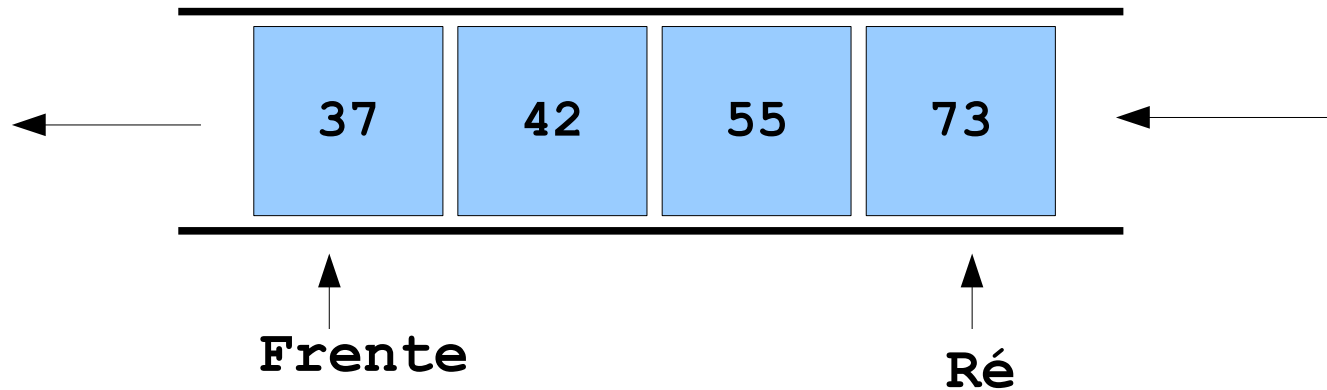


Estruturas de dados

Fila (Queue)

Fila (Queue)

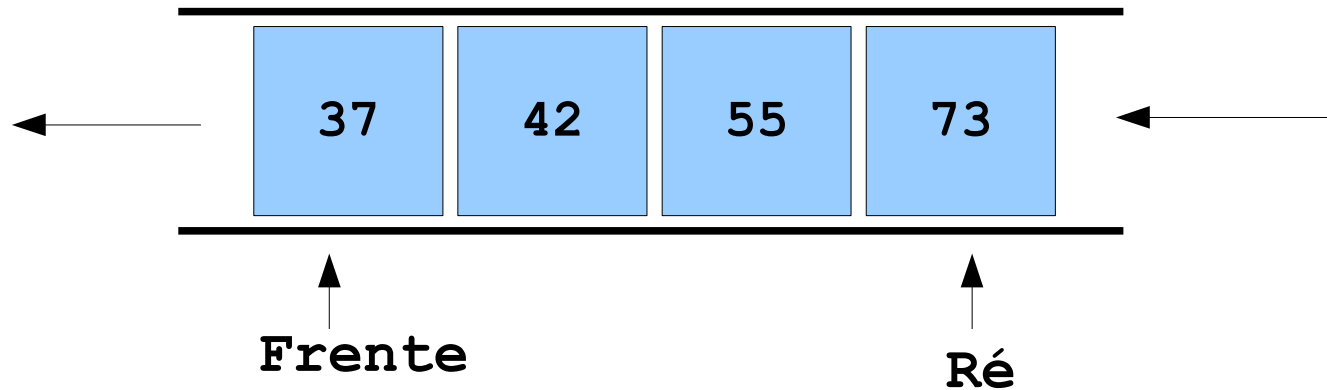
É uma lista onde a inserção é realizada em uma extremidade e a retirada na outra.



Critério "FIFO" (First In First Out)
(primeiro que entra é o primeiro que sai)

Fila (Queue)

É uma lista onde a inserção é realizada em uma extremidade e a retirada na outra.



Critério "FIFO" (First In First Out)
(primeiro que entra é o primeiro que sai)

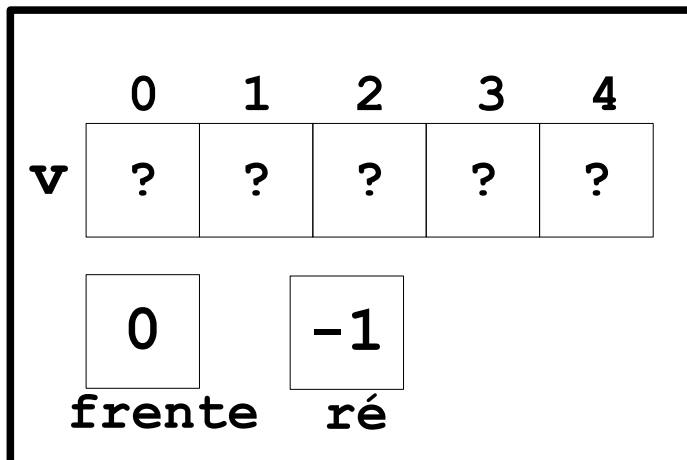
Fila (Queue)

Operações:

```
void criaFila (FilaCF *f)
int  insere  (FilaCF *f,int dado)
int  retira (FilaCF *f,int *dado)
int  estaCheia(FilaCF f)
int  estaVazia(FilaCF f)
int  consulta(FilaCF f,int *dado)
```

Representação por contiguidade física:

fila vazia



v : Vetor que armazena os nodos da pilha.

frente : índice do nodo que está no início da fila.

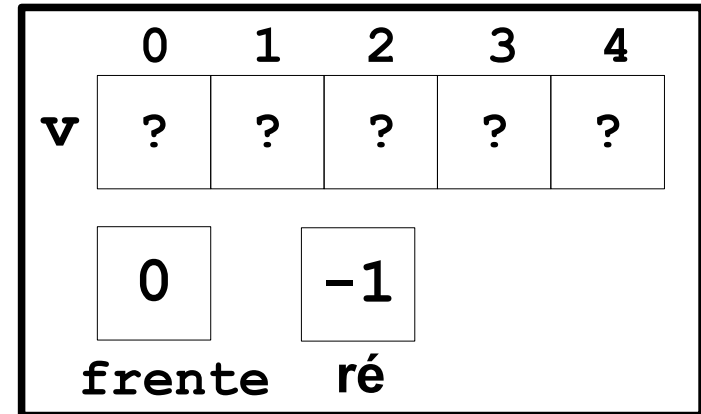
re: índice do nodo que está no fim da fila.

OBS: Valores iniciais de frente e ré tornam a inclusão do primeiro nodo igual as demais.

Fila (Queue)

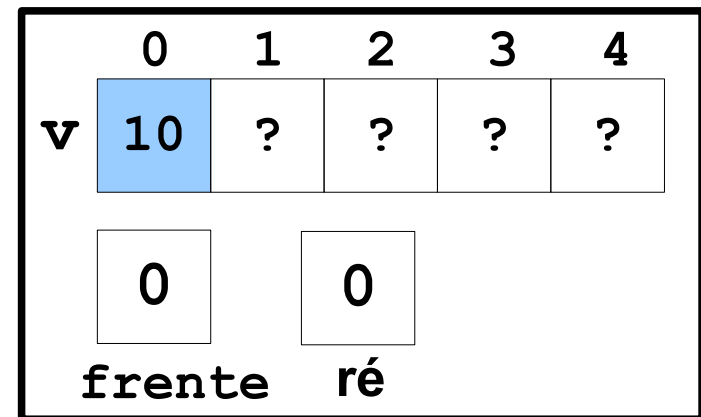
Funcionamento:

```
int main() {  
    FilaCF f;  
    criaFila(&f);
```



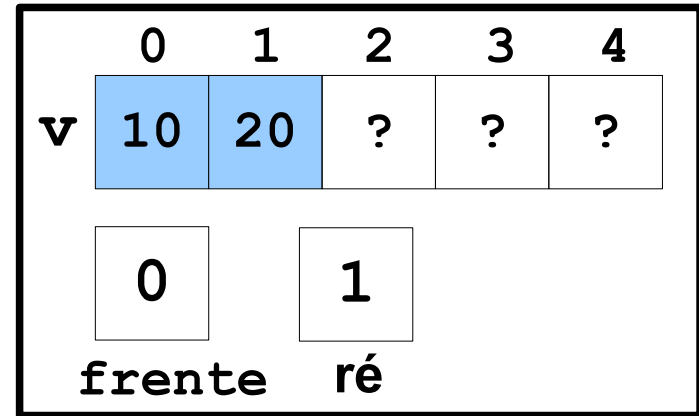
```
// Na inclusão a ré é incrementada  
// re++;
```

```
insere(&f, 10);
```



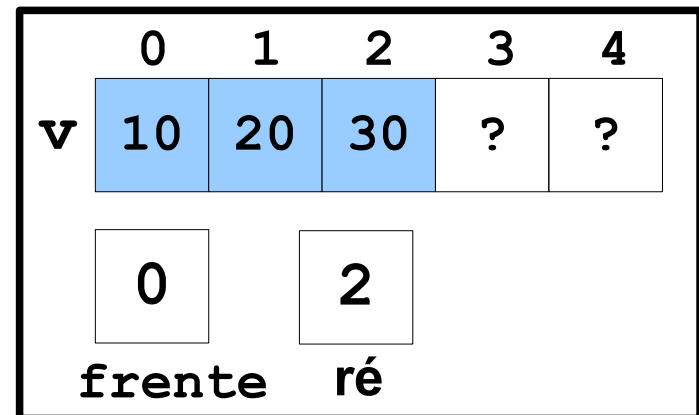
Fila (Queue)

`insere (&f, 20) ;`



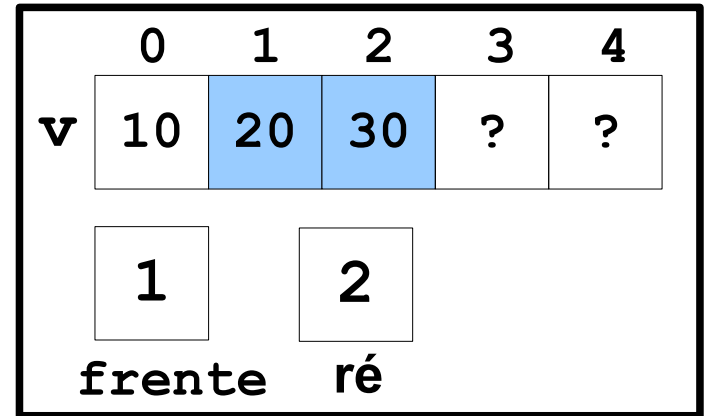
`insere (&f, 30) ;`

...



Fila (Queue)

```
retira(&f, &x) ;
```



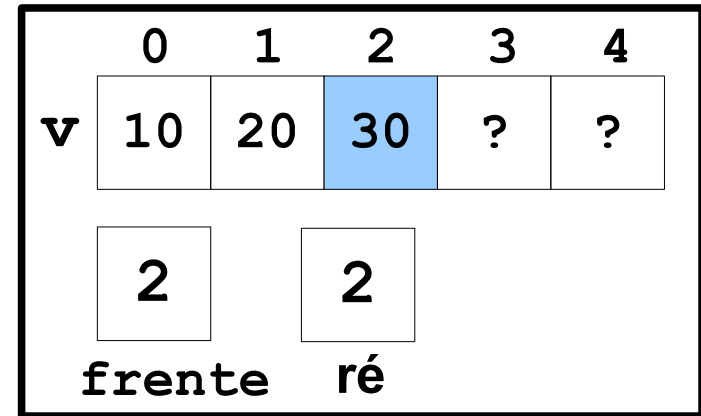
```
// Na inclusão a frente é incrementada  
// frente++;
```

Como verificar se a fila está vazia?

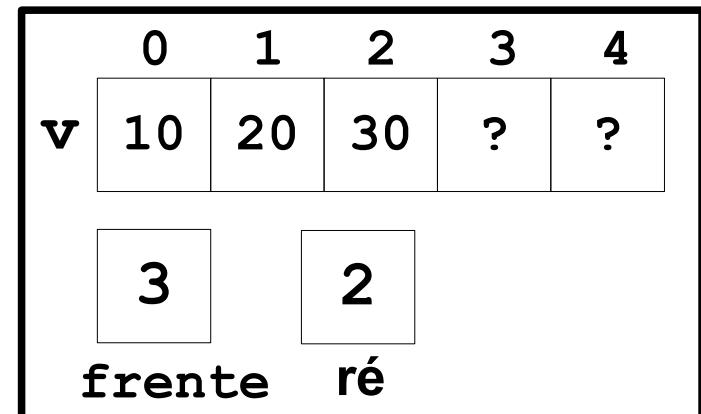
Como verificar se a fila está cheia?

Fila (Queue)

...
`retira(&f, &x) ;`



`retira(&f, &x) ;`

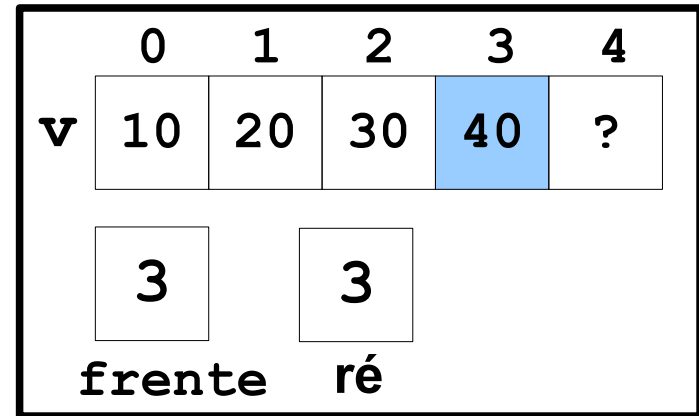


Como verificar se a fila está vazia?

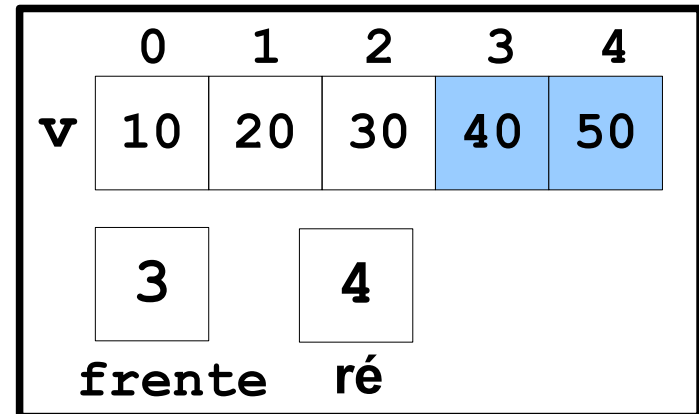
Ré < Frente

Fila (Queue)

...
`insere(&f, 40);`



`insere(&f, 50);`



Como verificar se a fila está cheia?

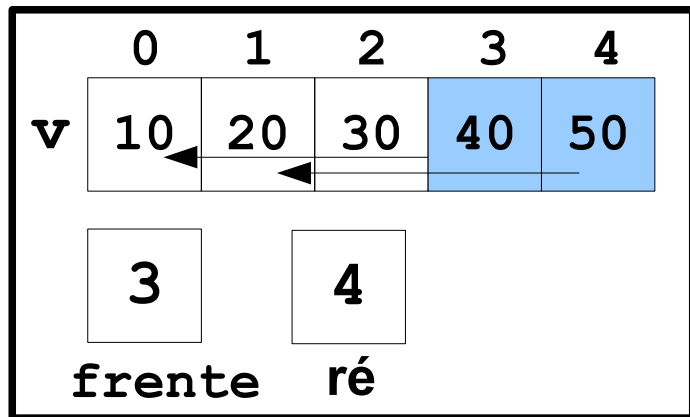
$$\text{Ré} = \text{MAX_NODOS} - 1$$

Fila (Queue)

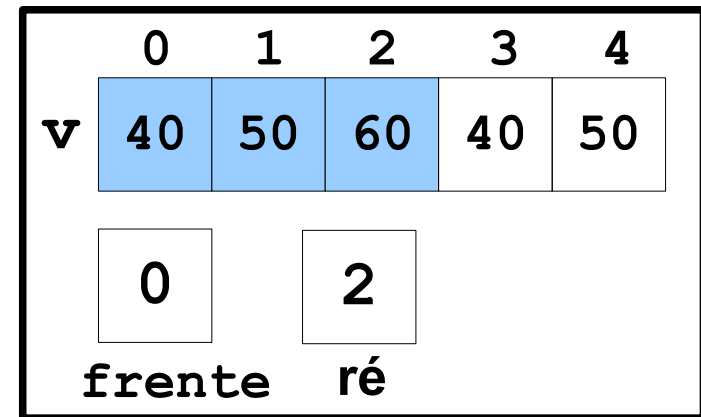
Dessa forma não há reaproveitamento do espaço já ocupado.

Como reaproveitar o espaço?

- 1) Deslocando os nodos para o início do vetor na situação de fila cheia.



Após a inclusão do nodo 60



Fila - Implementação

```
#define SUCESSO      0           /* Códigos de erro */
#define FILA_VAZIA  1
#define FILA_CHEIA  2

#define MAX_NODOS  5

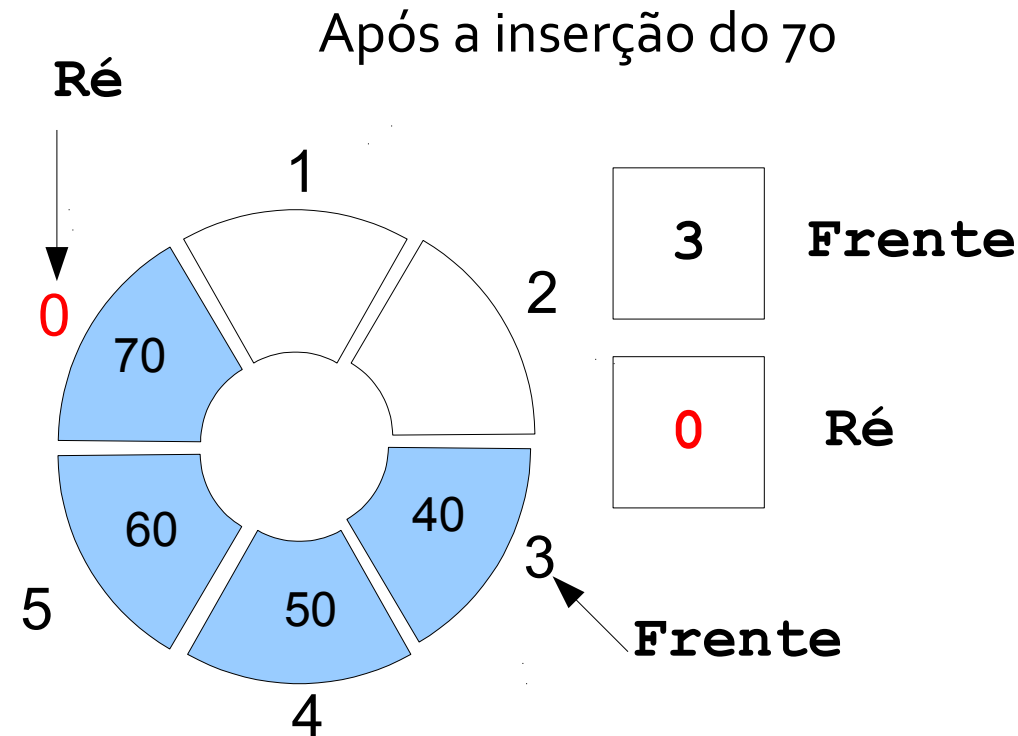
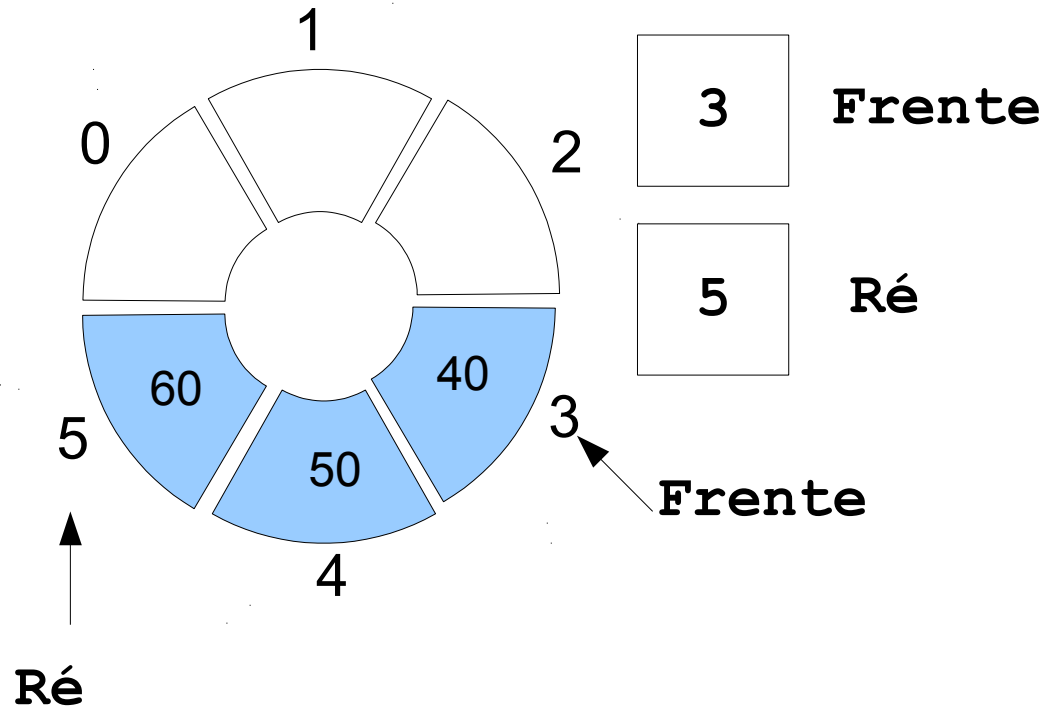
typedef struct {
    int v[MAX_NODOS];
    int frente;      /* índice da frente */
    int re;          /* índice da re */
} FilaCF;
```

Operações:

```
void criaFila (FilaCF *f)
int  insere (FilaCF *f,int dado)
int  retira(FilaCF *f,int *dado)
int  estaCheia(FilaCF f)
int  estaVazia(FilaCF f)
int  consulta(FilaCF f,int *dado)
```

Fila Circular

2) Criando uma fila circular, onde o último elemento é considerado adjacente ao primeiro.



Fila Circular

Incremento da ré para a inserção:

```
re++;                                OU  re = (re+1) % MAX_NODOS;  
if (re==MAX_NODOS)  
    re=0;
```

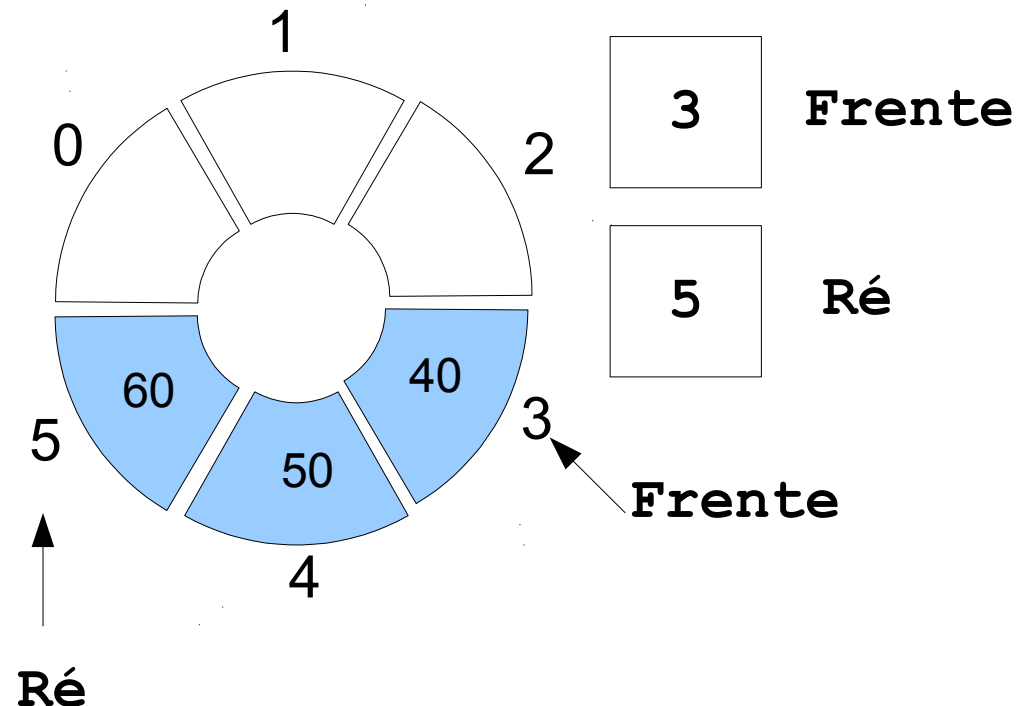
Ex: Para $\text{MAX_NODOS} = 6$

Antes:

Re = 5

Depois da inserção

Re = 0

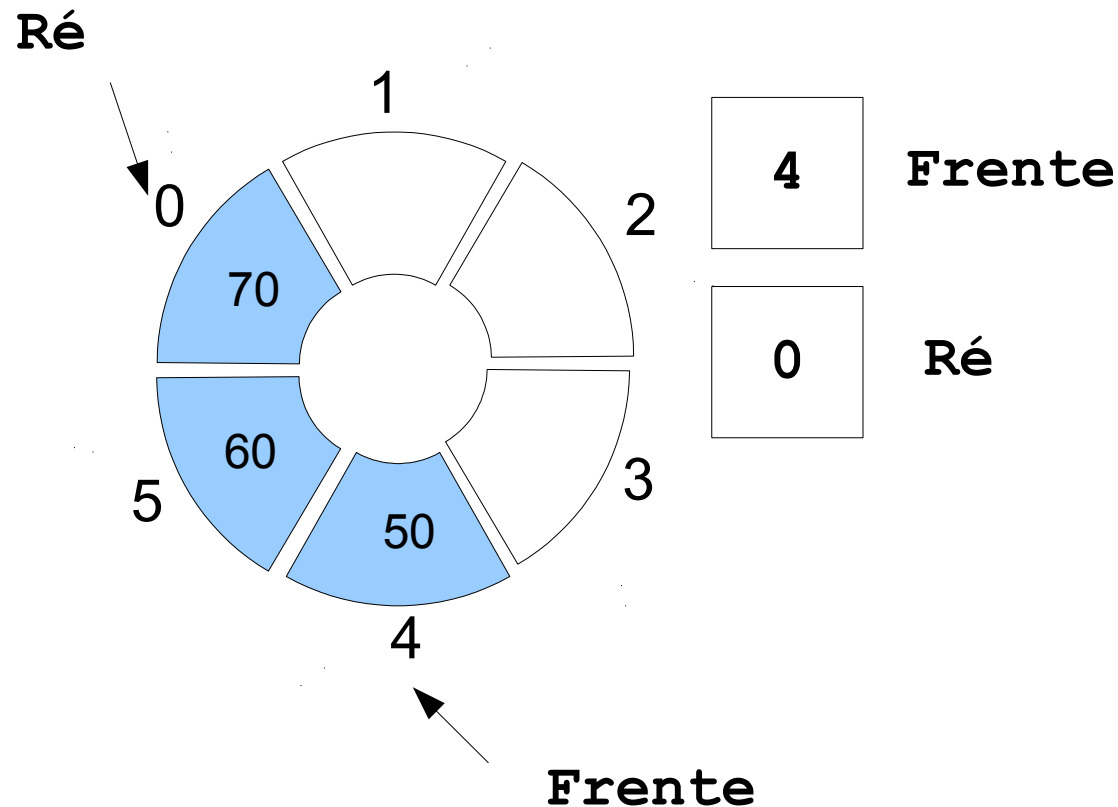


Fila Circular

Como verificar se a fila está vazia?

~~re < frente~~

Ré < Frente e a fila
não está vazia

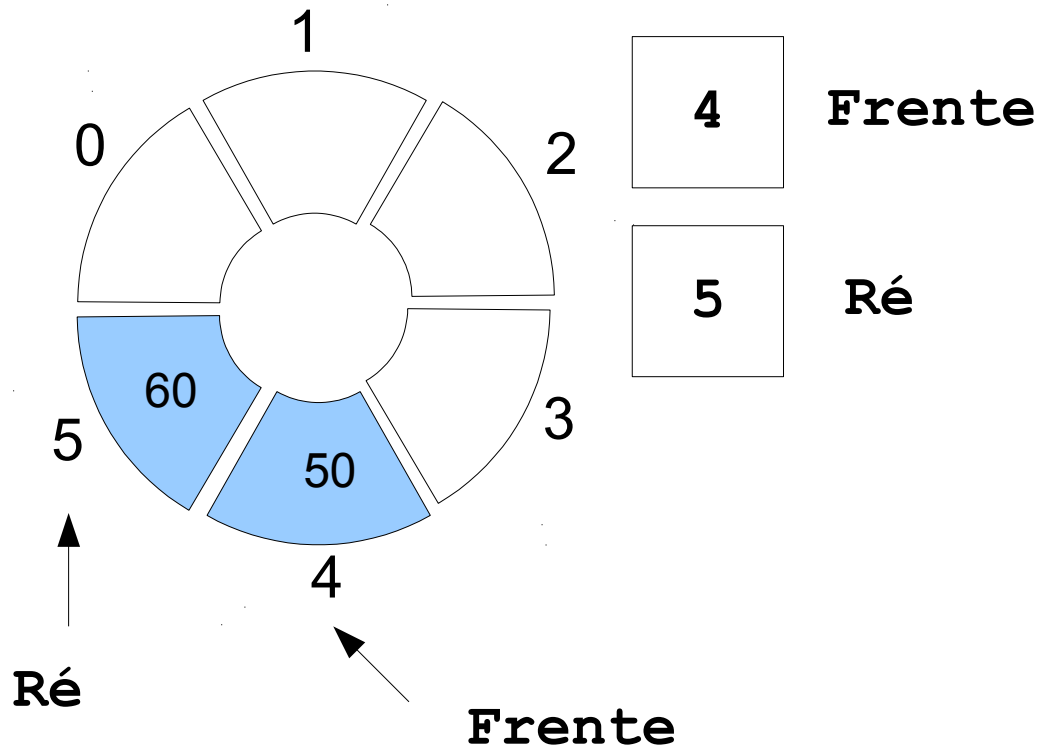


Fila Circular

Como verificar se a fila está cheia?

~~$re = MAX_NODOS - 1$~~

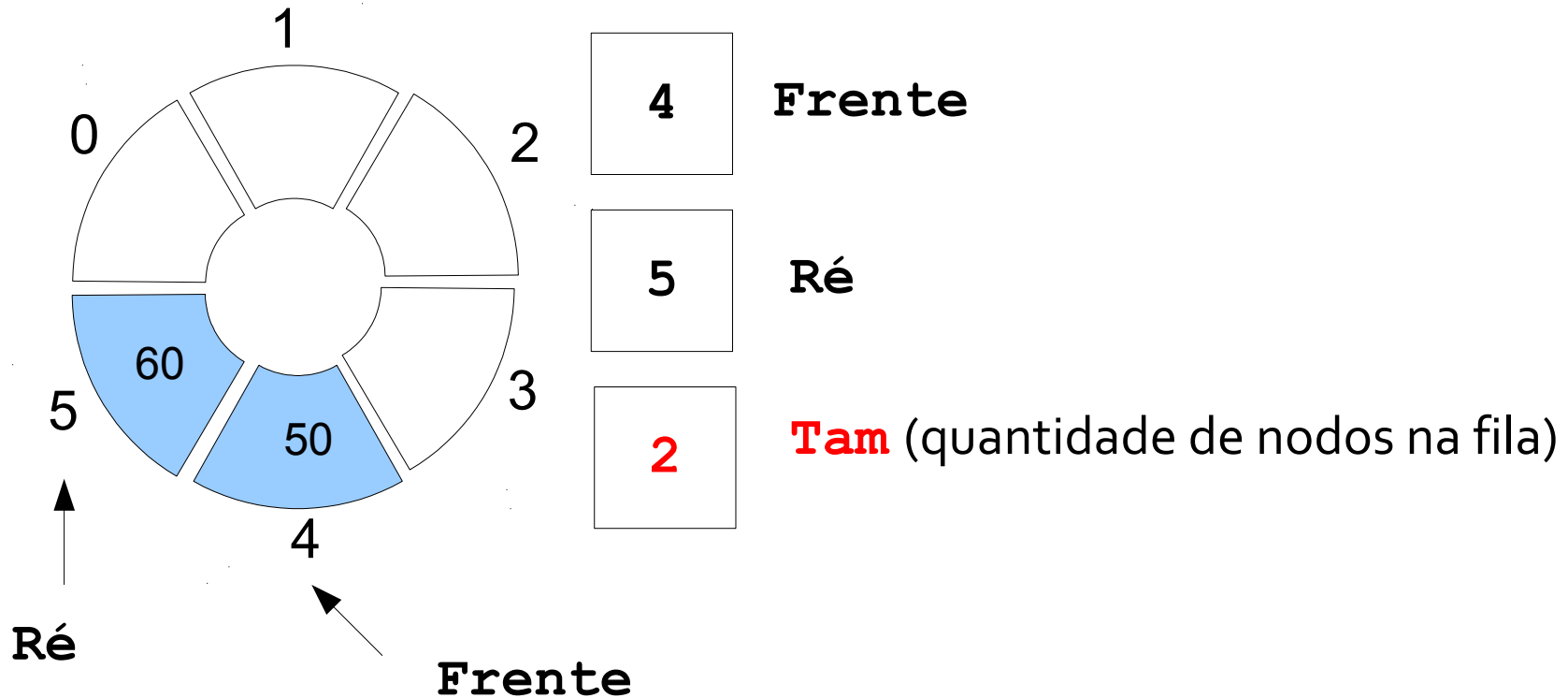
$MAX_NODOS = 6$



$Ré = MAX_NODOS - 1$
e a fila **não** está cheia

Fila Circular

Solução: Incluir um controle de tamanho:



Como verificar se a fila está vazia?

tam=0

Como verificar se a fila está cheia?

tam=MAX_NODOS

Fila - Implementação

```
#define SUCESSO      0          /* Códigos de erro */
#define FILA_VAZIA  1
#define FILA_CHEIA  2

#define MAX_NODOS  5

typedef struct {
    int v[MAX_NODOS];
    int frente;      /* índice da frente */
    int re;          /* índice da re */
    int tam;         /* Quantidade de nodos */
} FilaCF;
```

Operações:

```
void criaFila (FilaCF *f)
int  insere (FilaCF *f,int dado)
int  retira(FilaCF *f,int *dado)
int  estaCheia(FilaCF f)
int  estaVazia(FilaCF f)
int  consulta(FilaCF f,int *dado)
```

Fila – Uma aplicação

Preencher uma região delimitada por uma borda.

Ex: Considere a área armazenada em uma matriz

0	0	0	1	1	0	0		0	0	0	1	1	0	0
0	0	1	0	0	1	0		0	0	1	2	2	1	0
0	1	0	0	0	0	1		0	1	2	2	2	2	1
1	0	0	0	0	1	0		1	2	2	2	2	1	0
0	1	1	1	1	0	0		0	1	1	1	1	0	0

Criar uma fila

Inserir na fila o ponto inicial do preenchimento

Enquanto a fila não estiver vazia faça:

 Retirar um ponto P da fila

 Se P não foi pintado então

 Inserir na fila os pontos vizinhos a P que possuem cor
 diferente da borda e da cor do preenchimento

 Alterar a cor do ponto P para a cor do preenchimento

 Fim_se

Fim_Enquanto

OBS: Pontos vizinhos a P: direita, abaixo, esquerda e acima

Fila – Uma aplicação

	0	1	2	3	4	5	6
0	0	0	0	1	1	0	0
1	0	0	1	0	0	1	0
2	0	1	0	0	0	0	1
3	1	0	0	0	0	1	0
4	0	1	1	1	1	0	0

	0	1	2	3	4	5	6
0	0	0	0	1	1	0	0
1	0	0	1	0	0	1	0
2	0	1	2	0	0	0	1
3	1	0	0	0	0	1	0
4	0	1	1	1	1	0	0

P:2,2

2,2				
-----	--	--	--	--

		2,3	3,2		
--	--	-----	-----	--	--

Fila – Uma aplicação

	0	1	2	3	4	5	6
0	0	0	0	1	1	0	0
1	0	0	1	0	0	1	0
2	0	1	2	2	0	0	1
3	1	0	0	0	0	1	0
4	0	1	1	1	1	0	0

P:2,3

			3,2	2,4	3,3	1,3	
--	--	--	-----	-----	-----	-----	--

	0	1	2	3	4	5	6
0	0	0	0	1	1	0	0
1	0	0	1	0	0	1	0
2	0	1	2	2	0	0	1
3	1	0	2	0	0	1	0
4	0	1	1	1	1	0	0

P:3,2

		2,4	3,3	1,3	3,3	3,1	
--	--	-----	-----	-----	-----	-----	--

Fila – Uma aplicação

	0	1	2	3	4	5	6
0	0	0	0	1	1	0	0
1	0	0	1	0	0	1	0
2	0	1	2	2	2	0	1
3	1	0	2	0	0	1	0
4	0	1	1	1	1	0	0

P:2,4

3,3	1,3	3,3	3,1	2,5	3,4	1,4
-----	-----	-----	-----	------------	------------	------------

	0	1	2	3	4	5	6
0	0	0	0	1	1	0	0
1	0	0	1	0	0	1	0
2	0	1	2	2	2	0	1
3	1	0	2	2	0	1	0
4	0	1	1	1	1	0	0

P:3,3

1,3	3,3	3,1	2,5	3,4	1,4	3,4
-----	-----	-----	-----	-----	-----	------------

Fila – Uma aplicação

	0	1	2	3	4	5	6
0	0	0	0	1	1	0	0
1	0	0	1	2	0	1	0
2	0	1	2	2	2	0	1
3	1	0	2	2	0	1	0
4	0	1	1	1	1	0	0

	0	1	2	3	4	5	6
0	0	0	0	1	1	0	0
1	0	0	1	2	0	1	0
2	0	1	2	2	2	0	1
3	1	2	2	2	0	1	0
4	0	1	1	1	1	0	0

P:1,3

3,3	3,1	2,5	3,4	1,4	3,4	1,4
-----	-----	-----	-----	-----	-----	-----

P:3,3 P:3,1

2,5	3,4	1,4	3,4	1,4			
-----	-----	-----	-----	-----	--	--	--

Fila – Uma aplicação

	0	1	2	3	4	5	6
0	0	0	0	1	1	0	0
1	0	0	1	2	0	1	0
2	0	1	2	2	2	2	1
3	1	2	2	2	0	1	0
4	0	1	1	1	1	0	0

P:2,5

		3,4	1,4	3,4	1,4			
--	--	-----	-----	-----	-----	--	--	--

	0	1	2	3	4	5	6
0	0	0	0	1	1	0	0
1	0	0	1	2	0	1	0
2	0	1	2	2	2	0	1
3	1	2	2	2	2	1	0
4	0	1	1	1	1	0	0

P:3,4

			1,4	3,4	1,4			
--	--	--	-----	-----	-----	--	--	--

Fila – Uma aplicação

	0	1	2	3	4	5	6
0	0	0	0	1	1	0	0
1	0	0	1	2	2	1	0
2	0	1	2	2	2	2	1
3	1	2	2	2	2	1	0
4	0	1	1	1	1	0	0

P:1,4

		3,4	1,4					
--	--	-----	-----	--	--	--	--	--

	0	1	2	3	4	5	6
0	0	0	0	1	1	0	0
1	0	0	1	2	2	1	0
2	0	1	2	2	2	2	1
3	1	2	2	2	2	1	0
4	0	1	1	1	1	0	0

P:3,4 P:1,4

--	--	--	--	--	--	--	--	--