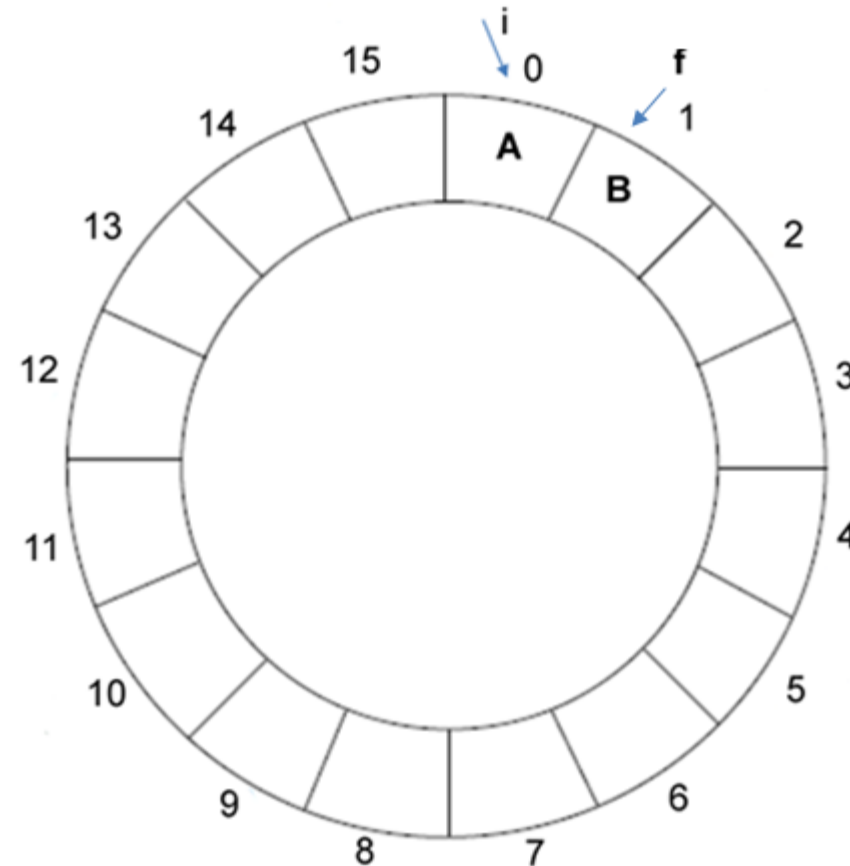
$n = 1$

$i = 0$

$f = 0$

$M = 16$

$\text{fila}[f] = \text{valor};$



Algoritmos e Estruturas de Dados I

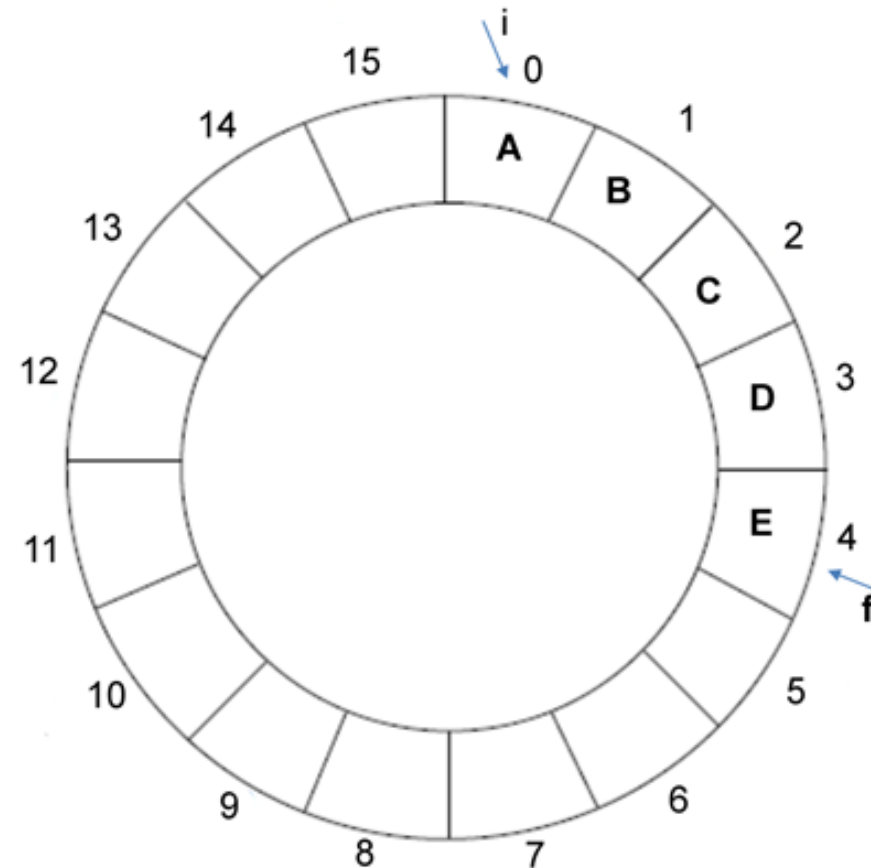


Fila – Uso da fila circular!

$n = 5$
 $i = 0$
 $f = 4$
 $M = 16$

Após algumas
inserções...

$f++;$



Algoritmos e Estruturas de Dados I



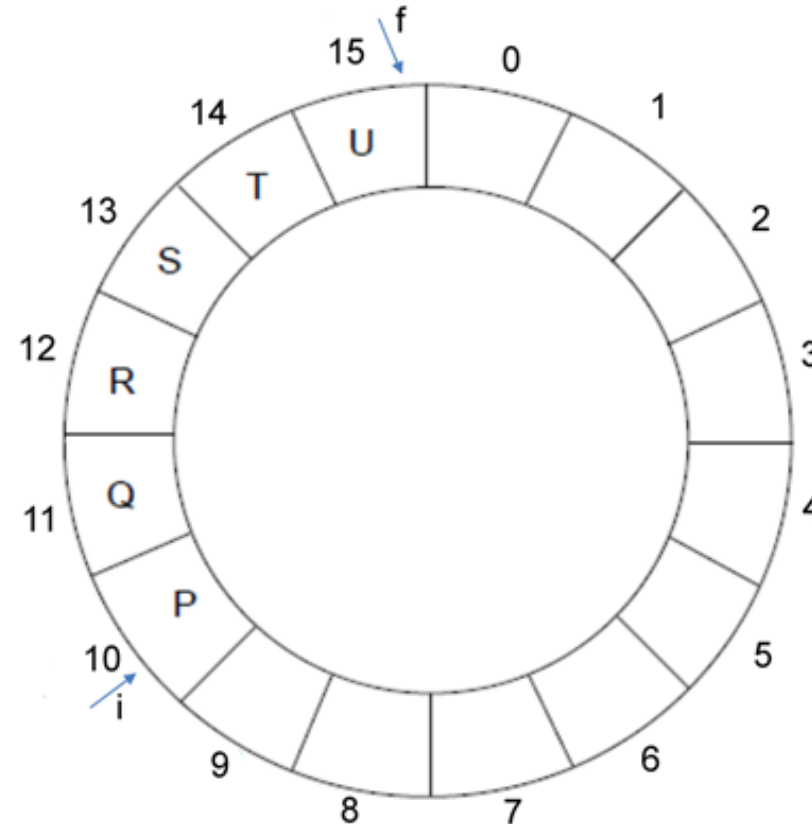
Fila – Uso da fila circular!

$n = 6$
 $i = 10$
 $f = 15$
 $M = 16$

**Mais inserções
e remoções...**

Como enfileirar?

$f++;$



Algoritmos e Estruturas de Dados I



Fila – Uso da fila circular!

$n = 6$

$i = 10$

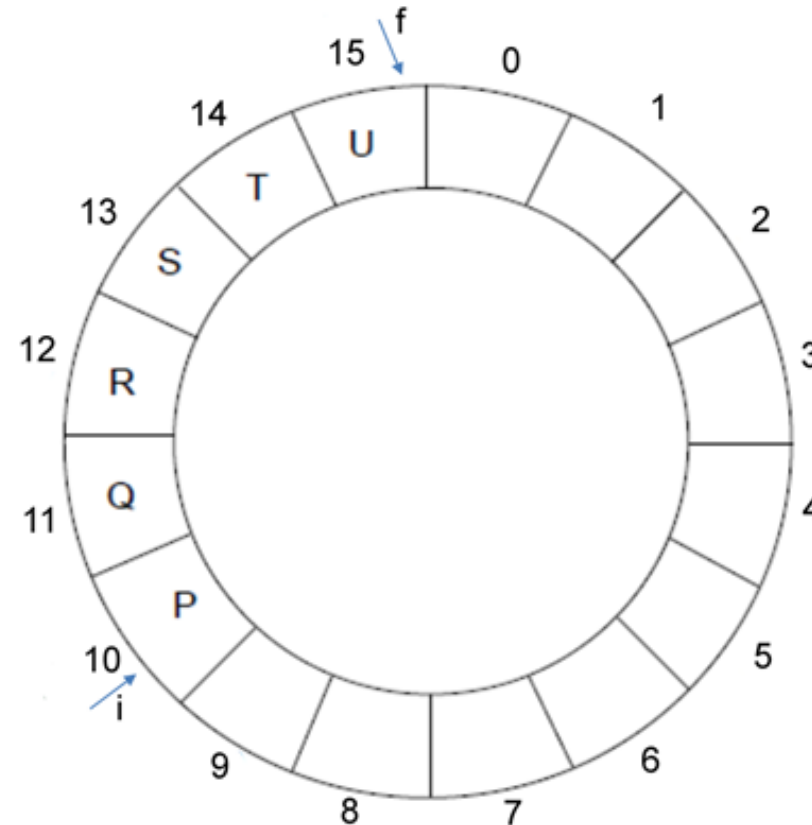
$f = 15$

$M = 16$

**Mais inserções
e remoções...**

Como enfileirar?

~~**$f++;$**~~



Algoritmos e Estruturas de Dados I



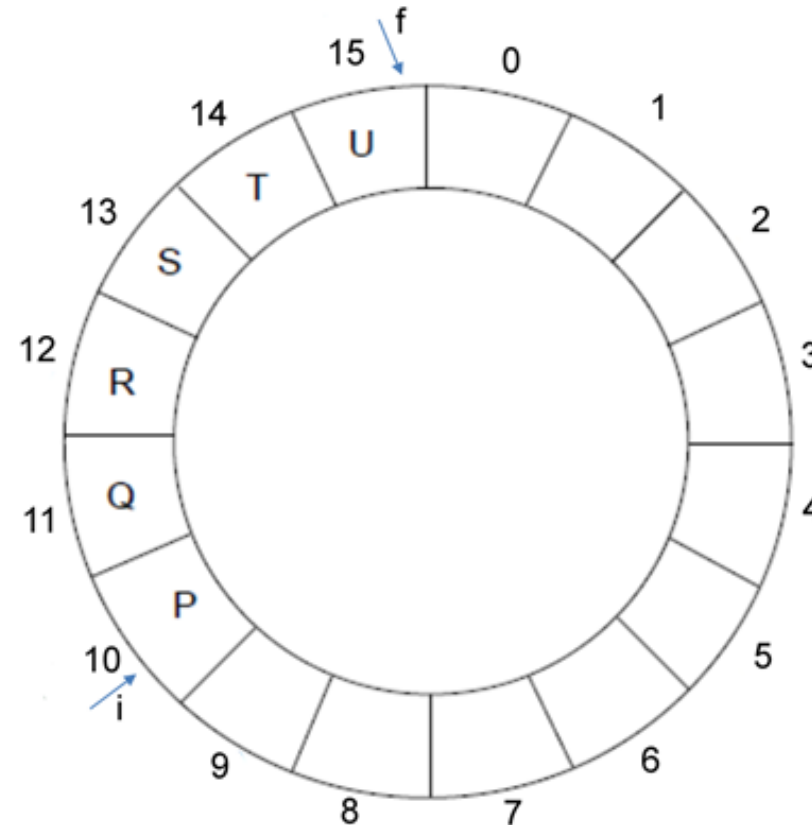
Fila – Uso da fila circular!

$n = 6$
 $i = 10$
 $f = 15$
 $M = 16$

$f == M-1$?



$f = 0;$

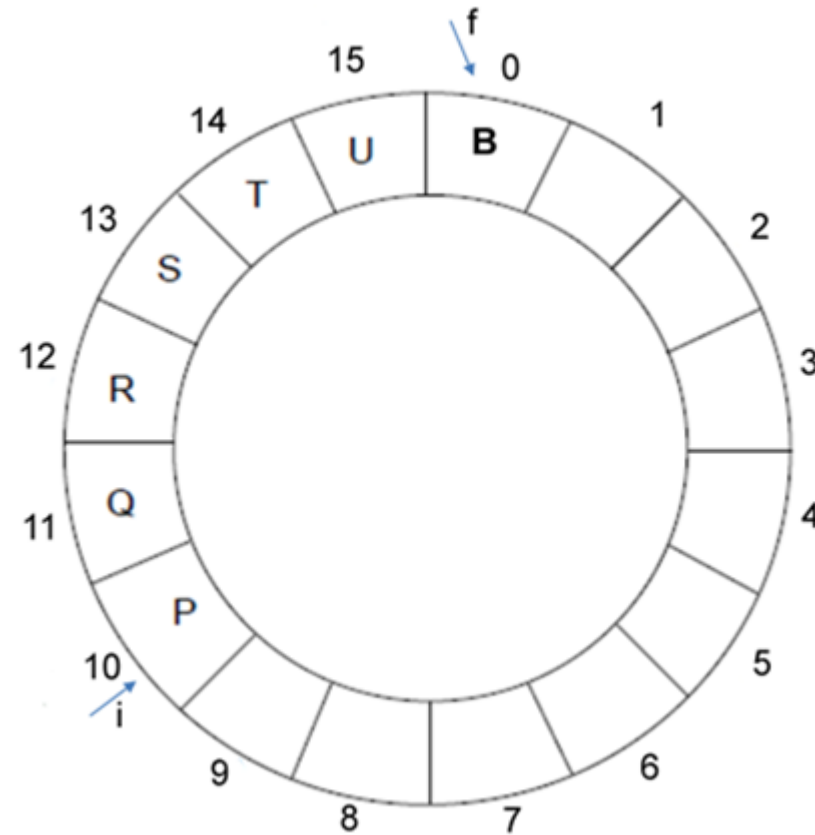


Algoritmos e Estruturas de Dados I



Fila – Uso da fila circular!

$n = 7$
 $i = 10$
 $f = 0$
 $M = 16$



Algoritmos e Estruturas de Dados I



Fila – implementação

```
#include <stdio.h>
#define M 10
typedef struct filaSeq{
    int valores[MAX];
    int i, f, n;
} filaSeq;
```

Algoritmos e Estruturas de Dados I



Fila – implementação

```
void inicializa_fila(int *i, int *f){  
    *i = -1;  
    *f = -1;  
}
```

Algoritmos e Estruturas de Dados I



Fila – implementação

```
int fila_vazia(int n){
    if (n==0)
        return 1;
    return 0;
}
int fila_cheia(int n){
    if (n==M)
        return 1;
    return 0;
}
```


Algoritmos e Estruturas de Dados I



Fila – implementação

```
void mostrar_Proximo(filaSeq *fila){
    if(!fila_vazia(fila->n)){
        printf("\nPróximo da fila: %d.\n",fila->valores[i]);
    }
    else{
        printf("\nFila vazia!\n");
    }
}
```

Algoritmos e Estruturas de Dados I



Fila – implementação

```
void mostrar_Ultimo(filaSeq *fila){
    if(!fila_vazia(fila->n)){
        printf("\nÚltimo da fila: %d.\n", fila->valores[f]);
    }
    else{
        printf("\nFila vazia!\n");
    }
}
```

Algoritmos e Estruturas de Dados I



Fila – implementação – ENFILEIRAR

- Consiste em inserir um valor no final da fila, caso a fila não esteja cheia.

Algoritmos e Estruturas de Dados I



Fila – implementação – ENFILEIRAR

```
void enfileirar(filaSeq *fila, int valor ){
    if(!fila_cheia(fila->n)){
        if(fila->f==M-1)
            fila->f=0;
        else
            (fila->f)++;
        fila->valores[fila->f]=valor;
        if((fila->i)==-1)
            (fila->i)++;
        (fila->n)++;
        printf("\nValor enfileirado!\n");
    }
    else
        printf("\nFila cheia!\n");
}
```

Algoritmos e Estruturas de Dados I



Fila – implementação – DESENFILAR

- Consiste em retirar um valor do início da fila e em seguida, ajustar o início.
- Só é possível se a pilha não estiver vazia.

Algoritmos e Estruturas de Dados I



Desenfileirar

$i = 0$

$M = 16$

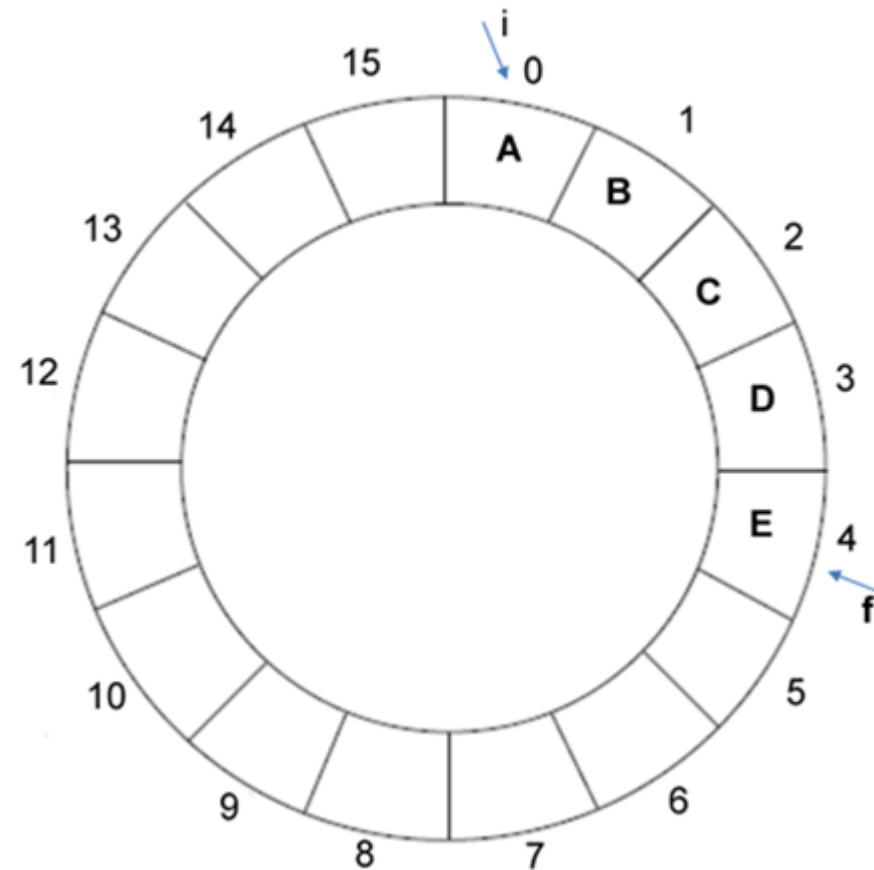
$n = 5$

$f = 4$

**Como
desenfileirar?**

$i++;$

$n--;$



Algoritmos e Estruturas de Dados I



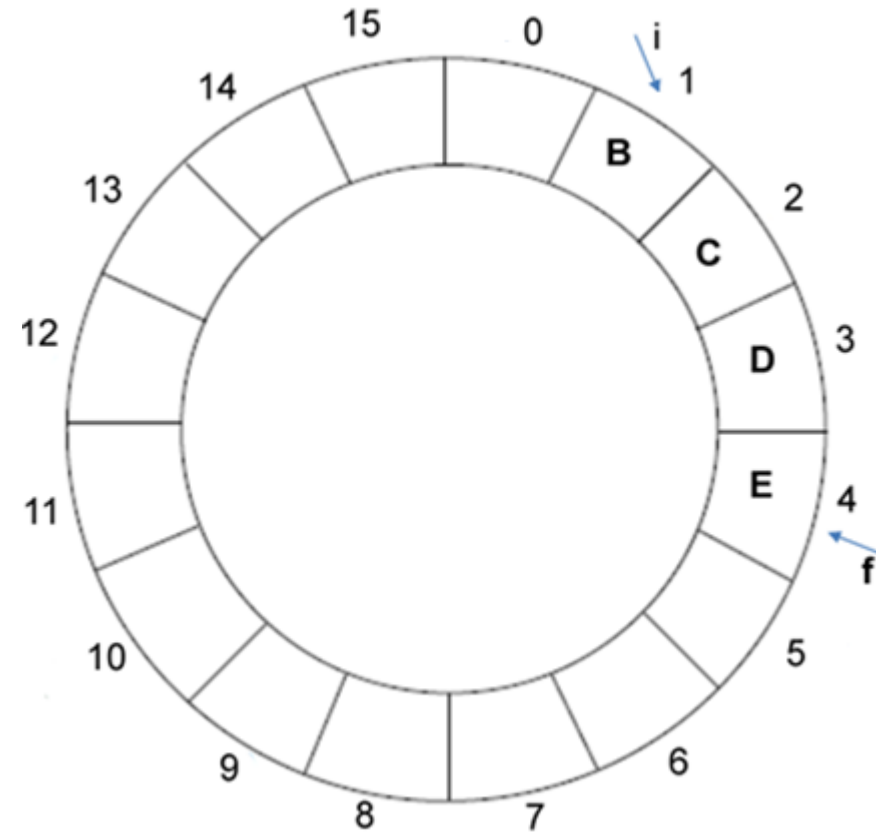
Desenfileirar

$i = 1$

$M = 16$

$n = 4$

$f = 4$



Algoritmos e Estruturas de Dados I



Desenfileirar

$i = 15$

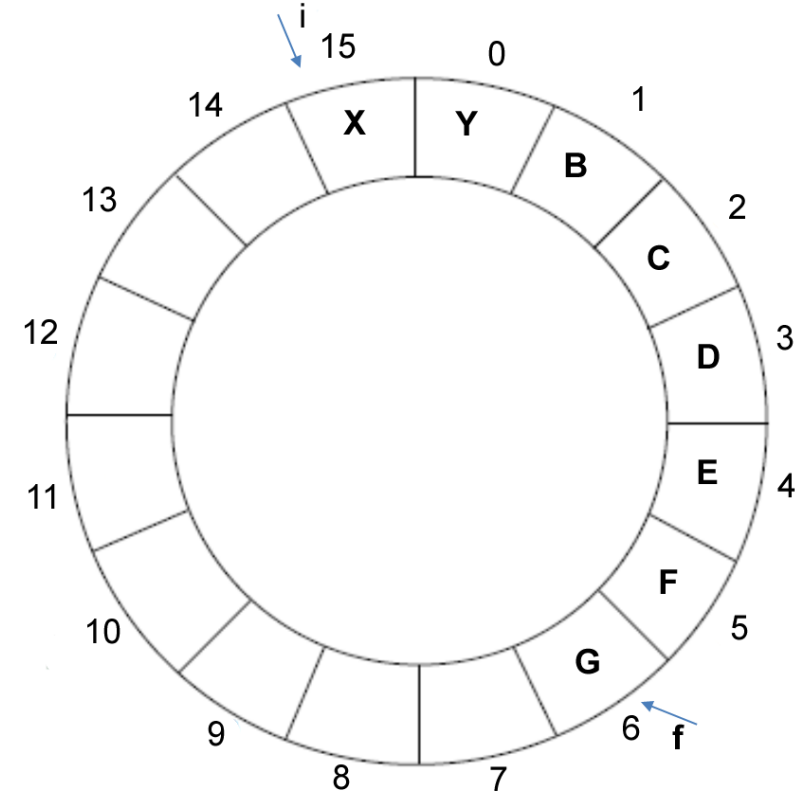
$M = 16$

$n = 8$

**Mais inserções
e remoções...**

Como desenfileirar?

$i++$; ???



Algoritmos e Estruturas de Dados I



Desenfileirar

$i = 15$

$M = 16$

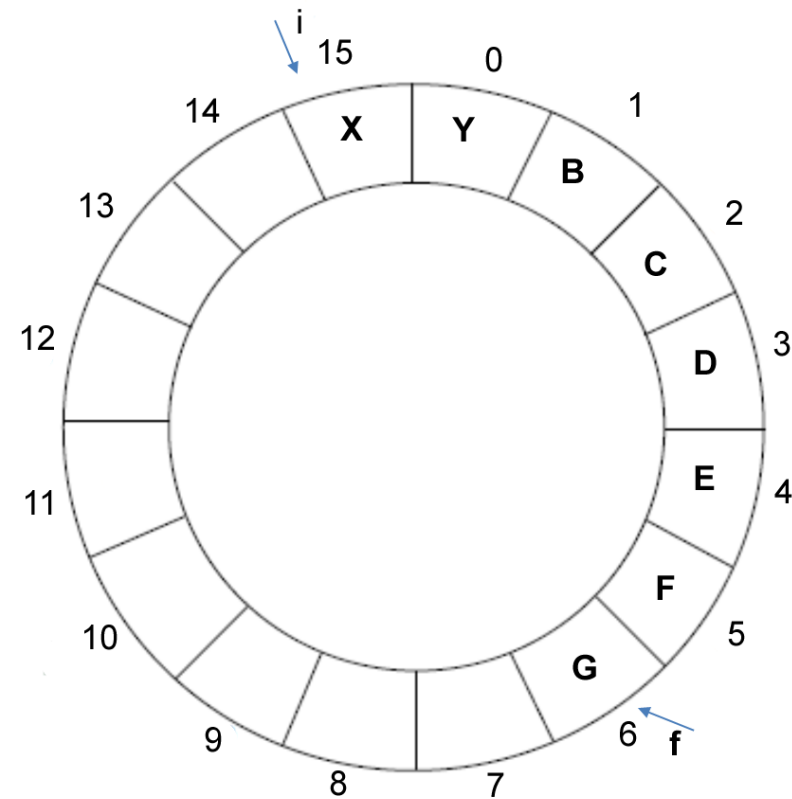
$n = 8$

$i == M-1 ?$



$i = 0;$

$n--;$



Algoritmos e Estruturas de Dados I

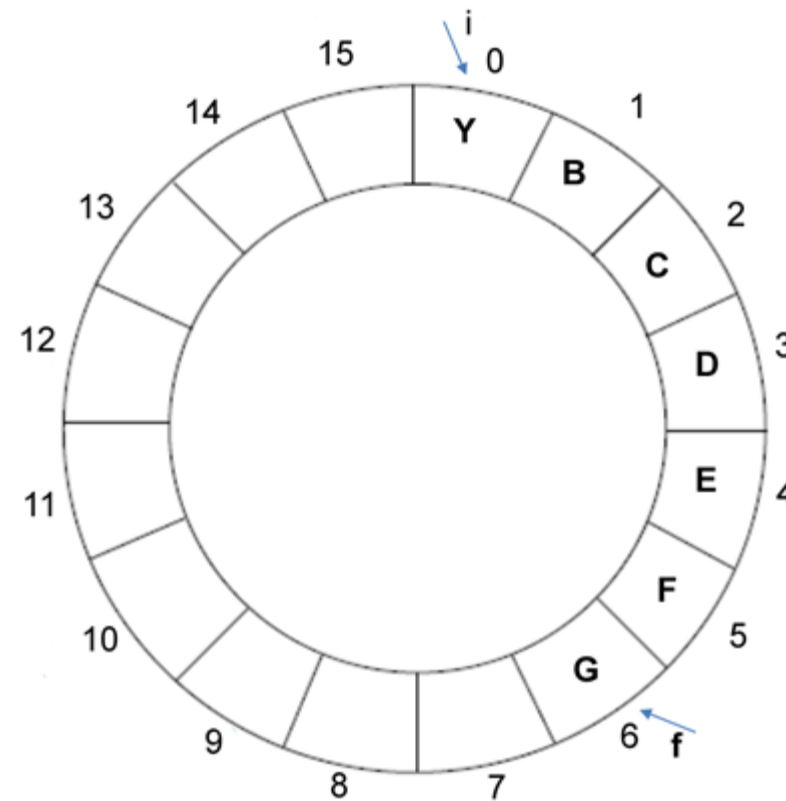


Desenfileirar

$i = 0$

$M = 16$

$n = 7$



Algoritmos e Estruturas de Dados I



Desenfileirar

$i = 0$

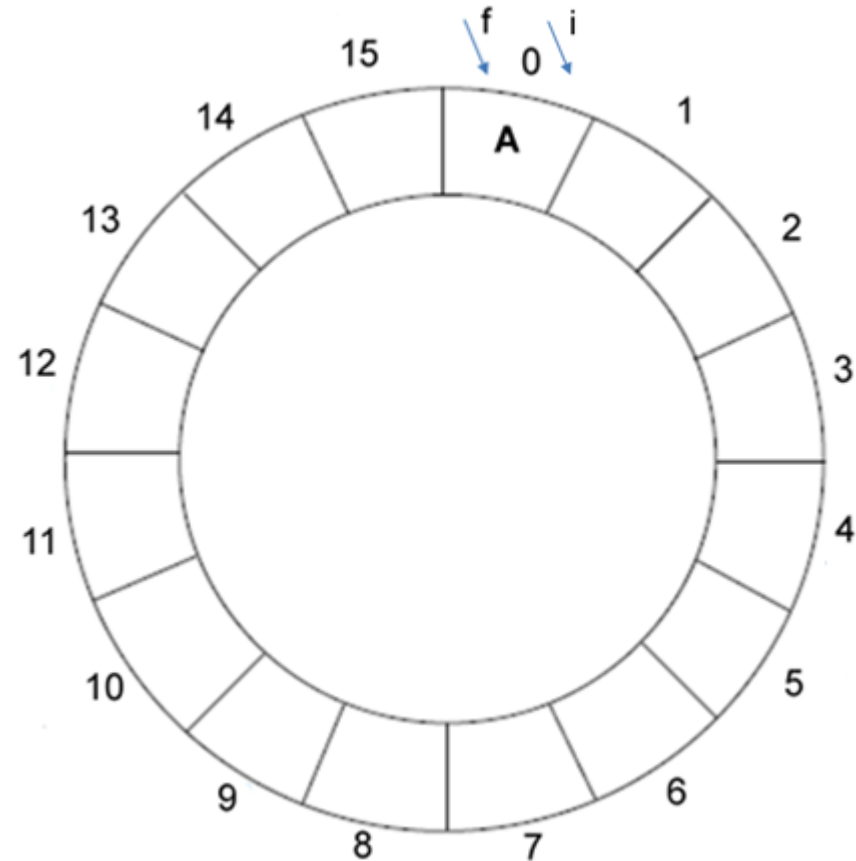
$f = 0$

$M = 16$

$n = 1$

**Como
desenfileirar?**

**$i = -1;$
 $f = -1;$
 $n--;$**



Algoritmos e Estruturas de Dados I



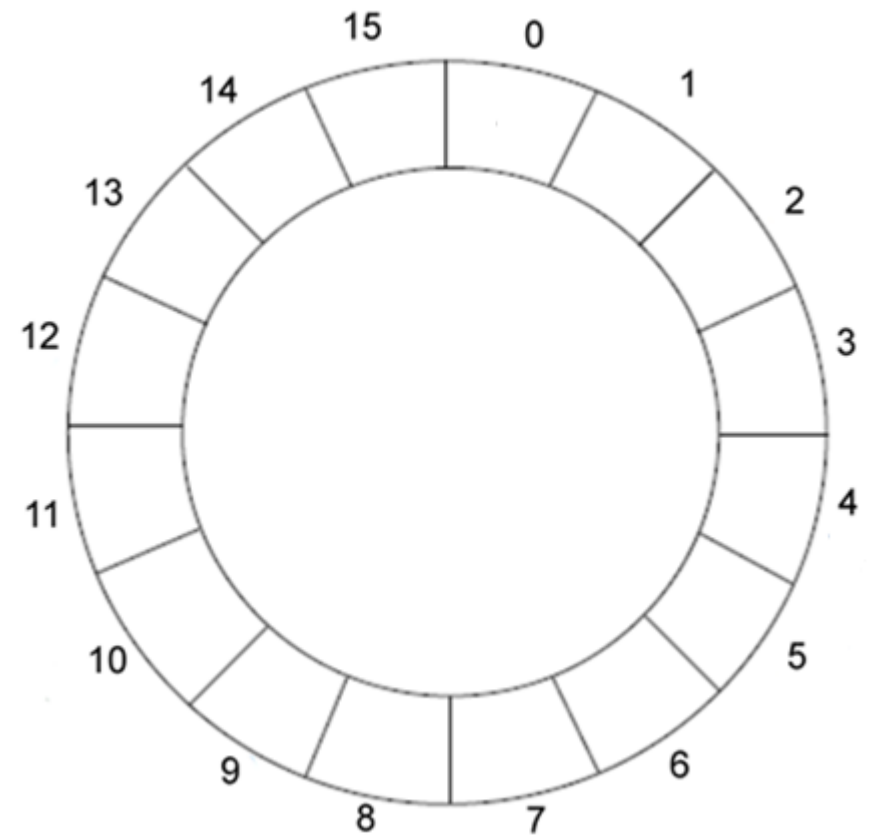
Desenfileirar

$i = -1$

$f = -1$

$M = 16$

$n = 0$



Algoritmos e Estruturas de Dados I



Fila – implementação – DESEMPILHAR

```
void desenfileirar(filaSeq *fila){
    if(!fila_vazia(fila->n)){
        printf("\nValor %d desenfileirado!\n", fila->valores[fila->i]);
        if(fila->n==1){//um só elemento na fila
            * fila->i =-1;
            * fila->f =-1;
        }
        else{//Mais de um elemento armazenado
            if(fila->i==M-1) //Última posição
                fila->i=0;
            else //Qualquer posição
                (fila->i)++;
        }
        (fila->n)--;
    }
    else
        printf("\nFila vazia!\n");
}
```

Algoritmos e Estruturas de Dados I



Fila – implementação

```
int main(){
    filaSeq fila[MAX];
    int valor, op;
    inicializa_fila(&fila->i,&fila->f);
    do{
        printf("\n1- Enfileirar");
        printf("\n2- Desenfileirar");
        printf("\n3- Mostrar próximo");
        printf("\n4- Mostrar último");
        printf("\n5- Sair");
        printf("\nInforme sua opção: ");
        scanf("%d",&op);
```

Algoritmos e Estruturas de Dados I



Fila – implementação

```
switch (op){
    case 1:{
        printf("\nInforme o valor a enfileirar: ");
        scanf("%d",&valor);
        enfileira(fila,valor);
        break;
    }
    case 2:{
        desenfileira(fila);
        break;
    }
}
```

Algoritmos e Estruturas de Dados I



Fila – implementação

```
case 3:{
    mostra_proximo(fila);
    break;
}
case 4:{
    mostra_ultimo(fila);
    break;
}
```


Algoritmos e Estruturas de Dados I



Fila – implementação

```
        case 5:{
            printf("\nFinalizando...\n");
            break;
        }
        default:{
            printf("\nOpção inválida!\n");
            break;
        }
    }
}while(op!=5);
}
```

Algoritmos e Estruturas de Dados I



Exercício

Faça um programa que implemente uma fila sequencial. O usuário deverá informar números inteiros, que serão inseridos na fila. Ao final da leitura de dados, o programa deverá inverter a ordem dos elementos na fila, isto é, os últimos serão os primeiros!

Você pode utilizar estruturas auxiliares, dentre as que já foram vistas. Decida que estrutura poderá auxiliar nessa tarefa!



FIM