

Aula 10 - Listas Recursivas

Prof. Me. Claudiney R. Tinoco

profclaudineytinoco@gmail.com

Faculdade de Computação (FACOM)
Bacharelado em Ciência da Computação (BCC)
Bacharelado em Sistemas de Informação (BSI)

Algoritmos e Estruturas de Dados 1 (AED1)
GBC024 - GSI006



Implementação Recursiva

▶ Exemplos:

$L: \{ 1, 2, 3 \} = 1 \rightarrow \{ 2, 3 \}$

$L: \{ 2, 3 \} = 2 \rightarrow \{ 3 \}$

$L: \{ 3 \} = 3 \rightarrow \{ \}$



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”.



▶ 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

SE [Lista é vazia **OU** elem < 1º nó da Lista] **ENTÃO**

Retorna 0;

SENÃO

SE [elem = 1º nó da Lista] **ENTÃO**

Fazer AUX apontar para o 1º nó da lista (nó cabeça);

Fazer a Lista apontar para o sucessor de AUX;

Liberar a memória alocada para o nó apontado por AUX;

Retorna 1;

SENÃO

[R = **remove_ord** (end. do campo PROX do 1º nó, elem);

Retorna R;

FIM-SE

FIM-SE

Fim

Critérios de Parada

Passo Recursivo



Referências

✓ Básica

- CELES, W., CERQUEIRA, R. e RANGEL, J. L. "Introdução a estruturas de dados". Campus Elsevier, 2004.
- TENENBAUM, A. M., LANGSAM, Y. e AUGENSTEIN, M.J. "Estrutura de Dados Usando C". Makron Books.

✓ Extra

- BACKES, André. "Programação Descomplicada Linguagem C". Projeto de extensão que disponibiliza vídeo-aulas de C e Estruturas de Dados. Disponível em: <https://www.youtube.com/user/progdescomplicada>. Acessado em: 25/04/2022.

✓ Baseado nos materiais dos seguintes professores:

- Prof. André Backes (UFU)
- Prof. Bruno Travençolo (UFU)
- Prof. Luiz Gustavo de Almeida Martins (UFU)

Dúvidas?

Prof. Me. Claudiney R. Tinoco
profclaudineytinoco@gmail.com

Faculdade de Computação (FACOM)
Universidade Federal de Uberlândia (UFU)