

Estruturas de Dados

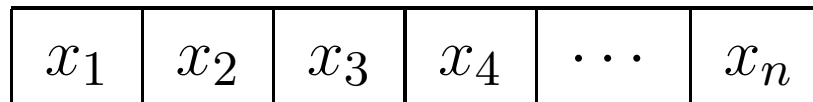
Cristina Gomes Fernandes

Listas lineares

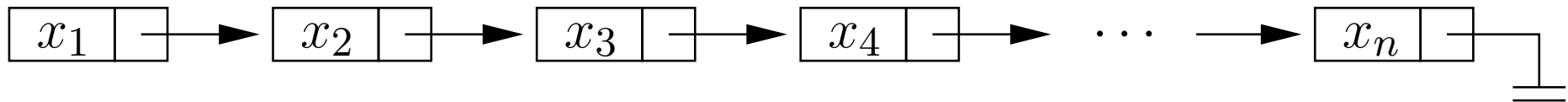
Seja (x_1, \dots, x_n) uma lista linear.

Formas de armazenamento:

● Vetor



● Lista ligada



Comparação

- **Quantidade de memória:** vetor é mais eficiente.
- **Inserção e remoção:**
em vetor, consome tempo linear no pior caso, e
em lista ligada, consome tempo constante.
- **Acesso ao elemento x_i :**
tempo constante em vetor e
tempo linear no pior caso em lista ligada.
- **Concatenação e quebra:**
tempo linear em vetor e
tempo constante em lista ligada.

Listas ligadas

Para uma célula p ,

$info(p)$: informação armazenada em p

$prox(p)$: apontador para a próxima célula da lista

Listas ligadas

Para uma célula p ,

$info(p)$: informação armazenada em p

$prox(p)$: apontador para a próxima célula da lista

BUSQUE (p, x):

recebe uma lista (apontada por) p e um elemento x e devolve (o apontador de) uma célula da lista p que contenha x , ou `NIL` caso x não ocorra na lista p .

Listas ligadas

Para uma célula p ,

$info(p)$: informação armazenada em p

$prox(p)$: apontador para a próxima célula da lista

BUSQUE (p, x):

recebe uma lista (apontada por) p e um elemento x e devolve (o apontador de) uma célula da lista p que contenha x , ou NIL caso x não ocorra na lista p .

BUSQUE (p, x)

1 $q \leftarrow p$

2 **enquanto** $q \neq \text{NIL}$ **e** $info(q) \neq x$ **faça**

3 $q \leftarrow prox(q)$

4 **devolva** q

Listas ligadas

Listas sem ordem...

INSIRA (p, x): recebe uma lista (apontada por) p e um elemento x e insere uma célula nova contendo x no fim da lista apontada por p .

Listas ligadas

Listas sem ordem...

INSIRA (p, x): recebe uma lista (apontada por) p e um elemento x e insere uma célula nova contendo x no fim da lista apontada por p .

INSIRA (p, x)

1 $t \leftarrow \text{NOVACÉLULA}(x)$

2 **se** $p = \text{NIL}$

3 **então** $p \leftarrow t$

4 **senão** $q \leftarrow p$

5 **enquanto** $\text{prox}(q) \neq \text{NIL}$ **faça**

6 $q \leftarrow \text{prox}(q)$

7 $\text{prox}(q) \leftarrow t$

Listas ligadas

REMOVA (p, x): recebe uma lista (apontada por) p e um elemento x e remove, se houver, uma célula que contenha x da lista apontada por p .

Listas ligadas

REMOVA (p, x): recebe uma lista (apontada por) p e um elemento x e remove, se houver, uma célula que contenha x da lista apontada por p .

REMOVA (p, x)

1 **se** $p \neq \text{NIL}$ **então**

2 **se** $\text{info}(p) = x$

3 **então** $t \leftarrow p$ $p \leftarrow \text{prox}(p)$ **LIBERACÉLULA**(t)

4 **senão** $q \leftarrow p$

5 **enquanto** $\text{prox}(q) \neq \text{NIL}$ **e** $\text{info}(\text{prox}(q)) \neq x$

6 **faça** $q \leftarrow \text{prox}(q)$

7 **se** $\text{prox}(q) \neq \text{NIL}$

8 **então** $t \leftarrow \text{prox}(q)$ $\text{prox}(q) \leftarrow \text{prox}(t)$

9 **LIBERACÉLULA**(t)

Listas ligadas

Listas ordenadas...

INSIRA (p, x): recebe uma lista (apontada por) p ordenada e um elemento x e insere uma célula nova contendo x na lista apontada por p .

Listas ligadas

Listas ordenadas...

INSIRA (p, x): recebe uma lista (apontada por) p ordenada e um elemento x e insere uma célula nova contendo x na lista apontada por p .

INSIRA (p, x)

```
1  $t \leftarrow \text{NOVACÉLULA}(x)$ 
2 se  $p = \text{NIL}$  ou  $x < \text{info}(p)$ 
3   então  $\text{prox}(t) \leftarrow p$     $p \leftarrow t$ 
4   senão  $q \leftarrow p$ 
5     enquanto  $\text{prox}(q) \neq \text{NIL}$  e  $\text{info}(\text{prox}(q)) < x$  faça
6        $q \leftarrow \text{prox}(q)$ 
7      $\text{prox}(t) \leftarrow \text{prox}(q)$     $\text{prox}(q) \leftarrow t$ 
```

Mais sobre listas ligadas

- apontadores e listas ligadas em C
- lista livre
- cabeça de lista
- listas circulares
- listas duplamente ligadas

Fila

Lista linear em que todas as inserções são feitas em uma das extremidades (fim) e todas as remoções são feitas na outra extremidade (início).

Fila

Lista linear em que todas as inserções são feitas em uma das extremidades (fim) e todas as remoções são feitas na outra extremidade (início).

Implementação sequencial:

um vetor $F[1 .. MAX]$ e duas variáveis inteiras *ini* e *fim*.

Fila

Lista linear em que todas as inserções são feitas em uma das extremidades (fim) e todas as remoções são feitas na outra extremidade (início).

Implementação sequencial:

um vetor $F[1 .. MAX]$ e duas variáveis inteiras ini e fim .

Operações:

- Inicialize(F, ini, fim)
- Insira(F, ini, fim, x)
- Remova(F, ini, fim)
- Primeiro(F, ini, fim)
- Vazia(F, ini, fim)

Implementação das operações

Inicialize (F, ini, fim)

1 $ini \leftarrow 0$

2 $fim \leftarrow 0$

Implementação das operações

Inicialize (F, ini, fim)

- 1 $ini \leftarrow 0$
- 2 $fim \leftarrow 0$

Insira (F, ini, fim, x)

- 1 $fim \leftarrow (fim \bmod MAX) + 1$
- 2 $F[fim] \leftarrow x$

Remova (F, ini, fim)

- 1 $ini \leftarrow (ini \bmod MAX) + 1$
- 2 **devolva** $F[ini]$

Implementação das operações

Inicialize (F, ini, fim)

- 1 $ini \leftarrow 0$
- 2 $fim \leftarrow 0$

Insira (F, ini, fim, x)

- 1 $fim \leftarrow (fim \bmod MAX) + 1$
- 2 $F[fim] \leftarrow x$

Remova (F, ini, fim)

- 1 $ini \leftarrow (ini \bmod MAX) + 1$
- 2 **devolva** $F[ini]$

Primeiro (F, ini, fim)

- 1 **devolva** $F[(ini \bmod MAX) + 1]$

Implementação das operações

Inicialize (F, ini, fim)

- 1 $ini \leftarrow 0$
- 2 $fim \leftarrow 0$

Insira (F, ini, fim, x)

- 1 $fim \leftarrow (fim \bmod MAX) + 1$
- 2 $F[fim] \leftarrow x$

Remova (F, ini, fim)

- 1 $ini \leftarrow (ini \bmod MAX) + 1$
- 2 **devolva** $F[ini]$

Primeiro (F, ini, fim)

- 1 **devolva** $F[(ini \bmod MAX) + 1]$

Vazia (F, ini, fim)

- 1 **se** $ini = fim$
- 2 **então devolva** VERDADE
- 3 **senão devolva** FALSO

Aplicações

- cálculo de distância
(Problema do ratinho)
- implementação de Round-Robin
(escalonamento de processos numa CPU, RRDtool)

Round-Robin é uma técnica usada para distribuição de carga entre servidores ou para interpolar valores em um gráfico.