Listas Duplamente Encadeadas

Disciplina: Estrutura de Dados

Luciano Moraes Da Luz Brum

Universidade Federal do Pampa – Unipampa – Campus Bagé

Email: lucianobrum18@gmail.com

Tópicos

- Revisão – Listas Simplesmente Encadeadas
- Listas Duplamente Encadeadas.
  - Motivação;
  - Operações sobre listas;
  - Implementação;

Estruturas de Dados

Revisão – Listas Simplesmente Encadeadas

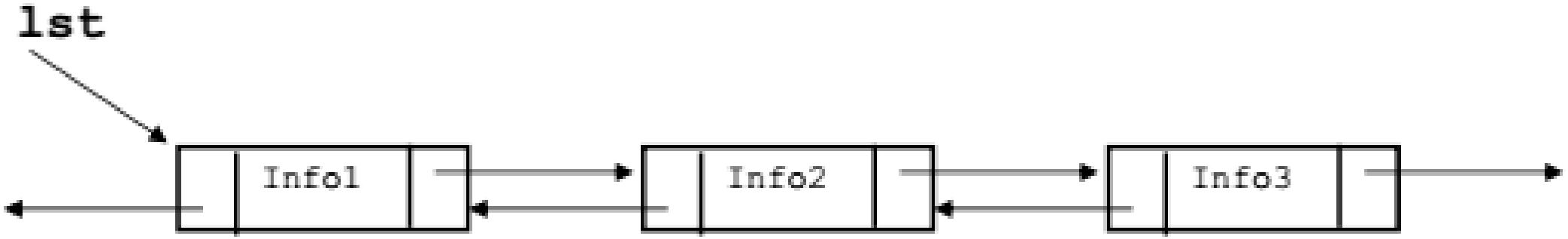
- Estrutura de dados dinâmicas
- Podem crescer ou diminuir à medida que são inseridos ou removidos dados.
- Sequência de elementos encadeados, que possuem dois campos:
  - Informação a ser armazenada;
  - Ponteiro para o próximo elemento da lista;
- A lista possui um ponteiro para o primeiro nodo.
- O ponteiro para o último elemento da lista é NULL.

Estruturas de Dados

Problemas:

- Não temos como percorrer eficientemente os elementos da lista em ordem inversa.
- Para excluir um elemento, mesmo se tivermos um ponteiro para ele, devemos percorre a lista toda para encontrar o nó anterior.
- Para a solução destas dificuldades, surgem as listas duplamente encadeadas.

Listas Duplamente Encadeadas



- Cada elemento tem um ponteiro para o próximo elemento e um ponteiro para o elemento anterior.
- Dado um elemento, é possível acessar o próximo e o anterior.
- Dado um ponteiro para o último elemento da lista, é possível percorrer a lista em ordem inversa

Listas Duplamente Encadeadas


```
struct listadupla{
        int info;
        struct listadupla *ant;
        struct listadupla *prox;
};typedef struct listadupla Lista;
```

Listas Duplamente Encadeadas

- Das operações com listas encadeadas, veremos:
  - Inicialização/Criação da lista vazia;
  - Inserção de elementos;
  - Busca de elementos;
  - Impressão dos elementos na tela;
  - Remoção de elementos;
  - Liberação da estrutura da memória;

Listas Duplamente Encadeadas

- Operação para criação da lista:

```
Lista *criavazia(){
    return NULL;
}
```

Listas Duplamente Encadeadas

- Operação para inserção na lista:

```
/*Inserção no início: retorna a lista atualizada*/
Lista *lst2_insere (Lista* lst, int val)
{
    Lista *novo = (Lista*)malloc(sizeof(Lista));
    novo->info = val;
    novo->prox = lst;
    novo->ant = NULL;
    /*verifica se lista não estava vazia*/
    if(lst != NULL)
        lst->ant = novo;
    return novo;
}
```

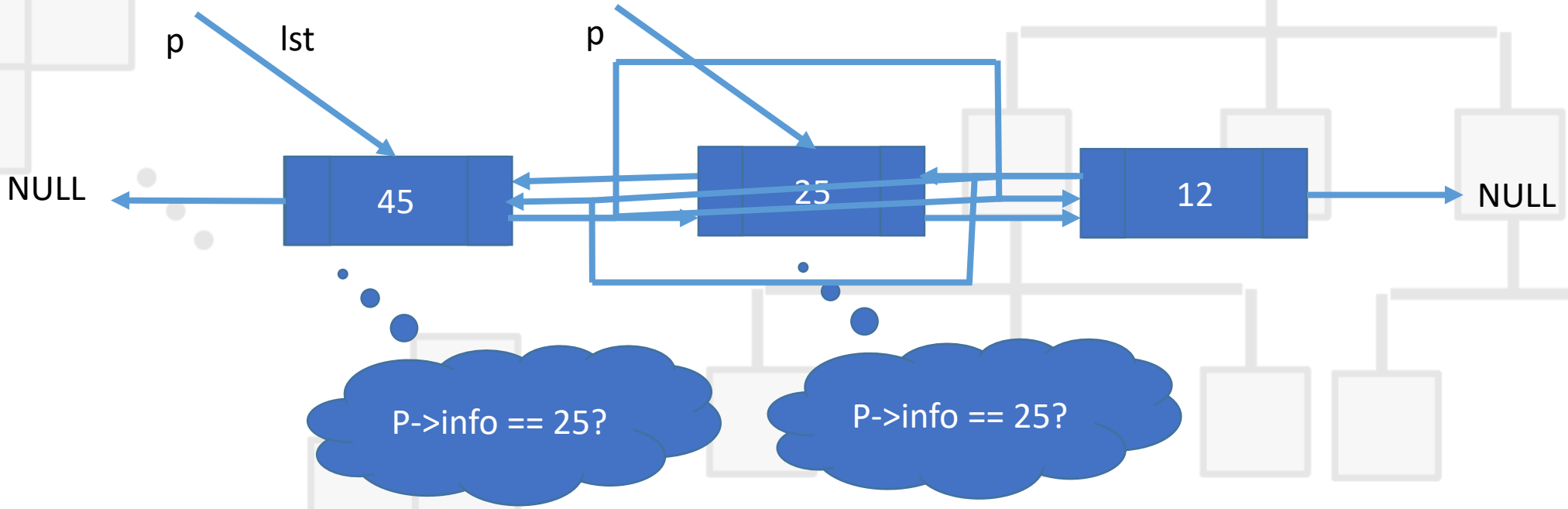
Listas Duplamente Encadeadas

- Operação para busca de um elemento na lista:

```
/*Inserção no início: retorna a lista atualizada*/
Lista *lst2_busca (Lista* lst, int val)
{
    Lista *p;
    for(p = lst; p != NULL; p = p->prox){
        if(p->info == val);
            return p;
    }
    return NULL;
}
```

Listas Duplamente Encadeadas

- Operação para remoção de um elemento da lista:
  - Exemplo: Remover o número 25.



Listas Duplamente Encadeadas

- Operação para remoção de elementos da lista:

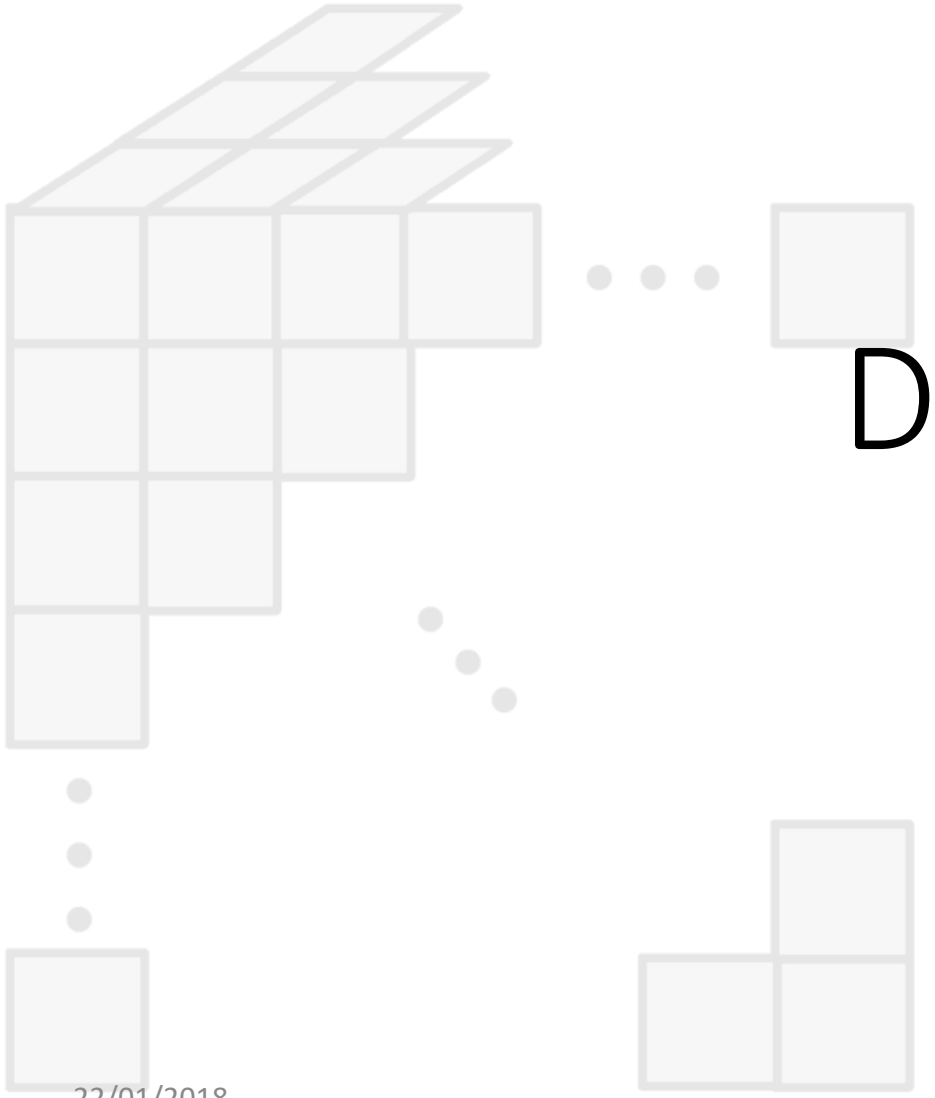
```
/*função retira: retira elemento da lista*/
Lista *lst2_retira (Lista* lst, int val)
{
    /*procura o elemento na lista*/
    Lista *p = busca(lst, val);

    if(p == NULL){
        return lst; /*não achou o elemento: retorna lista inalterada*/
    }
    /*retira elemento do encadeamento*/
    if(lst == p){ /*testa se é o primeiro elemento*/
        lst = p->prox;
    }else{
        p->ant->prox = p->prox;
    }
    if(p->prox != NULL){ /*testa se é o último elemento*/
        p->prox->ant = p->ant;
    }
    free(p);
    return lst;
}
```

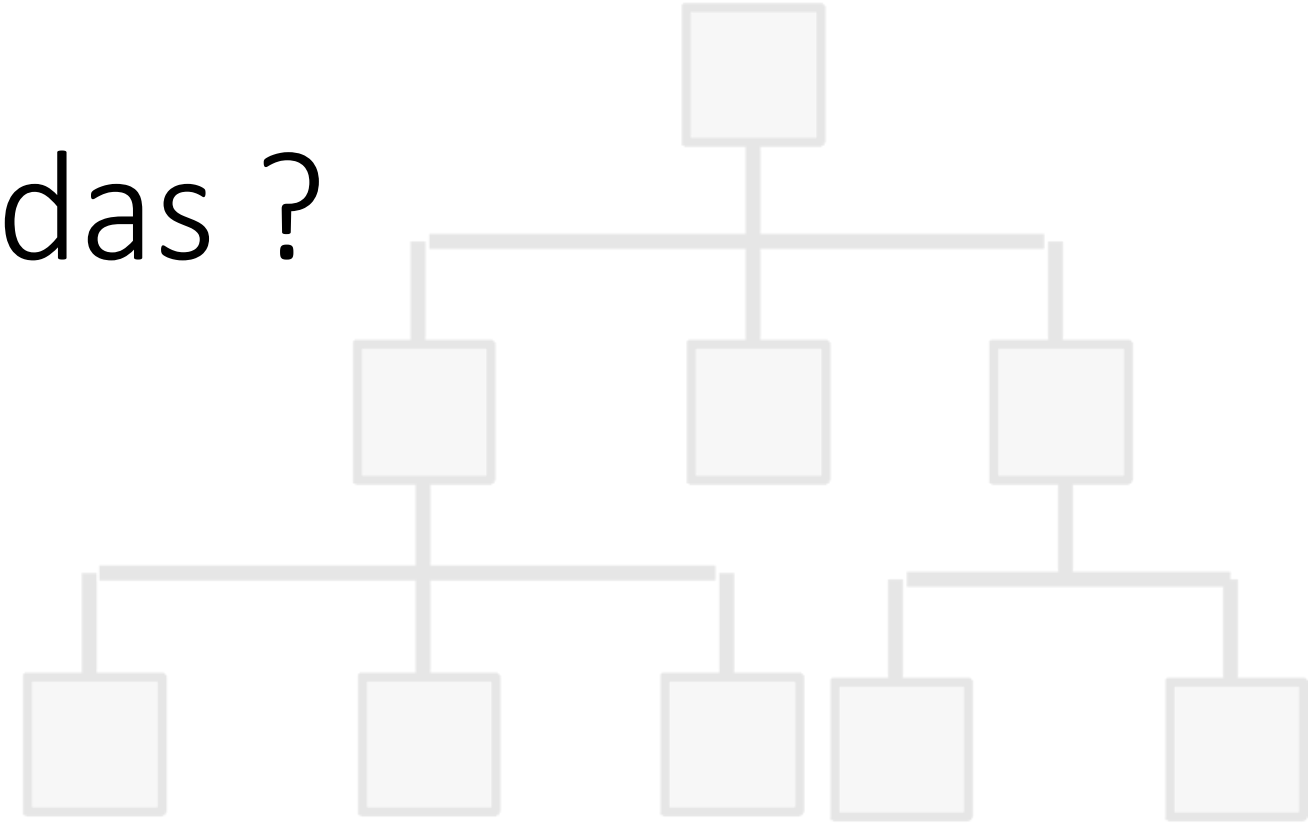
Listas Duplamente Encadeadas

- Operação para liberar a estrutura da lista da memória:

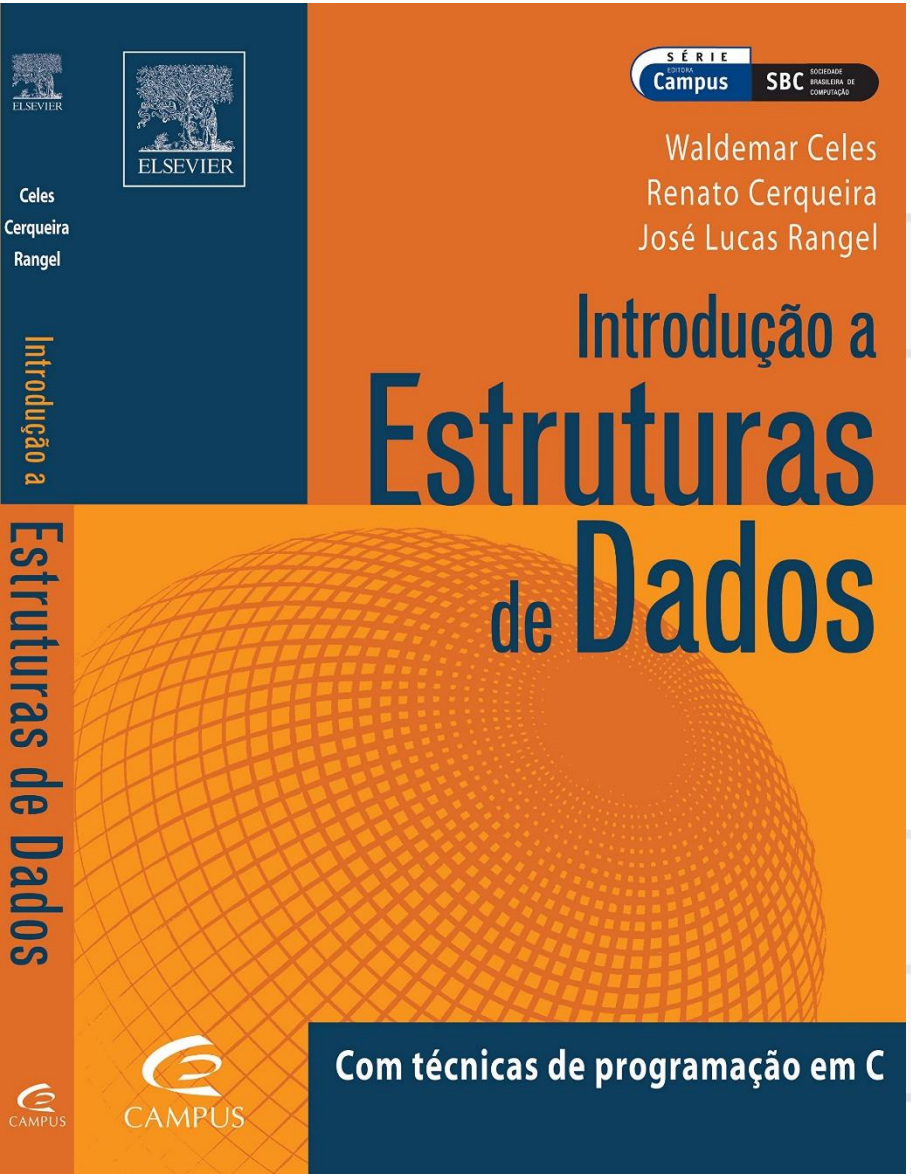
```
void lst2_libera (Lista *lst)
{
    Lista *p = lst;
    while (p != NULL){
        Lista *t = p->prox; /*guarda referência p/ próximo elemento*/
        free(p);
        p = t; /*faz p apontar para o próximo*/
    }
}
```

Dúvidas ?





Referências



- CELES, Waldemar; CERQUEIRA, Renato; RANGEL, José Lucas. Introdução a Estruturas de Dados ~~c~~om técnicas de programação em C. Rio de Janeiro: Elsevier (Campus), 2004. 4ª Reimpressão. 294 p.