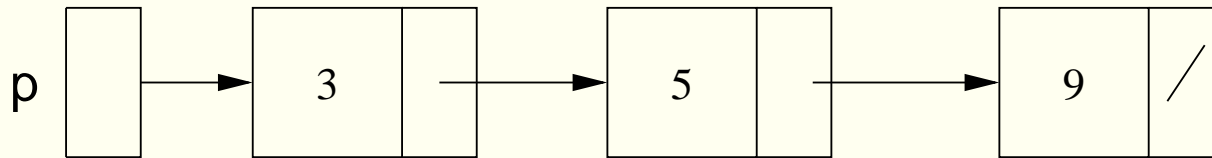


**MC102 - Algoritmos e programação de computadores**

# **Aula 28: Listas ligadas**

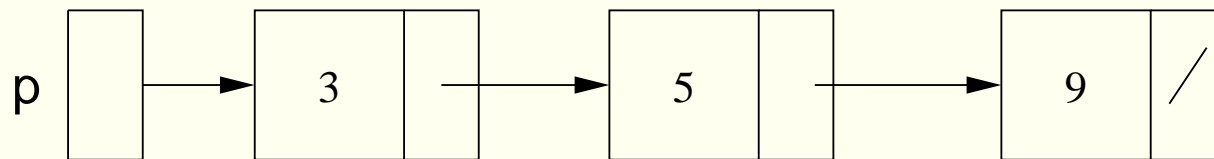
# Lista Ligada



Seqüência de nós, em que cada nó possui

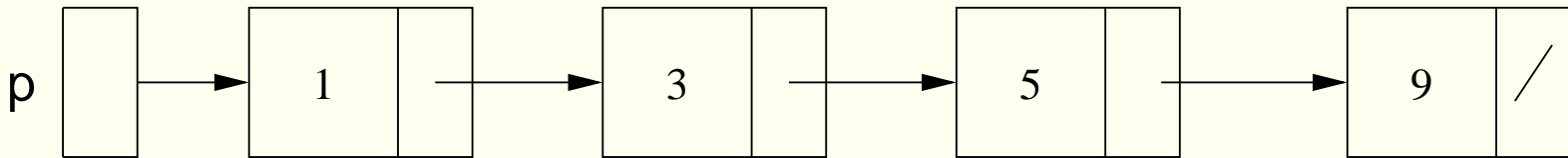
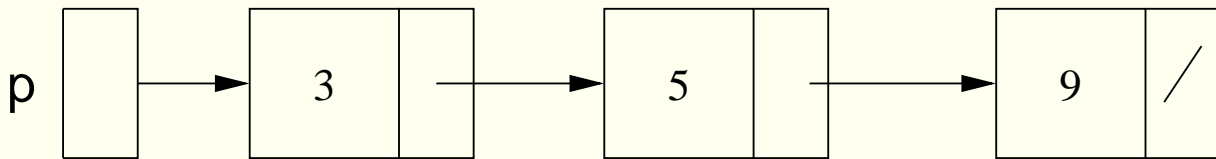
- um ou mais campos de informação e
- um apontador para o próximo nó da lista.
  
- Veja o código: lista.c

## Como liberar uma lista



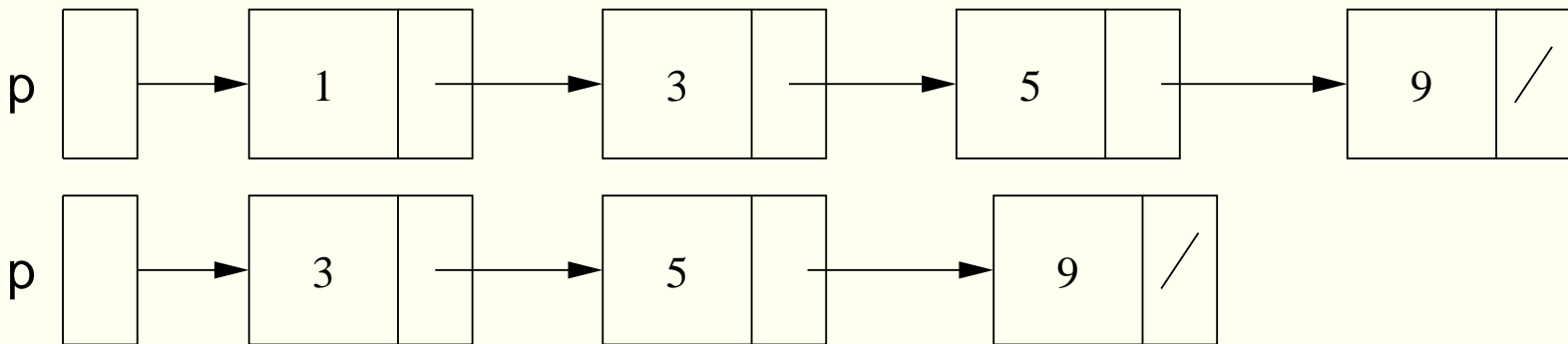
- `free(p)` libera apenas o primeiro nó;
- é necessário percorrer a lista liberando todos os nós;
- Veja o código: `libera.c`

## Inserção no começo da lista



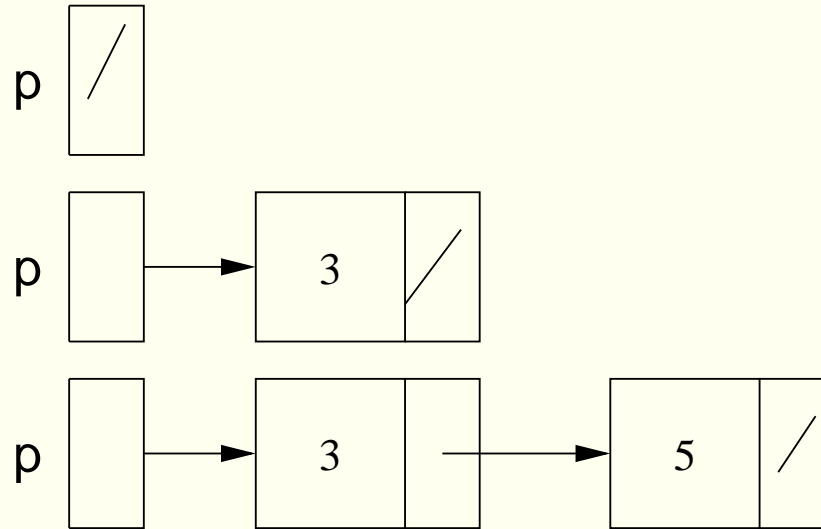
- o endereço armazenado em p deve ser alterado
- Veja o código: `insere_comeco.c`

## Remoção no começo da lista



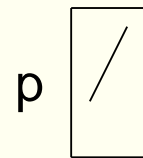
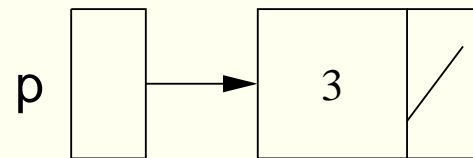
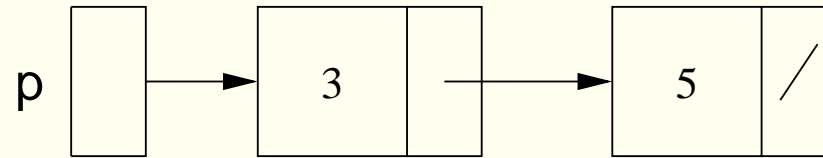
- o endereço armazenado em `p` deve ser alterado
- Veja o código: `remove_comeco.c`

# Inserção no final da lista



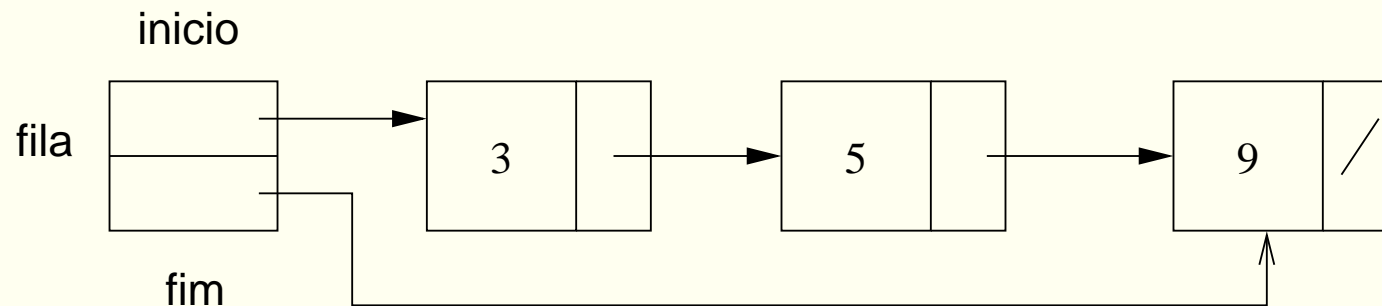
- o endereço armazenado em `p` será alterado caso a lista esteja vazia ou
- o campo prox do último elemento será alterado.
- Veja o código: `insere_final.c`

## Remoção no final da lista



- o campo prox do último elemento será alterado caso a lista contenha mais de um elemento ou
- o endereço armazenado em p será alterado.
- Veja o código: `remove_final.c`

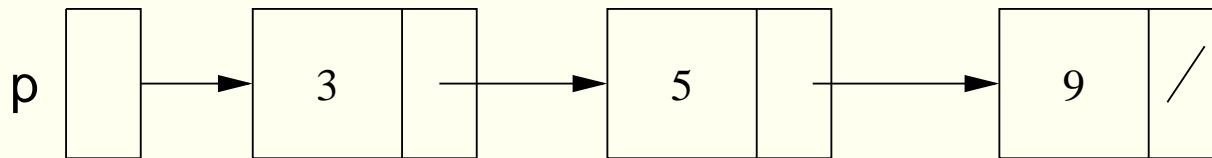
# Filas



- Estrutura de dados em que
  - inserções são feitas no final e
  - remoções são feitas no começo
- Veja o código `fila.c`

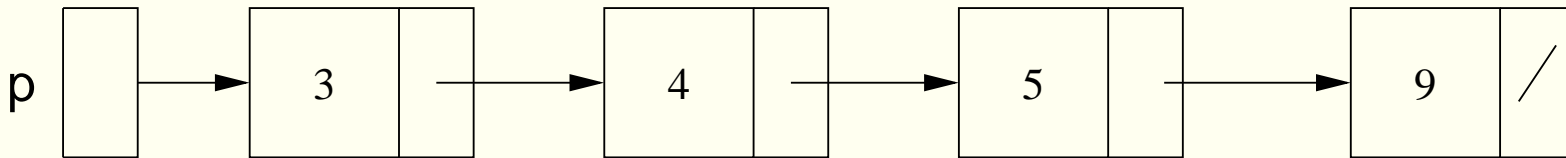
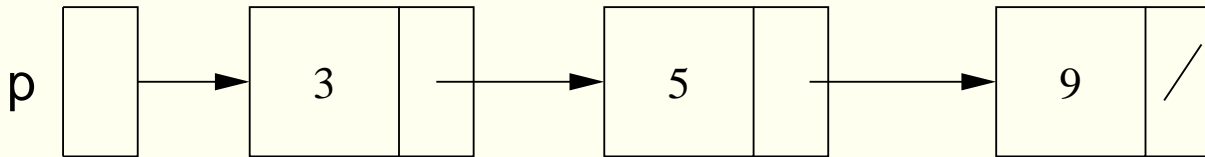
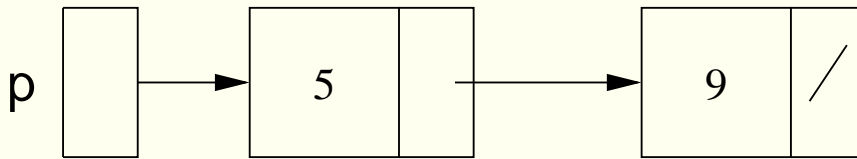


# Pilha



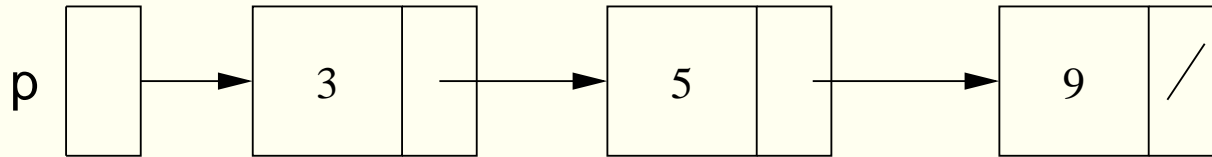
- Estrutura de dados em que
  - inserções são feitas no começo e
  - remoções são feitas no começo
- Como seria o código pilha.c?

## Inserção ordenada



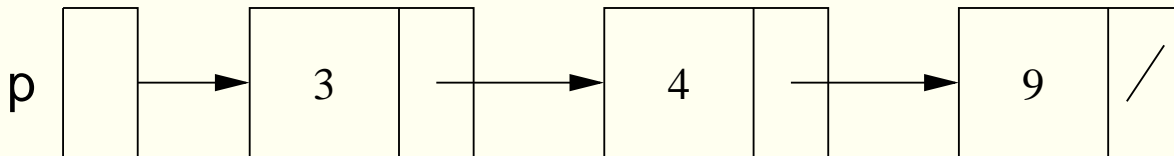
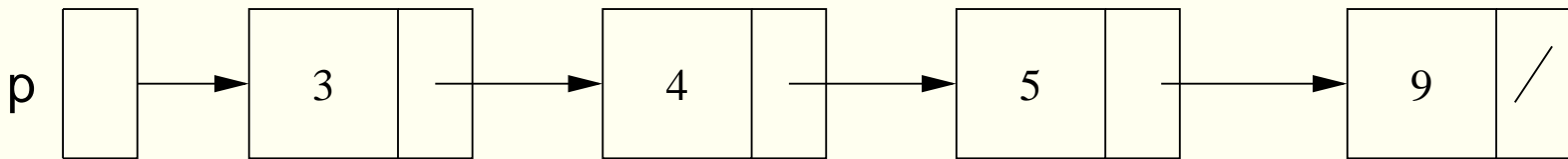
- O valor de p pode ser alterado caso a inserção ocorra no início da lista;
- A inserção também pode ocorrer no meio ou no final da lista.

# Busca



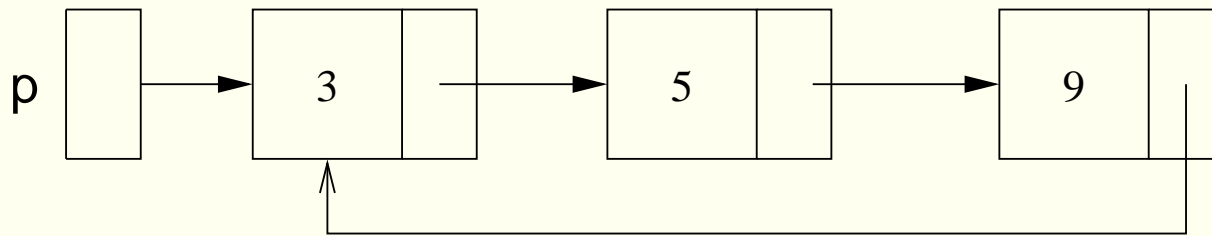
- A lista deve ser percorrida, procurando-se o elemento desejado.
- A busca é interrompida quando:
  - O elemento foi encontrado ou
    - \* chegamos ao final da lista (lista não ordenada)
    - \* encontramos um elemento maior que o desejado (lista ordenada)

# Remoção



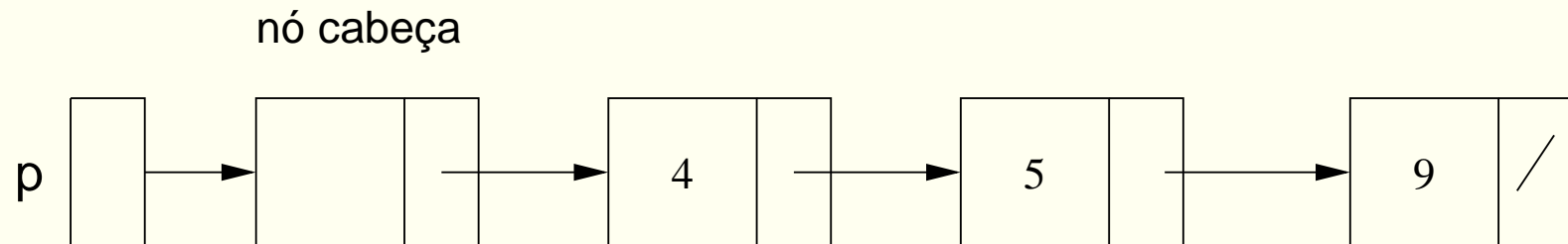
- Procura-se o elemento a ser removido
- Atualiza-se a lista, caso ele tenha sido encontrado

## Lista circular



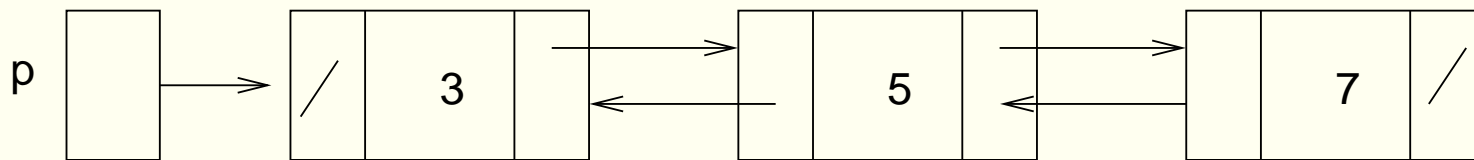
- O último elemento aponta para o primeiro.

# Lista com nó cabeça



- O primeiro nó não armazena elementos da lista, mas pode armazenar dados sobre a lista, como o número de nós da lista.
- Simplifica as implementações. Reescreva todas as operações de inserção e remoção e comprove!

## Lista duplamente ligada



- Cada nó tem um apontador para o próximo elemento e para o elemento anterior na lista