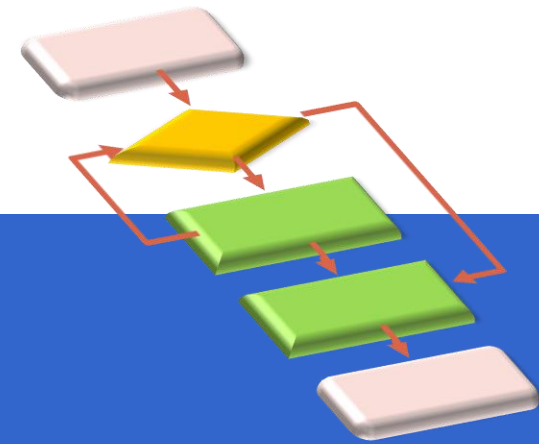




INSTITUTO DE COMPUTAÇÃO



Algoritmos II

ALOCAÇÃO DINÂMICA DE MEMÓRIA

Tipos Especiais de Listas Encadeadas Dinâmicas

Prof.^a Vanessa de Oliveira Campos

Listas

- Listas são conjuntos de elementos, objetos, variáveis, tarefas, ou qualquer coisa que se possa enumerar e formar um conjunto.



Listas

- Casos especiais de listas simplesmente encadeadas:
 - Fila;
 - Pilha;
 - Circular.



Fila



**INSTITUTO DE
COMPUTAÇÃO**

Fila

- Uma fila é um conjunto de itens a partir do qual podem-se eliminar itens numa extremidade (chamada início da fila) e no qual podem-se inserir itens na outra extremidade (chamada final da fila).



Fila

- Numa fila existe uma regra básica a ser seguida:
 - Primeiro a Chegar é o Primeiro a Sair;
 - Do inglês: FIFO – First In, First Out;
- Um novo elemento da fila somente pode ser inserido na última posição (fim da fila);
- Um elemento só pode ser removido da primeira posição (início da fila).



Fila

Exemplos de uso de filas na computação:

- Filas de impressão:

- Impressoras tem uma fila, caso vários documentos sejam impressos, por um ou mais usuários, os primeiros documentos impressos serão de quem enviar primeiro;

- Filas de processos:

- Vários programas podem estar sendo executados pelo sistema operacional. O mesmo tem uma fila que indica a ordem de qual será executado primeiro;



Fila

Variações de Filas:

- Fila de Prioridades:

- Cada item tem uma prioridade. TNoS mais prioritários podem ser atendidos antes, mesmo não estando no início da fila;

- Fila Circular:

- Neste tipo de fila, os elementos nem sempre são removidos ao serem atendidos, mas voltam ao fim da fila para serem atendidos novamente mais tarde.



Pilhas



**INSTITUTO DE
COMPUTAÇÃO**

Pilhas

- **Regra básica:**
 - Insere-se elementos no topo da pilha;
 - Remove-se ou utiliza-se apenas o elemento que estiver no topo da pilha.
- Do inglês: LIFO - *Last In, First Out*;
- **Último a entrar, primeiro a sair.**



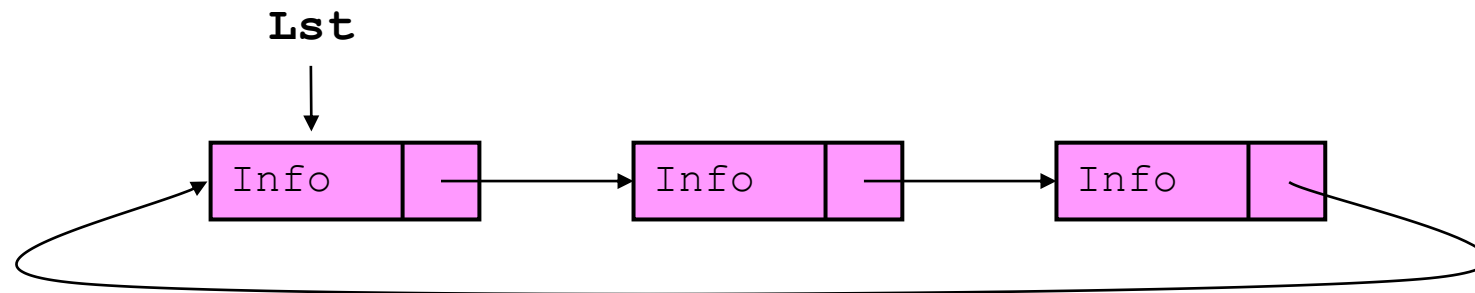
Pilhas

- São exemplos de uso de pilha em um sistema:
 - Funções recursivas em compiladores;
 - Mecanismo de desfazer/refazer dos editores de texto;
 - Navegação entre páginas Web.



Listas Circulares

- Lista circular:
 - o último elemento tem como próximo o primeiro elemento da lista, formando um ciclo
 - a lista pode ser representada por um ponteiro para um elemento inicial qualquer da lista



Listas Circulares

- Exemplo - Função para imprimir uma lista circular
 - visita todos os elementos a partir do ponteiro do elemento inicial até alcançar novamente esse mesmo elemento
 - se a lista é vazia, o ponteiro para um elemento inicial é NULL

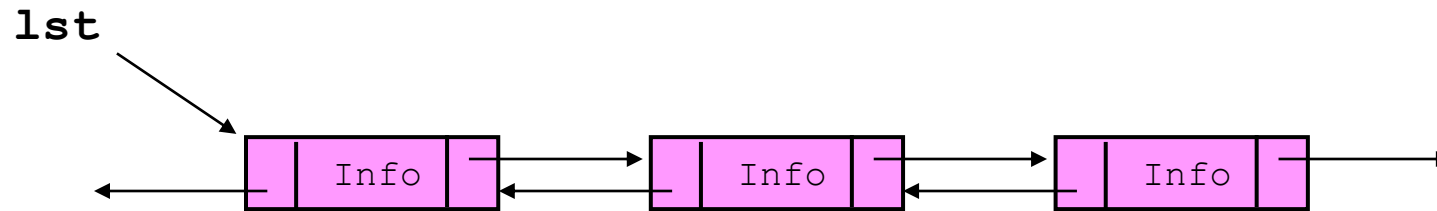


Listas Circulares

```
/* função imprime: imprime valores dos elementos */
void lccirc_imprime (TNo* lst)
{
    TNo* p;
    p = lst;           /* faz p apontar para o nó inicial */
    if (p!=NULL)        /* testa se lista não é vazia */
    {
        do /* percorre lista */
        {
            printf("%d\n", p->info); /* imprime informação do nó */
            p = p->prox;             /* avança para o próximo nó */
        } while (p != lst);
    }
}
```



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.



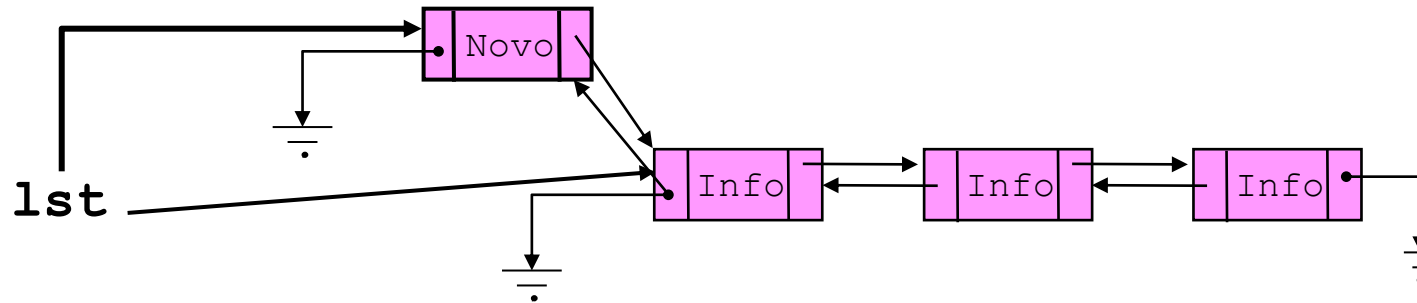
Listas Duplamente Encadeadas – Exemplo

```
struct lista2 {  
    int info;  
    struct lista2* ant;  
    struct lista2* prox;  
};  
typedef struct lista2 TLista2;
```

- lista encadeada armazenando valores inteiros
 - struct **lista2**
 - estrutura dos nós da lista
 - tipo **TLista2**
 - tipo dos nós da lista




```
/* inserção no início: retorna a lista atualizada */
TLista2* lst2_inserere (TLista2* lst, int val)
{
    TLista2* novo = (TLista2*) malloc(sizeof(TLista2));
    novo->info = val;
    novo->prox = lst;
    novo->ant = NULL;
    if (lst != NULL) /* verifica se lista não estava vazia */
        lst->ant = novo;
    return novo;
}
```



Listas Duplamente Encadeadas

Exemplo: Função para retirar um elemento da lista.

- `p` aponta para o elemento a retirar;
- se `p` aponta para um elemento no meio da lista:
 - o anterior passa a apontar para o próximo: `p->ant->prox = p->prox;`
 - o próximo passa a apontar para o anterior: `p->prox->ant = p->ant;`
- se `p` aponta para o último elemento:
 - não é possível escrever `p->prox->ant`, pois `p->prox` é `NULL`
- se `p` aponta para o primeiro elemento
 - não é possível escrever `p->ant->prox`, pois `p->ant` é `NULL`
 - é necessário atualizar o valor da lista, pois o primeiro elemento pode ser removido.



```
/* função retira: remove elemento da lista */
TLista2* lst2_retira (TLista2* lst, int val)
{
    TLista2* p = busca(lst, val);

    if (p == NULL) /* não achou o elemento */
        return lst; /* retorna lista inalterada */

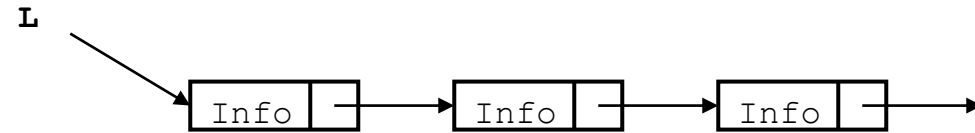
    /* retira elemento do encadeamento */
    if (lst == p) /* se é o primeiro elemento */
        lst = p->prox;
    else
        p->ant->prox = p->prox;

    if (p->prox != NULL) /* se não é o último elemento */
        p->prox->ant = p->ant;

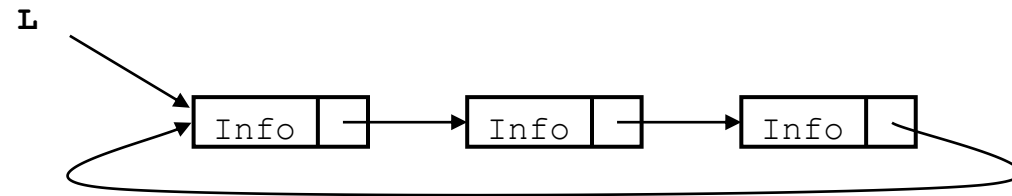
    free(p);
    return lst;
}
```

Resumo

Listas encadeadas



Listas circulares



Listas duplamente encadeadas

