



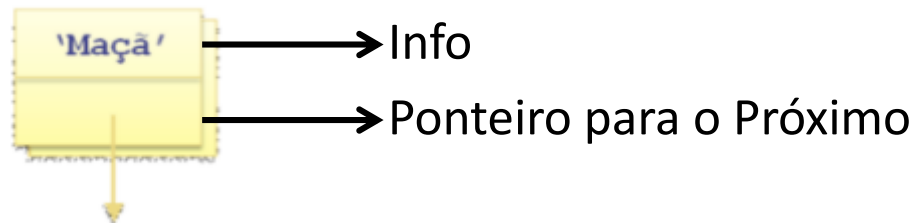
Estruturas de Dados

Lista Duplamente Encadeada



- Um **nó** em uma **Lista Encadeada Simples** possui basicamente dois itens:

- Informação armazenada
- Ponteiro para o próximo

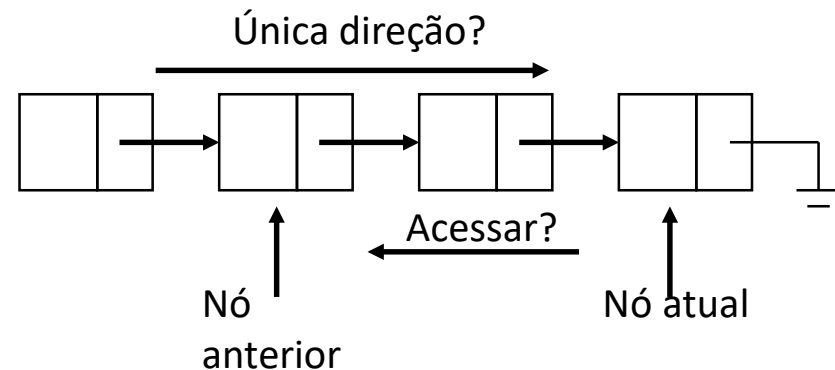


- **Se** a informação não for um dado simples (int, float, double...)?
 - Criar vários campos, um para cada informação. Isto é, **struct**
- TADs da Lista Encadeada Simples!

Lista Duplamente Encadeada



- Quais as principais **limitações** de uma da **Lista Encadeada Simples**?
 - Percorre em uma única direção
 - Não acessar o nó anterior/adjacente
- Como solucionar tais limitações?
 - Criar uma variável ponteiro para o nó anterior
 - Lista Duplamente Encadeada



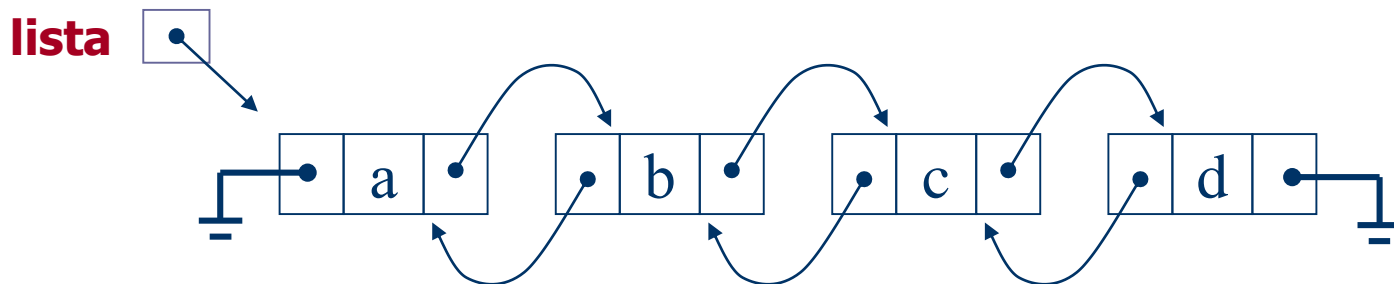


- Características da **lista duplamente encadeada**
 - Cada **nó** da lista tem um **ponteiro** tanto para o **próximo** quanto para o **anterior**
 - Dado um **nó**, pode-se encontrar com facilidade os elementos que lhe são **adjacentes**
 - Se tiver um ponteiro para o **último** nó da lista, pode-se facilmente percorrer **ordem inversa**
 - Proveem flexibilidade, permitindo que os itens sejam rearrumados eficientemente, semelhante a lista encadeada simples



- Definição?

É uma extensão da lista encadeada simples, que consiste de uma sequência linear de nós dinamicamente alocados e encadeados por meio de ponteiros (anterior e próximo)





- A lista duplamente encadeada é constituída de um nó com 3 campos:



- A manipulação da lista é mais complexa, porém algumas operações são diretamente beneficiadas
- Um conjunto maior de ligações precisam ser atualizadas



- Aplicações que utilizam Lista Duplamente Encadeada?
 - Histórico do navegador
 - CTRL + Z e CTRL + Y
 - Aplicações que necessitam acessar informação de um elemento antecessor
 - CTRL + T e CTRL + SHIFT + T



- Principal diferença da **lista duplamente encadeada** para a **lista encadeada**!
 - É a maior facilidade para **navegação**!

Se **não existe** a necessidade de se **percorrer** a lista de trás para frente, a lista simplesmente encadeada é a mais interessante, pois é mais simples



TADs

- Criar
- Vazia
- Listagem
- Inserir Início, Fim e Meio
- Remover Início, Fim e Meio

Lista Duplamente Encadeada – Implementação *struct*



```
struct ITEM{  
    int valor;  
};
```

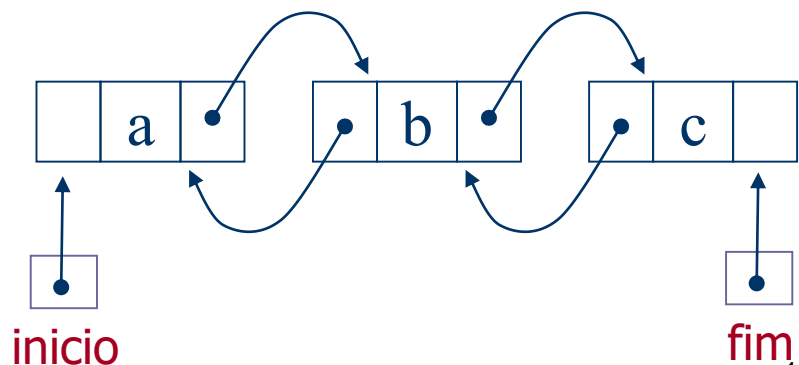
O que está sendo feito?



```
struct NO{  
    ITEM item;  
    struct NO *prox;  
    struct NO *ant;  
};
```

O que está sendo feito?

```
struct LISTA_DUPLAMENTE_LIGADA{  
    NO *inicio;  
    NO *fim;  
};
```



Lista Duplamente Encadeada – Criar lista



- Pré-condição: Nenhuma
- Pós-condição: inicia a estrutura de dados
- Como implementar?

```
int cria(LISTA_DUPLAMENTE_LIGADA *lista){
```

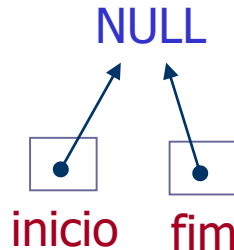
```
    lista->inicio = NULL;
```

```
    lista->fim = NULL;
```

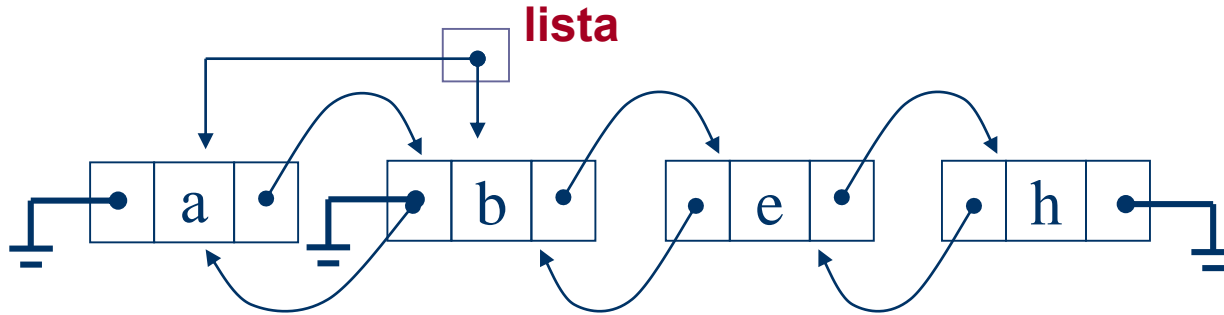
```
    return 1;
```

```
}
```

O que está sendo feito?



Lista Duplamente Encadeada – Inserir no início



Lista Duplamente Encadeada – Essa inserção é no início?



```
int insere(LISTA_DUPLAMENTE_LIGADA *lista, ITEM *item){  
    NO *noNovo = (NO*)malloc(sizeof(NO)); //cria um novo nó
```

```
    if(noNovo != NULL){  
        noNovo->item = *item;  
        noNovo->prox = lista->inicio;  
        noNovo->ant = NULL;
```

```
    if(lista->inicio != NULL)
```

```
        lista->inicio->ant = noNovo;
```

```
    else  
        lista->fim = noNovo;
```

```
    lista->inicio = noNovo;  
    return 1;
```

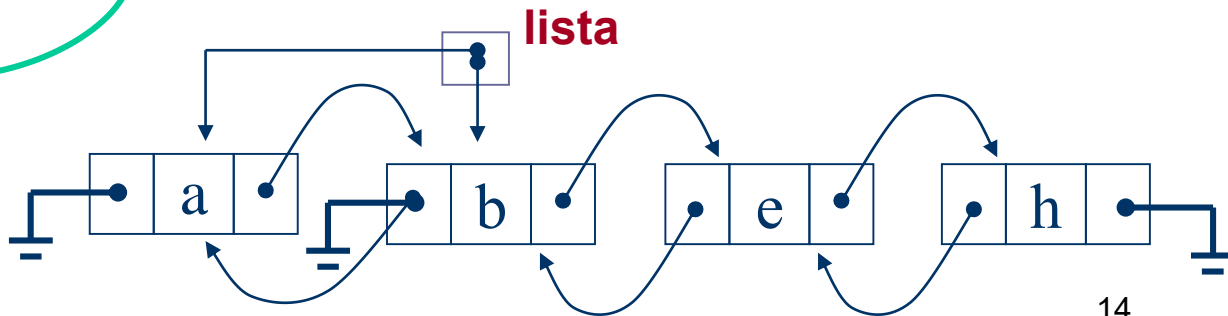
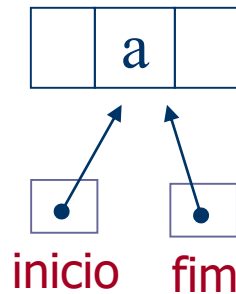
```
    return 0;
```

```
}
```

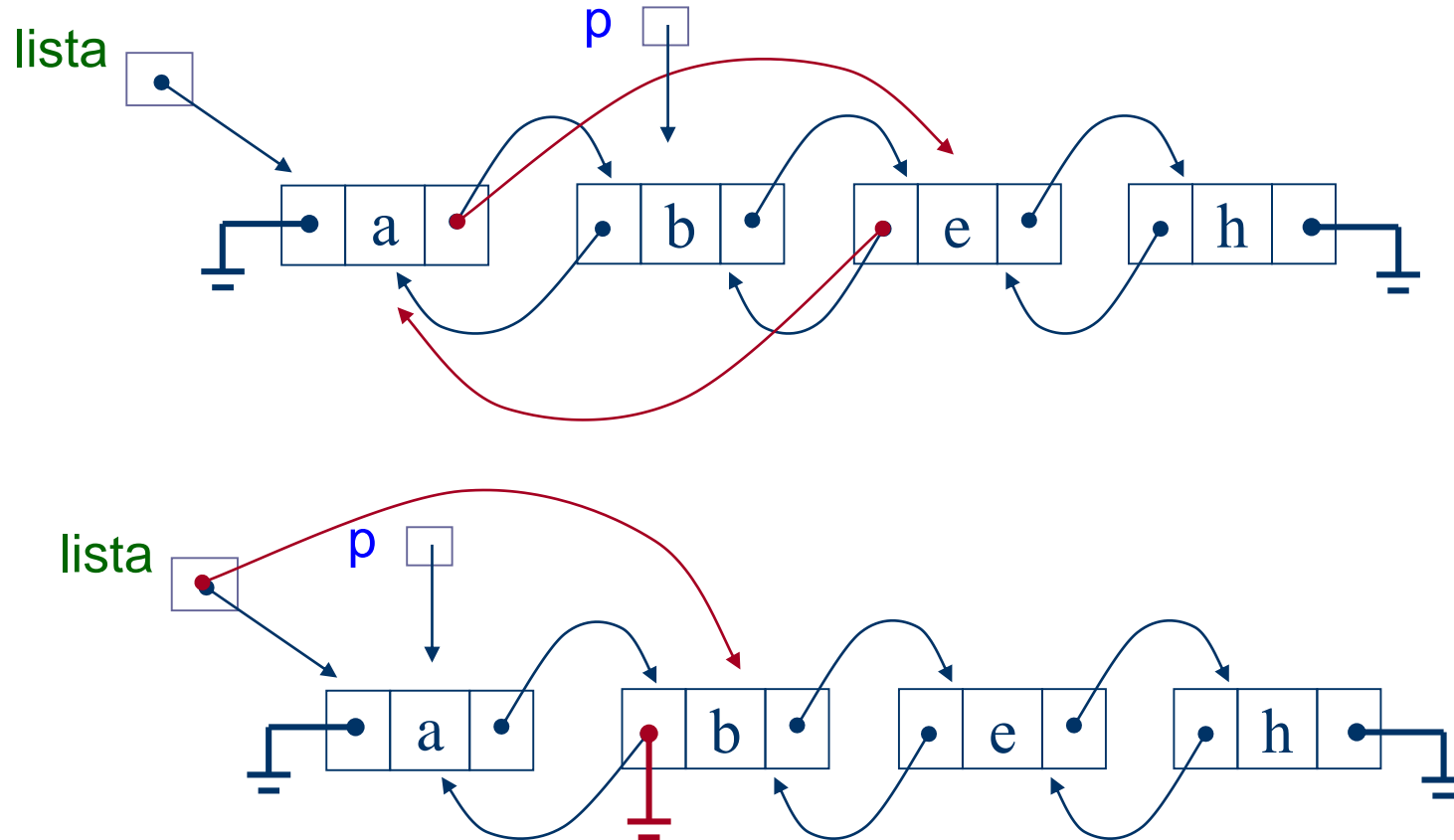
O que está sendo feito?

O que está sendo feito?

O que está sendo feito?



Lista Duplamente Encadeada – Remoção



Lista Duplamente Encadeada – Essa remoção é no início?



```
int remove(LISTA_DUPLAMENTE_LIGADA *lista){
```

```
    NO* noAux;
```

```
    if(!vazia(lista)){
```

```
        noAux = lista->inicio;
```

```
        if(lista->inicio == lista->fim){
```

```
            lista->inicio = NULL;
```

```
            lista->fim = NULL;}
```

```
        else{
```

```
            lista->inicio->prox->ant = NULL;
```

```
            lista->inicio = lista->inicio->prox;}
```

```
        free(noAux);
```

```
        return 1;
```

```
    }
```

```
    return 0;
```

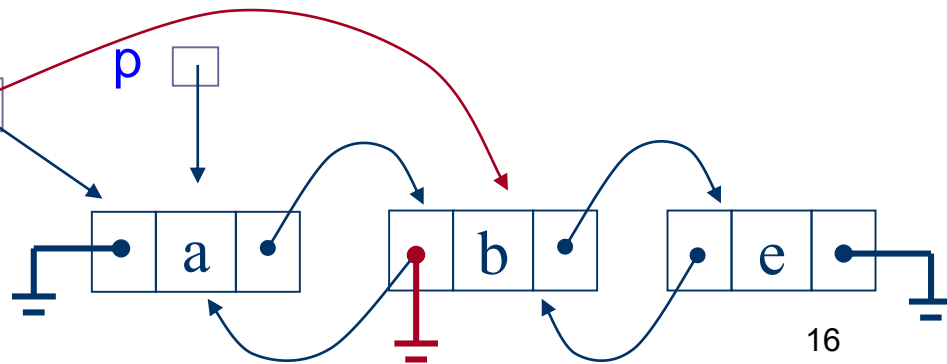
```
}
```

O que está linha faz?

O que está sendo feito?

O que está sendo feito?

lista



Lista Duplamente Encadeada – Essa remoção é no fim?

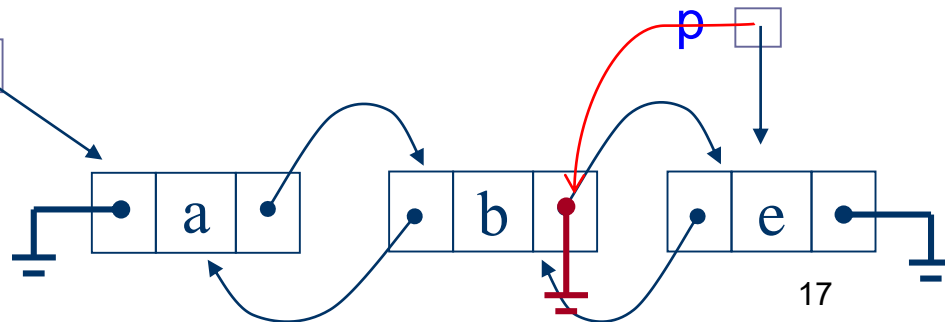


```
int remove(LISTA_DUPLAMENTE_LIGADA *lista){  
    if(!vazia(lista)){  
        NO* noAux = lista->fim;  
  
        if(lista->inicio == lista->fim){  
            lista->inicio = NULL;  
            lista->fim = NULL;  
        }  
        else{  
            lista->fim->ant->prox = NULL;  
            lista->fim = lista->fim->ant;  
        }  
  
        free(noAux);  
        return 1;  
    }  
    return 0;  
}
```

O que está linha faz?

O que está sendo feito?

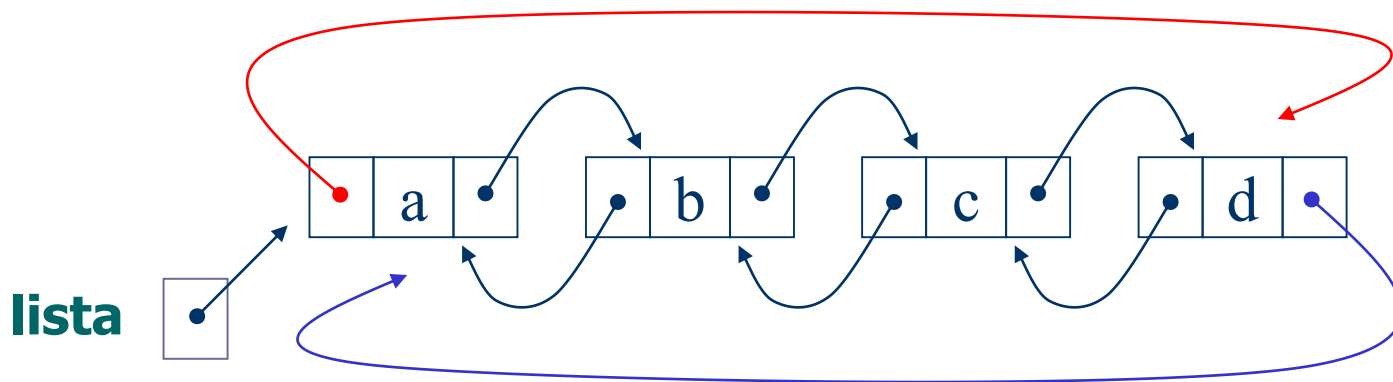
lista



Lista Duplamente Encadeada



- O que acontece se o **anterior do primeiro** é o **último** e o **próximo do último** é o **primeiro**
- Lista Duplamente Encadeada e **Circular**





Estruturas de Dados

Lista Duplamente Encadeada