

## MC-102 — Aula 20

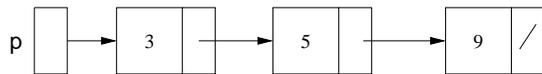
### Listas Ligadas II

Instituto de Computação – Unicamp

Segundo Semestre de 2009

Navigation icons

## Busca



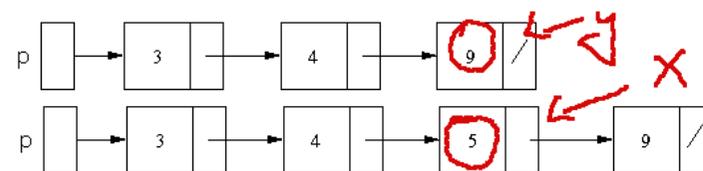
- A lista deve ser percorrida, procurando-se o elemento desejado.
- A busca é interrompida quando:
  - O elemento foi encontrado ou
    - chegamos ao final da lista (lista não ordenada)
    - encontramos um elemento maior que o desejado (lista ordenada)
- Ver: busca.c

Navigation icons

## Roteiro

Navigation icons

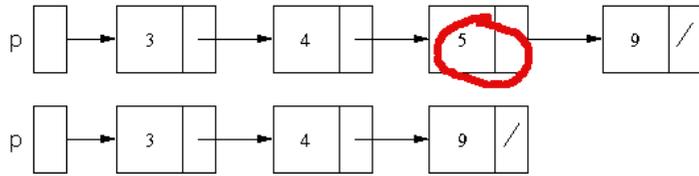
## Busca e Inse



- Inserir na lista um nó com conteúdo  $x$ 
  - antes do nó com conteúdo  $y$ , se  $y$  estiver na lista
  - no final da lista caso contrário
- O valor de  $p$  pode ser alterado caso a inserção ocorra no início da lista (lista vazia ou  $y$  é o primeiro nó)
- A inserção pode ocorrer no meio ou no final da lista.
- Ver: busca-insere.c

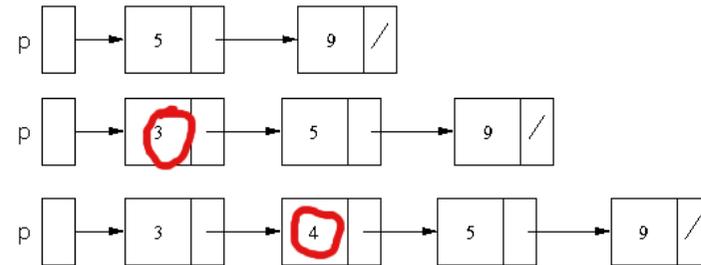
Navigation icons

## Busca e Remove



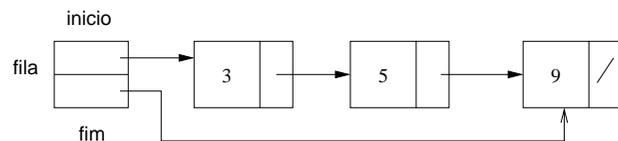
- Procura-se o elemento a ser removido
- Atualiza-se a lista, caso ele tenha sido encontrado
- Ver: busca-remove.c

## Inserção ordenada



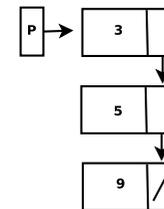
- O valor de p pode ser alterado caso a inserção ocorra no início da lista;
- A inserção também pode ocorrer no meio ou no final da lista.
- Ver: insere-ordenado.c

## Filas



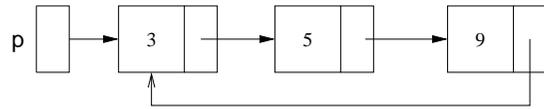
- Estrutura de dados em que
  - inserções são feitas no final e
  - remoções são feitas no começo
- Veja o código fila.c

## Pilha



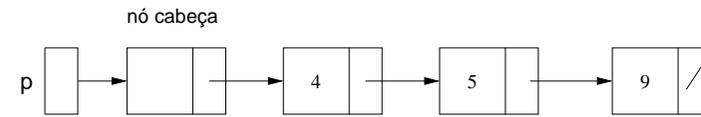
- Estrutura de dados em que
  - inserções são feitas no começo e
  - remoções são feitas no começo
- Ver pilha.c

## Lista circular



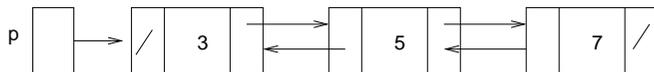
- O último elemento aponta para o primeiro.

## 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