

LISTAS DUPLAMENTE ENCADEADAS

1

Profa. Fabrícia Damando Santos
fabriadamando@gmail.com

LISTAS

○ Listas encadeadas

- caracteriza-se por formar um encadeamento simples entre os elementos: cada elemento armazena um ponteiro para o próximo elemento da lista
- O encadeamento simples também dificulta a retirada de um elemento da lista.
- Temos que percorrer a lista, elemento por elemento, para encontrarmos o elemento anterior, pois, dado um determinado elemento, não temos como acessar diretamente seu elemento anterior.

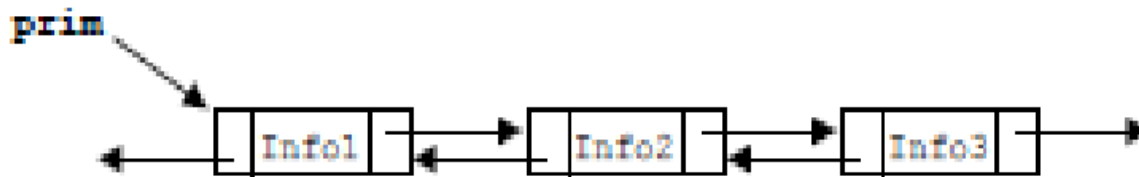


SOLUÇÃO:

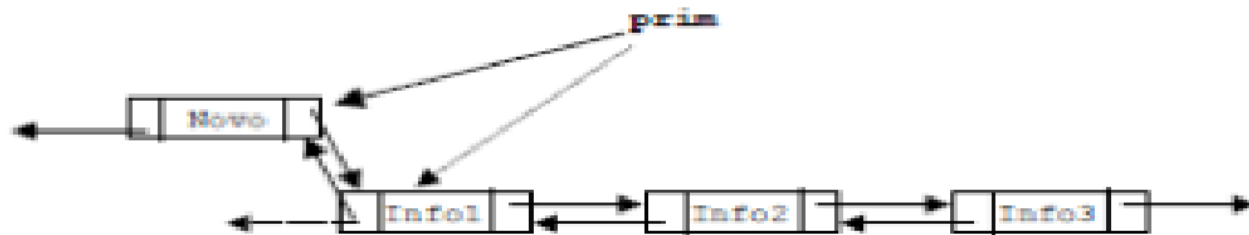
Listas Duplamente Encadeadas

LISTAS DUPLAMENTE ENCADEADAS

- Cada elemento tem:
 - um ponteiro para o próximo elemento
 - um ponteiro para o elemento anterior
- Desta forma, dado um elemento, podemos acessar ambos os elementos adjacentes: o próximo e o anterior.



INSERÇÃO DE UM NOVO ELEMENTO NA LISTA



Inserção de um novo elemento no início da lista.

- Encadear um novo elemento no início da lista
- Ele tem:
 - Como próximo elemento o antigo primeiro elemento da lista;
 - como anterior o valor NULL.
 - o novo elemento passa a ser o primeiro da lista, e deve ser retornado como valor da lista atualizada

```
struct lista2 {  
    int info;  
    struct lista2* ant;  
    struct lista2* prox; };  
  
typedef struct Lista2 Lista2;
```

LISTA VAZIA

- Inicio = null
- Fim = null

CASO LISTA VAZIA

- Inserção no início da lista
- O elemento inserido será o primeiro e último
- Se início == null
 - Início = novo
 - Fim = novo
 - Novo->próximo = null
 - novo.->ant = null

CASO LISTA VAZIA

- Inserção no fim da lista
- O elemento inserido será o primeiro e último
- Se início == null
 - Início = novo
 - Fim =novo
 - Novo->prox = null
 - Novo.ant = null

CASO A LISTA JÁ TENHA ELEMENTOS

- Se o elemento a ser inserido for no **final** da lista
 - $\text{Fim} \rightarrow \text{prox} = \text{novo}$
 - $\text{Novo} \rightarrow \text{ant} = \text{fim}$
 - $\text{Novo} \rightarrow \text{prox} = \text{null}$
 - $\text{Fim} = \text{novo}$

CASO A LISTA JÁ TENHA ELEMENTOS

- Se o elemento a ser inserido for no **início** da lista
 - Novo->prox = inicio
 - Inicio ->ant = novo
 - Novo->ant = null
 - Inicio = novo

BUSCA ELEMENTO NA LISTA - CONSULTA

- A função de busca
 - recebe a informação referente ao elemento que queremos buscar
 - tem como valor de retorno o ponteiro do nó da lista que representa o elemento.
 - Caso o elemento não seja encontrado na lista, o valor retornado é NULL.

REMOÇÃO NO INÍCIO

- Se $\text{aux} == \text{inicio}$
 - $\text{Inicio} = \text{aux} \rightarrow \text{prox}$
 - Se $(\text{inicio} \neq \text{null})$
 - $\text{Inicio} \rightarrow \text{ant} = \text{null}$
 - $\text{Aux} = \text{inicio}$

REMOÇÃO NO FIM

- Se $\text{auxo} == \text{fim}$
 - $\text{Fim} = \text{fim} \rightarrow \text{ant}$
 - $\text{Fim} \rightarrow \text{prox} = \text{null}$
 - $\text{Aux} = \text{null}$

REMOÇÃO NO MEIO

- $Aux \rightarrow ant \rightarrow prox = aux \rightarrow prox$
- $Aux \rightarrow prox \rightarrow ant = aux \rightarrow ant$
- $Aux = aux \rightarrow prox$

REMOÇÃO DE ELEMENTO DA LISTA

- A função de remoção fica mais complicada, pois temos que acertar o encadeamento duplo
- Usar a função de busca para localizar o elemento
- Em seguida acertar o encadeamento liberando o elemento ao final
 - $p \rightarrow \text{ant} \rightarrow \text{prox} = p \rightarrow \text{prox};$
 - $p \rightarrow \text{prox} \rightarrow \text{ant} = p \rightarrow \text{ant};$
- o anterior passa a apontar para o próximo e o próximo passa a apontar para o anterior
- Quando **p** apontar para um elemento no **meio da lista** - as duas atribuições acima são suficientes acertar o encadeamento
- Se **p for o primeiro**, não podemos escrever $p \rightarrow \text{ant} \rightarrow \text{prox}$, pois $p \rightarrow \text{ant}$ é NULL;
 - além disso, temos que atualizar o valor da lista, pois o primeiro elemento será removido

EXERCÍCIOS

- Implemente uma lista duplamente encadeada
 - Inserção início
 - Inserção fim
 - Busca
 - Exclusão no início
 - Exclusão no fim
 - Exclusão de um determinado valor
 - Impressão