

Estruturas de Dados Clássicas – Listas – Parte 2

Prof. Bárbara Quintela

barbaraquintela@pucminas.cesjf.br





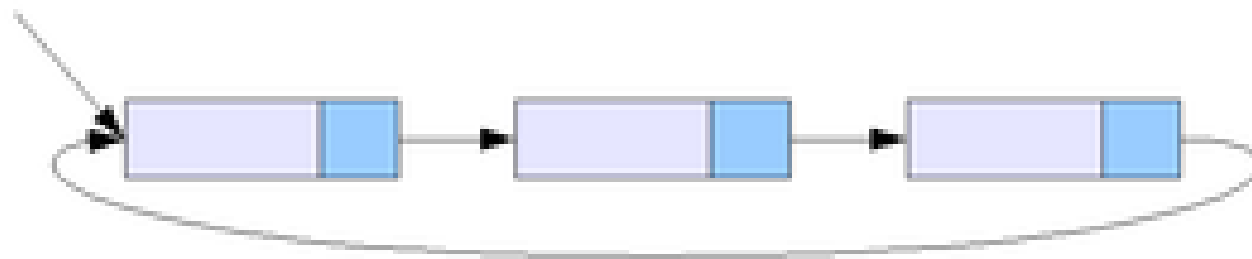
Listas Circulares

- Algumas aplicações necessitam representar conjuntos cíclicos
 - Ex: aplicação que manipula figuras geométricas, as arestas que delimitam uma face podem ser agrupadas em uma estrutura circular
 - Podemos usar listas circulares.
- Em uma lista circular → o ultimo elemento tem como próximo o primeiro da lista
 - Não faz sentido falar em primeiro e ultimo...



Listas Circulares

- A lista pode ser representada por um ponteiro para um elemento inicial qualquer da lista
- Para percorrer elementos visitamos todos a partir do ponteiro do elemento inicial ate alcançar o mesmo novamente
- Se a lista vazia o ponteiro para o elemento inicial vale NULL





Código para percorrer lista circular

```
void lcirc_imprime(Lista* l)
    Lista* p = l; /*faz apontar para no inicial*/
    /*testa se lista não e vazia*/
    if (p) do {
        printf("%d\n", p → info); /* imprime */
    }while (p != 1)                /* avança */
}
```



Exemplo

- Escrever um programa que crie uma lista **circular** a partir de n valores lidos, percorra a lista e escreva os valores na lista, **inverta** a lista e escreva novamente tais valores.



Lista Duplamente Encadeada

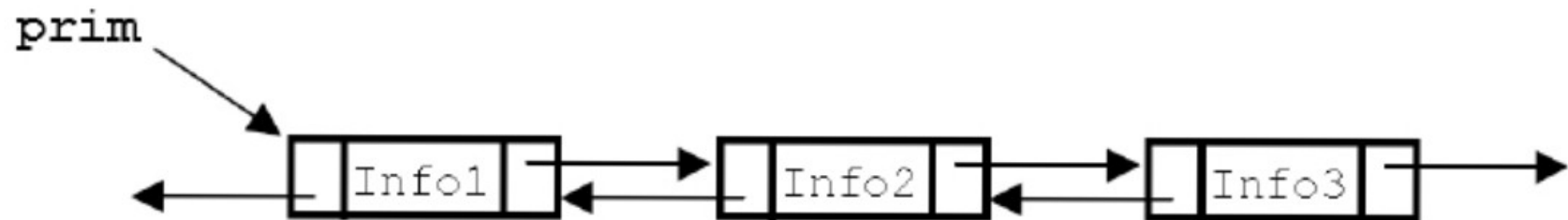
Até o momento vimos encadeamento **simples**

- Dificulta percorrer sentido contrário
- Dificulta remover elemento da lista
 - Mesmo com ponteiro devemos percorrer todos os elementos para encontrar o anterior
- Lista **duplamente encadeada**
 - Cada elemento tem ponteiro para o próximo e para o anterior



Listas duplamente encadeadas

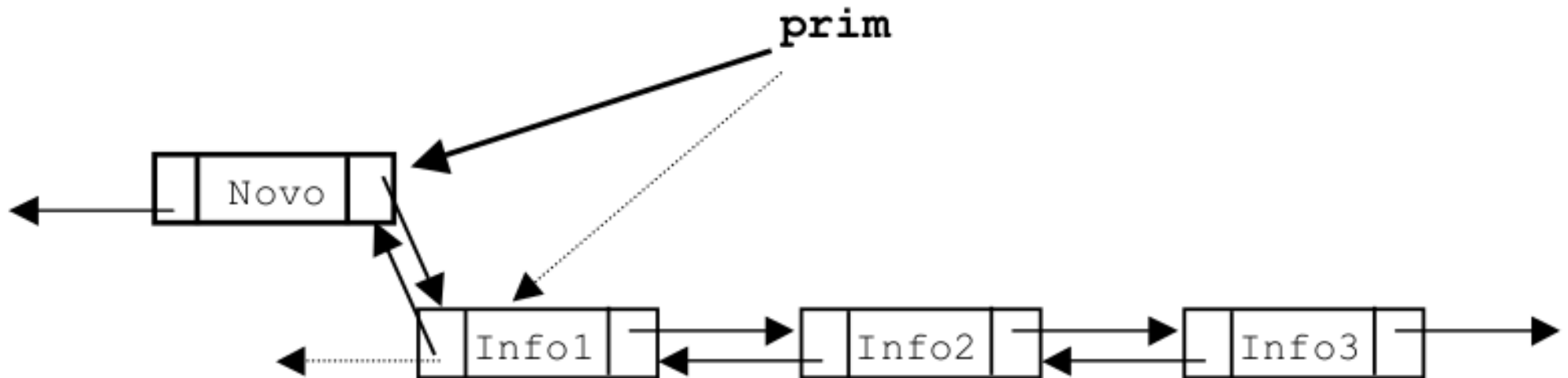
- Estrutura de uma lista duplamente encadeada





Lista Duplamente Encadeada

- Inserir elemento no início:





Exemplo

- Implemente funções para inserir e retirar um elemento de uma **lista duplamente ligada**



Exercícios - Revisão

- Escrever uma função que receba três listas ordenadas, sendo duas já preenchidas e, preencha a terceira lista, inicialmente vazia, com os elementos das listas recebidas em ordem crescente.
- Implemente uma função que receba duas pilhas, p1 e p2, e passe todos os elementos da pilha p2 para o topo da pilha p1. Ao final a p2 estará vazia e p1 conterá todos os elementos das duas pilhas. A função deve obedecer o protótipo:
`void concatena_pilhas(Pilha* p1, Pilha* p2);`