

Aula 11: Filas

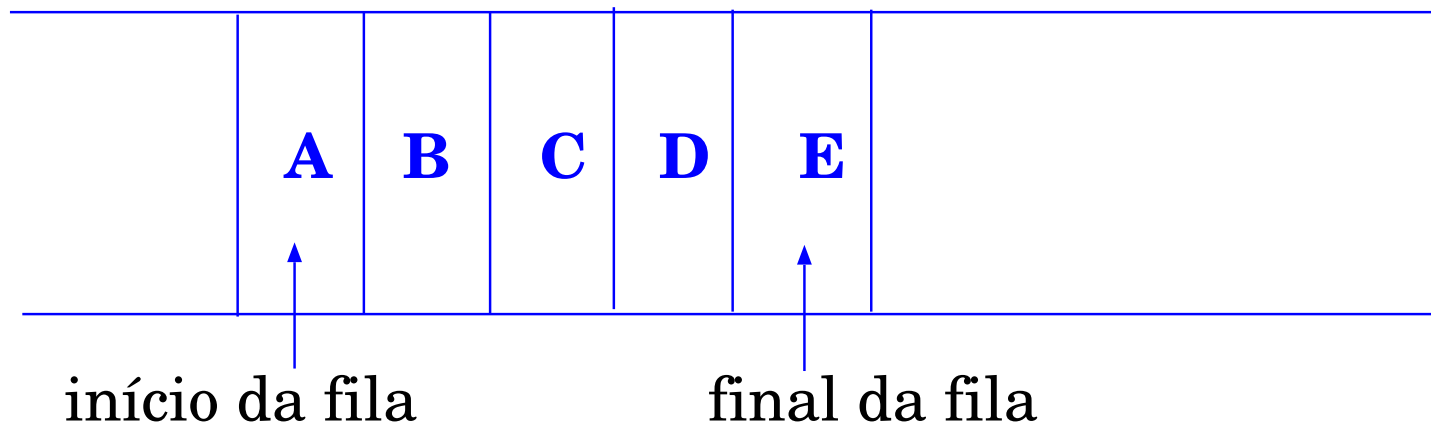
⇒ Conceito

⇒ Algoritmos de inserção e remoção

⇒ Deques

Filas

➡ Filas: lista em que todas as inserções ocorrem em uma das extremidades e todas as remoções ocorrem na outra.



➡ Início da fila: extremidade onde ocorrem as remoções

➡ Final da fila: extremidade onde ocorrem as inserções

Filas

- ➡ Forma de armazenamento:
 - ▬ Em alocação sequencial,
 - ▬ usar vetores;
 - ▬ um ponteiro indica o início da fila (frente);
 - ▬ um ponteiro indica o final da fila (retaguarda).
- ➡ Operações básicas:
 - ▬ inserção
 - ▬ remoção
- ➡ Situações extremas:
 - ▬ fila cheia
 - ▬ fila vazia

Filas

⇒ Operações inválidas:

- ▬ inserção em fila cheia ([overflow](#))
- ▬ remoção em fila vazia ([undeflow](#))

⇒ Forma de operação:

- ▬ primeiro a entrar, primeiro a sair (FIFO);
- ▬ a ordem das inserções coincide com a ordem das remoções.

Operação de uma Fila

➡ Exemplo de operação de uma fila:

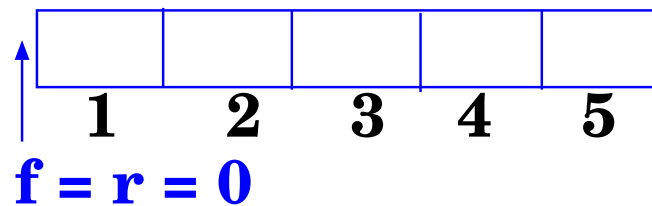
f = ponteiro para a frente;

r = ponteiro para a retaguarda;

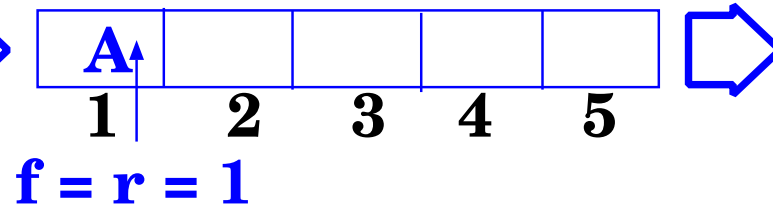
fila vazia: $f = r = 0$.

Operação de uma Fila (cont.)

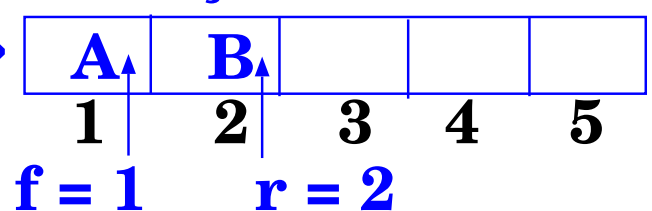
situação 1: fila vazia



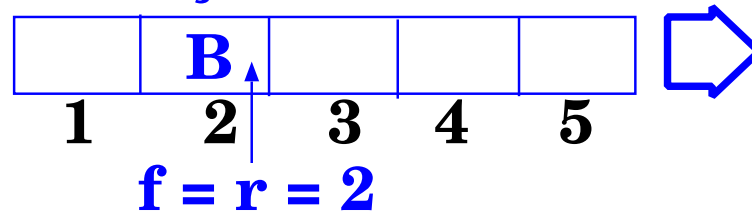
situação 2: inserir A



situação 3: inserir B

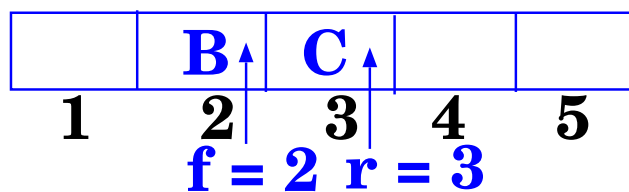


situação 4: remover

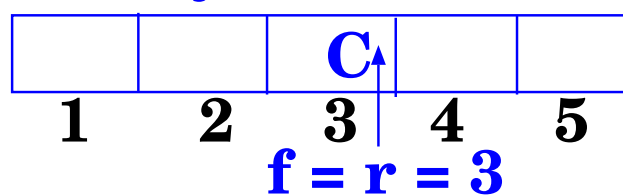


Operação de uma Fila (cont.)

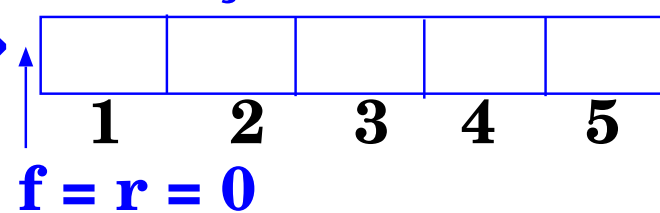
situação 5: inserir C



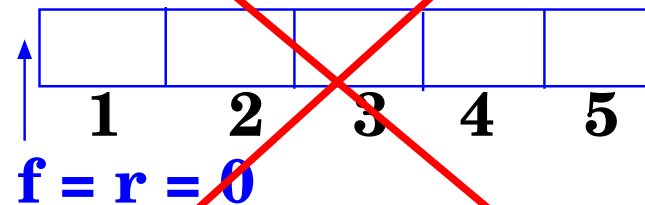
situação 6: remover



situação 7: remover



situação 8: remover



underflow

Animação



Remover quando $f=r=0 \rightarrow$ underflow

Animar

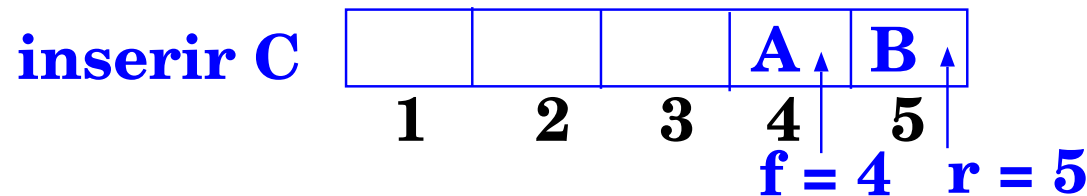
Voltar

Inserção e Remoção em Filas

➡ Algoritmos de inserção e remoção em filas:

- ➡ mais elaborados do que no caso de pilhas;
- ➡ uso de dois ponteiros:
 - ➡ f = frente;
 - ➡ r = retaguarda.
- ➡ fila vazia: $f = r = 0$
- ➡ em uma inserção, r é incrementado;
- ➡ em uma remoção, f é incrementado, exceto quando a fila se tornar vazia.
- ➡ Consequência: a fila tende a se mover para a direita.
- ➡ A situação pode provocar a ocorrência de falsos overflows.

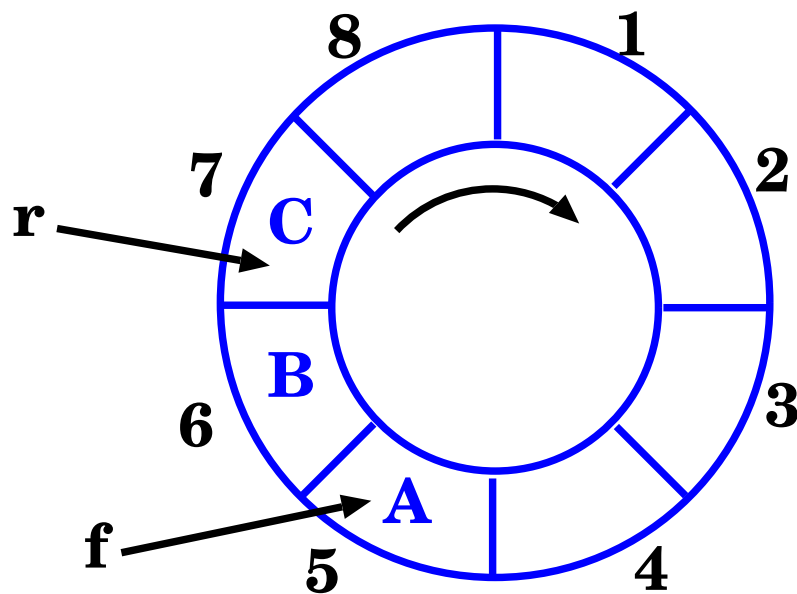
➡ Exemplo de falso overflow:



Filas

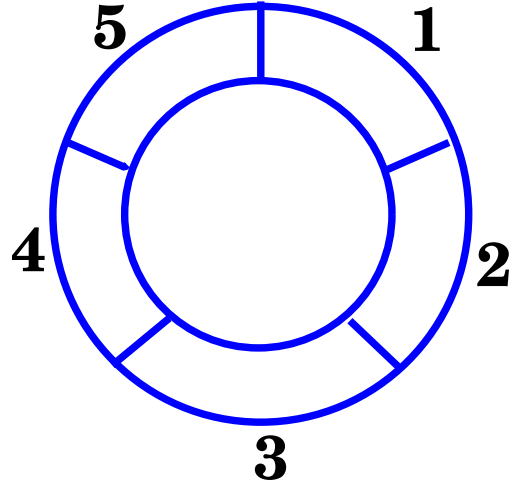
➡ Alternativa:

- ➡ Dispor os dados na fila de forma circular
- ➡ Lista circular
- ➡ A fila está armazenada em um vetor F , de M posições
- ➡ As posições $F[1]$ e $F[M]$ são tratadas como contíguas



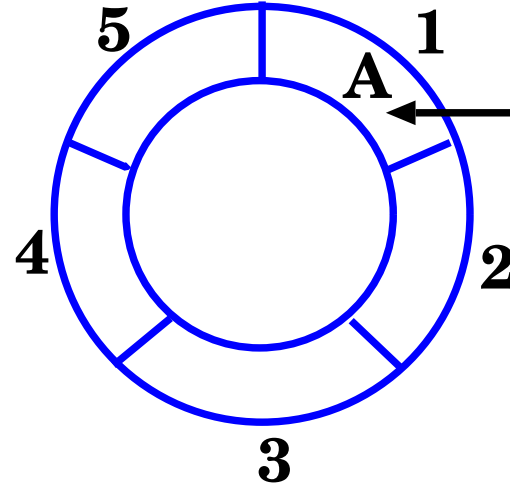
Exemplo de Operação

situação 1: inicial (fila vazia)



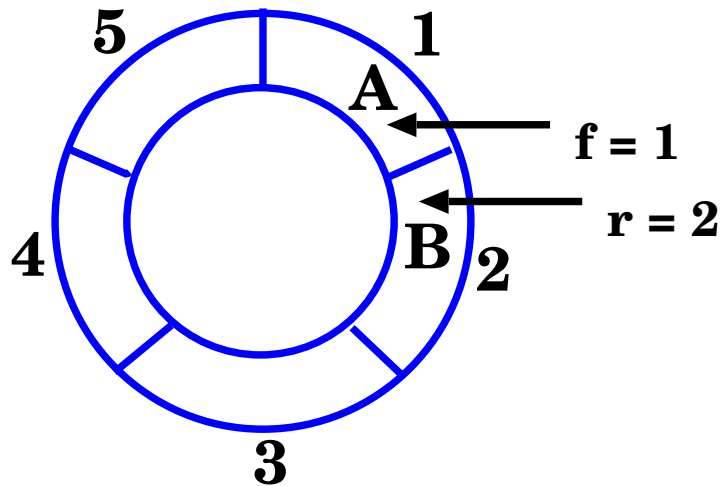
$f = r = 0$

situação 2: inserir A

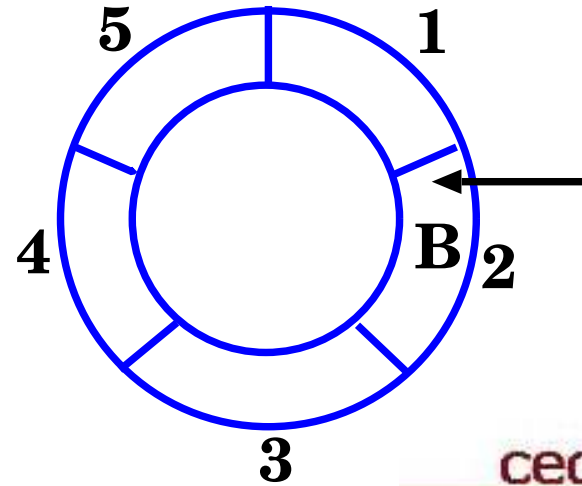


$f = r = 1$

situação 3: inserir B



situação 4: remover



$f = r = 2$

Algoritmo de inserção

- ▢ A fila se encontra armazenada em F ;
- ▢ Os ponteiros f , r indicam a frente e a retaguarda, respectivamente;
- ▢ O vetor F possui M posições, sendo $F[1]$ e $F[M]$ tratadas como contíguas;
- ▢ A variável $prov$ armazena provisoriamente a posição calculada.

➡ Algoritmo: inserção na fila F

```

prov := (r mod M)+1
se prov ≠ f então
    r := prov
    F[ r ] := novo_valor
    se f = 0 então f := 1
senão overflow
  
```

▢ fila vazia: $f = r = 0$

▢ overflow: $f = (r \bmod M)+1$, após a inserção

▢ a informação a ser inserida é $novo_valor$

Algoritmo de remoção

- ▣ A fila se encontra armazenada no vetor F ;
- ▣ Os ponteiros f , r indicam a frente e a retaguarda, respectivamente;
- ▣ O vetor F possui M posições, sendo $F[1]$ e $F[M]$ tratadas como contíguas;

➡ Algoritmo: remoção da fila F

se $f \neq 0$ então

$\text{valor_recuperado} := F[f]$

se $f = r$ então

$f := r := 0$

senão

$f := (f \bmod M) + 1$

senão underflow

▣ fila vazia: $f = 0$

▣ underflow: remoção com $f = 0$

▣ a informação a ser removida é transferida para valor_recuperado

Exercícios

- ➡ Certo ou errado?
Uma fila implementada em um vetor de forma não circular pode apresentar falso overflow com apenas uma posição de memória ocupada.
- ➡ Determinar o número total de elementos existentes em uma fila, a partir dos valores dos ponteiros f e r .

Tempo: 6 minutos

Exercícios (solução)



Certo ou errado?

Uma fila implementada em um vetor de forma não circular pode apresentar falso overflow com apenas uma posição de memória ocupada.

Resposta: Certo.

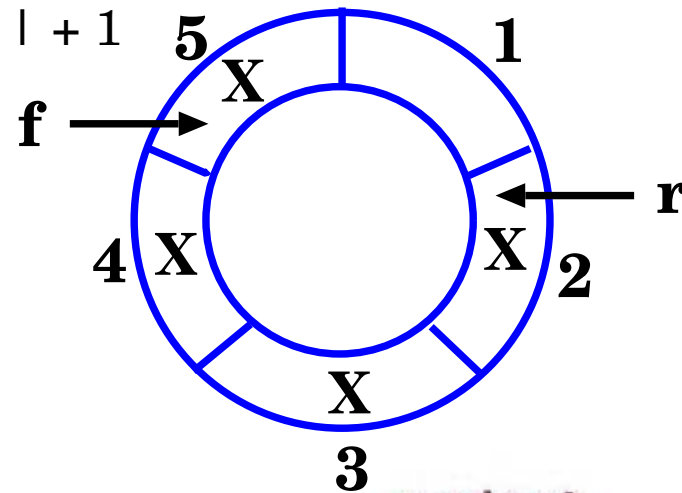
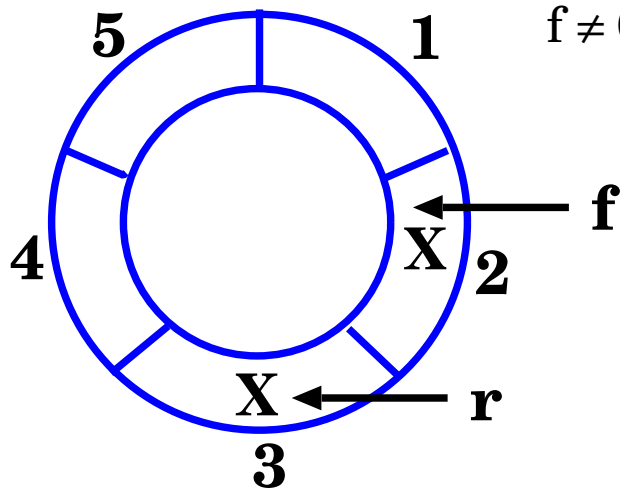


Determinar o número total de elementos existentes em uma fila, a partir dos valores dos ponteiros f e r .

Resposta: n = número de elementos da fila

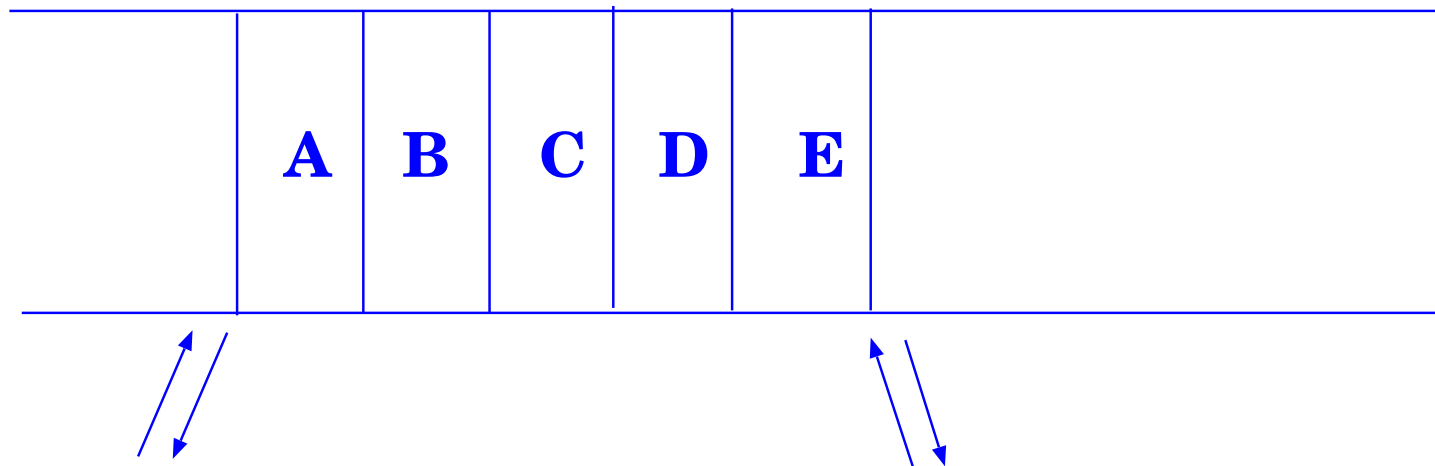
$$f = 0 \Rightarrow n = 0$$

$$f \neq 0 \Rightarrow n = |r - f| + 1$$



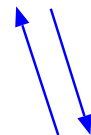
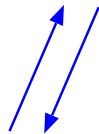
Deque ou Filas Duplas

⇒ Listas em que as inserções e remoções ocorrem somente nas suas extremidades.



⇒ Os conceitos de base e topo (pilhas) e de início e fim (filas) não se aplicam.

Tipos comuns de Deques

A**B****C****D****E****deque geral****A****B****C****D****E****deque de saída restrita**

remoção apenas
em uma das extremidades

 A single blue arrow pointing up-right at the front of the deque, indicating that only removal is allowed at this end.
**A****B****C****D****E****deque de entrada restrita**

inserção apenas
em uma das extremidades

 A single blue arrow pointing down-left at the front of the deque, indicating that only insertion is allowed at this end.


Deque

➡ Forma de armazenamento:

- ▢ Em alocação seqüencial,
 - ▢ usar vetores;
 - ▢ um ponteiro indica a posição da extremidade 1, e um outro ponteiro a da extremidade 2.

➡ Operações básicas:

- ▢ inserção;
- ▢ remoção.

Dequeues

⇒ Situações extremas:

- ▬ deque vazio;
- ▬ deque cheio.

⇒ Operações inválidas:

- ▬ inserção em deque cheio ([overflow](#))
- ▬ remoção de deque vazio ([underflow](#))

Operação de um Deque

⇒ Exemplo de operação de um deque:

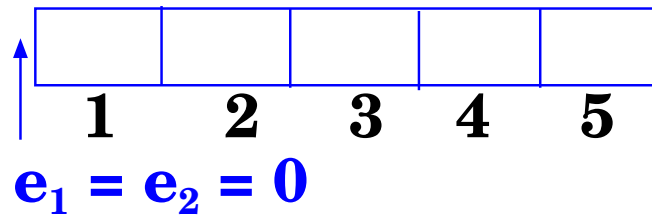
e_1 = ponteiro para a extremidade 1;

e_2 = ponteiro para a extremidade 2;

Deque vazio: $e_1 = e_2 = 0$.

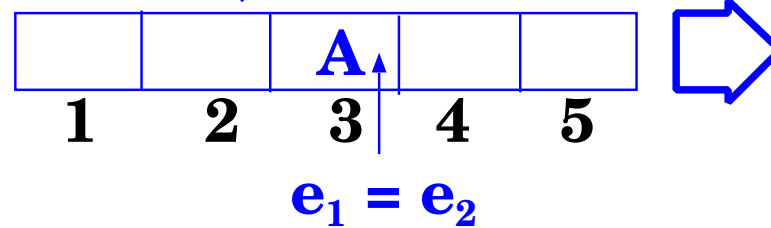
Operação de um Deque

situação 1: Deque vazio



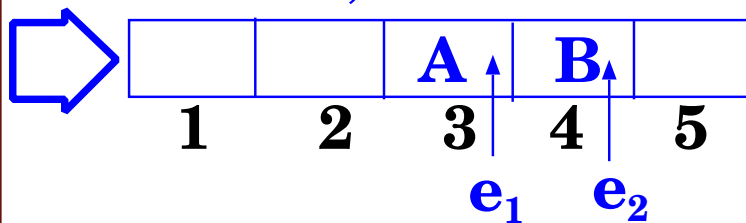
situação 2:

inserir A, extremidade 1



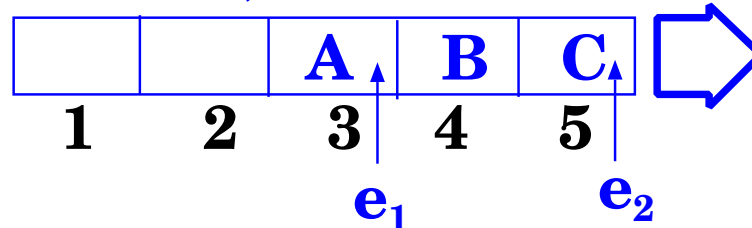
situação 3:

inserir B, extremidade 2



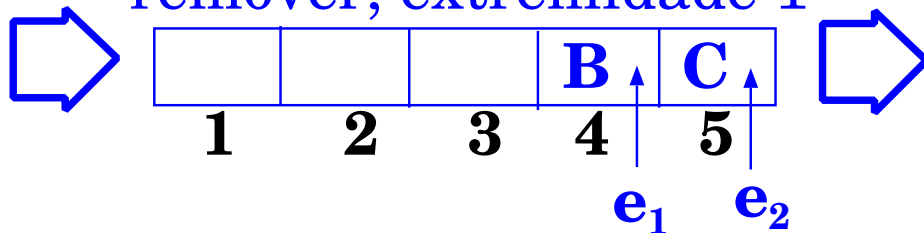
situação 4:

inserir C, extremidade 2

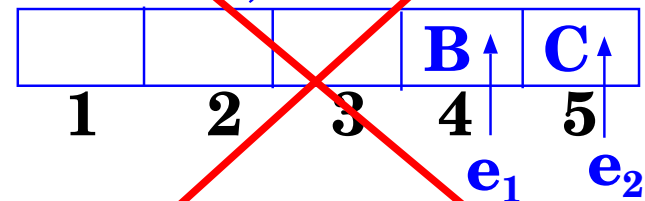


Operação de um Deque

situação 5:
remover, extremidade 1



situação 6:
inserir D, extremidade 2

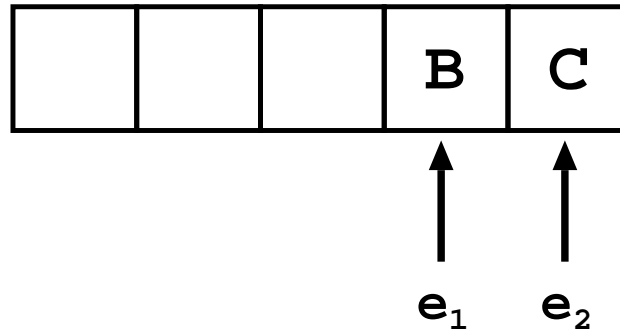


false overflow

Animação



Inserir na extremidade 2 -> falso overflow



Animar

Voltar

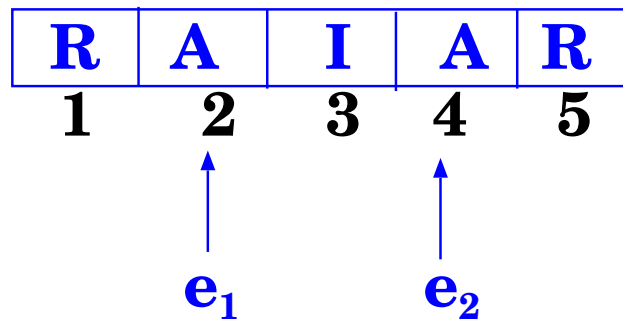
Exercício

➡ Uma palavra é um palíndromo se a sequência de letras que a forma é a mesma, quer seja lida da esquerda para a direita ou da direita para a esquerda (exemplo: raiar). Escrever um algoritmo eficiente para reconhecer se uma dada palavra é um palíndromo. Escolher a estrutura de dados conveniente para representar a palavra.

Tempo: 12 minutos

Exercício (solução)

- ⇒ Estrutura de dados: deque.
Forma de armazenamento: a palavra se encontra armazenada em um vetor D , de tamanho M .
Os ponteiros e_1 e e_2 apontam para as extremidades esquerda e direita de D , respectivamente.



- ⇒ Situação inicial: $e_1 = 1$ e $e_2 = M$

Exercício (solução)

⇒ Algoritmo: palíndromo

$e_1 := 1$

$e_2 := M$

enquanto $e_1 < e_2$ faça

se $D[e_1] \neq D[e_2]$ então $e_1 := M$

senão $e_1 := e_1 + 1$

$e_2 := e_2 - 1$

se $e_1 = M$ então resposta NÃO

senão resposta SIM

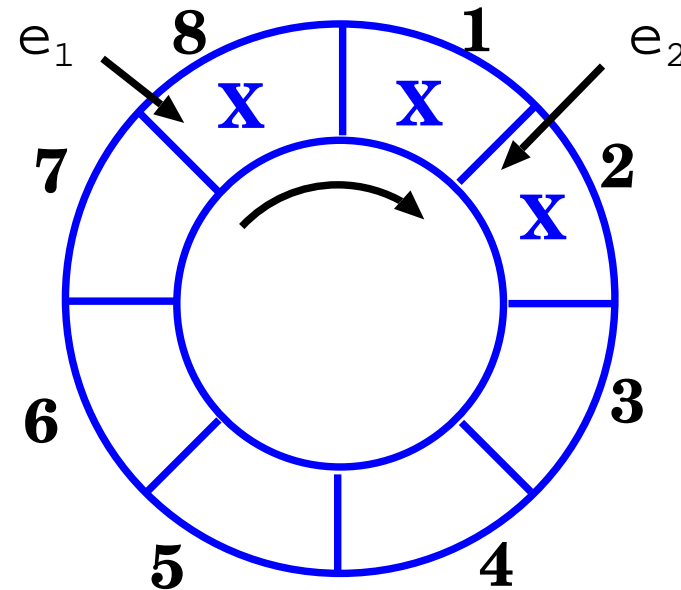
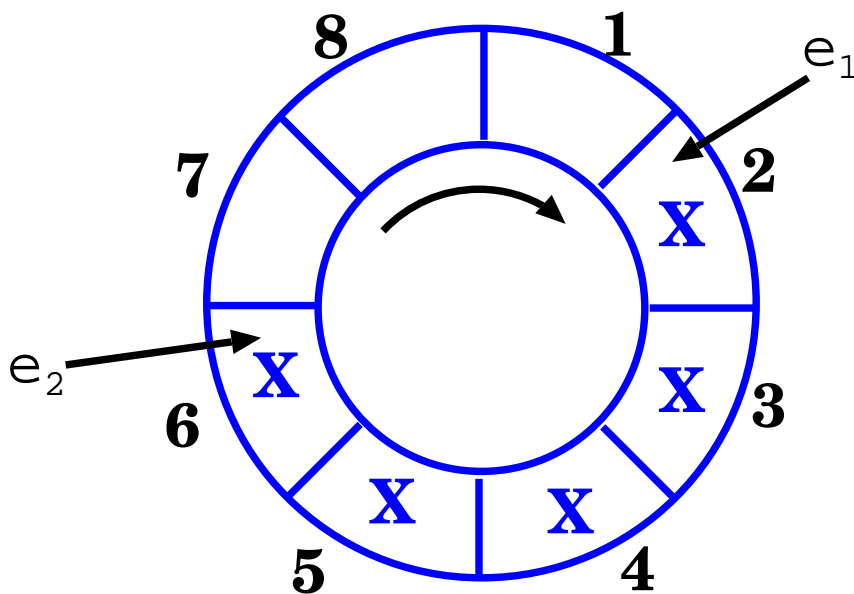
⇒ Complexidade: $O(M)$, algoritmo ótimo

Inserção e Remoção em Deques

- ➡ Mais elaborados do que no caso de filas;
Uso de dois ponteiros:
 e_1 = ponteiro para a extremidade 1;
 e_2 = ponteiro para a extremidade 2;
Deque vazio: $e_1 = e_2 = 0$
- ➡ Inserção na extr. 1: decrementar e_1 , exceto deque vazio
Inserção na extr. 2: incrementar e_2
- ➡ Remoção na extr. 1: incrementar e_1
Remoção na extr. 2: decrementar e_2 , exceto deque vazio
- ➡ Assim como no caso de filas, o deque pode se mover ao longo do vetor que o armazena.

Dequeues

- ➡ Implementação adequada: usar lista circular
- ➡ O deque está armazenado em um vetor D , de M posições
- ➡ As posições $D[1]$ e $D[M]$ são tratadas como contíguas.



- ➡ O deque se desenvolve no sentido horário, de e_1 para e_2

Exercício Final

➡ Escrever algoritmos para inserção e remoção em um deque.

O deque está armazenado em uma lista circular, de tamanho M .

Usar os ponteiros e_1 e e_2 para indicar as extremidades 1 e 2.

A condição de deque vazio é $e_1 = e_2 = 0$

Escrever a condição de deque cheio.