

## AED1 - Lista 3

### Listas encadeadas

Seguem alguns exercícios relacionados com listas encadeadas.

1 - [4.3.5] - Uma lista é crescente se o conteúdo de cada célula não é maior que o conteúdo da célula seguinte. Escreva uma função que faça uma busca em uma lista crescente. Faça versões para listas com e sem cabeça. Faça uma versão recursiva e outra iterativa.

2 - [4.4.1] - Critique a seguinte variante da função Remove:

```
void Remove (celula *p, celula *lst) {
    celula *lixo;
    lixo = p->seg;
    if (lixo->seg == NULL) p->seg = NULL;
    else p->seg = lixo->seg;
    free(lixo);
}
```

3 - [4.5.1] - Por que a seguinte versão de Insere não funciona?

```
void Insere (int y, célula *p) {
    célula nova;
    nova.conteudo = y;
    nova.seg = p->seg;
    p->seg = &nova;
}
```

4 - [4.6.3] - Escreva uma função para remover de uma lista encadeada todos os elementos que contém x. Faça uma versão iterativa e uma recursiva.

5 - [4.7.1, 4.7.2] - Convertendo vetores e listas.

a) Escreva uma função que copie um vetor para uma lista encadeada.

b) Escreva uma função que copie uma lista encadeada para um vetor.

6 - [4.7.5] - Escreva uma função que concatene duas listas encadeadas (isto é, "pendure" a segunda no final da primeira).

7 - [4.7.7] - Escreva uma função que receba uma lista encadeada e devolva o endereço de uma célula que esteja o mais próximo possível do ponto médio da lista. Faça isso sem calcular explicitamente o número  $n$  de células da lista e o quociente  $n/2$ .

Para revisar conceitos sobre listas encadeadas e encontrar mais exercícios, acesse:

- <https://www.ime.usp.br/~pf/algoritmos/aulas/lista.html>