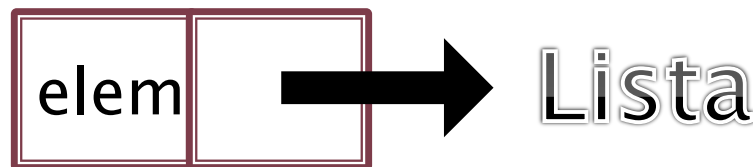


Implementação Recursiva de Listas Lineares

Algoritmos e Estrutura de Dados 1
Prof.: Luiz Gustavo Almeida Martins

Implementação Recursiva

- ▶ Uma lista pode ser definida recursivamente:



Implementação Recursiva

► Exemplos:

$L: \{ 1, 2, 3 \} = 1 -> \{ 2, 3 \}$

$L: \{ 2, 3 \} = 2 -> \{ 3 \}$

$L: \{ 3 \} = 3 -> \{ \}$

Implementação Recursiva

- ▶ As operações básicas de lista também podem ser redefinidas recursivamente.
 - Independe da forma de implementação utilizada.
 - Definição do passo recursivo e dos critérios de parada.
 - Código + simples e limpo.

Implementação Recursiva

▶ Vantagens:

- Algumas operações ficam bem mais simples se implementadas recursivamente.
- Em estruturas mais complexas, tais como árvores, as implementações recursivas são mais naturais.

▶ Desvantagem:

- Em geral, são **MENOS** eficientes em termos de esforço computacional.
 - Aumenta o número de chamadas de funções, as quais são operações relativamente “caras”.

Implementação Recursiva

- ▶ Operação insere ordenada recursiva:
 - Critérios de parada:
 - Final da Lista (cauda = Lista vazia).
 - Elemento **MENOR OU IGUAL** ao conteúdo do nó cabeça
 - Passo recursivo:
 - Inserir elemento no restante (cauda) da lista

inteiro `insere_ord` (endereço da lista, elem)

Critério de Parada

Início

SE Lista é vazia OU elem \leq 1º nó da Lista ENTÃO

 Alocar um NOVO nó;

 SE não foi possível alocar ENTÃO

 Retorna 0;

 FIM-SE

 Atribuir elem ao campo INFO do NOVO nó;

 Fazer NOVO nó apontar para o 1º nó da Lista;

 Fazer a Lista apontar para o NOVO nó;

 Retorna 1;

SENÃO

R = `insere_ord` (end. do campo PROX do 1º nó da Lista, elem);

 Retorna R;

FIM-SE

Passo Recursivo

Fim

Implementação Recursiva

▶ Exercício:

Como ficaria a operação **remove ordenada recursiva**?

- **Critérios de parada:**

- Final da lista (**cauda = Lista vazia**)
- Elemento **MENOR** que conteúdo do nó cabeça
- Elemento **IGUAL** ao conteúdo do nó cabeça

- **Passo recursivo:**

- Remover elemento do restante (cauda) da lista

inteiro **remove_ord** (endereço da lista, elem)

Início

