

Estruturas de Dados - Aula Extra 3

Listas Encadeadas

Cada elemento da lista é uma estrutura chamada **nó**, que além de guardar um valor, aponta para o próximo elemento da lista.

```
struct No {  
    tipo_valor      valor;  
    No              *prox;  
};
```

O primeiro elemento da lista é indicado por um ponteiro simples:

```
No * lista;
```

Podemos inserir um elemento em diferentes posições:

No início:

- a) Apontar o “prox” do novo elemento para o primeiro da lista;
- b) Apontar o “lista” para o novo elemento.

No fim:

- a) Apontar o “prox” do novo elemento da lista para “NULL”;
- b) Encontrar o último elemento da lista;
- c) Apontar o “prox” do último elemento para o novo elemento.

No meio:

- a) Procurar o elemento depois do qual se quer adicionar o novo elemento;
- b) Apontar o “prox” do novo elemento para o “prox” do elemento antigo;
- c) Apontar o “prox” do elemento antigo para o novo elemento.

Exercícios

- 1) Crie uma lista e uma função **insereInicio**, inserindo 10 elementos na lista.
- 2) Crie uma função que **imprima** todos os elementos da lista, para testar o exercício 1.
- 3) Crie uma função **insereFim** e a utilize no lugar de **insereInicio**. Veja a diferença no resultado.
- 4) Crie uma função **insereOrdenadoCrescente** e a utilize no lugar de **insereFim**. Veja a diferença no resultado.

Podemos remover um elemento de diferentes posições:

Do início:

- a) Criar um apontador “tmp” para o primeiro elemento da lista;
- b) Apontar o “lista” para o “prox” do primeiro elemento;
- c) Remover (delete) o elemento apontado por “tmp”.

Do fim:

- a) Encontrar o penúltimo elemento da lista;
- b) Remover (delete) o elemento apontado por “prox” do penúltimo;
- c) Apontar “prox” do penúltimo para NULL.

No meio:

- a) Procurar o elemento anterior ao que se deseja remover;
- b) Criar um apontador “tmp” para o elemento a ser removido (“prox” do anterior);
- c) Apontar o “prox” do anterior para o “prox” do elemento a ser removido;
- d) Remover (delete) o elemento apontado por “tmp”.

Exercícios

- 1) Crie uma função **removeInicio**, e modifique o programa anterior para remover o primeiro elemento.
- 2) Crie uma função **removeFim**, e modifique o programa anterior para remover também o último elemento.
- 3) Crie uma função **removeMeio**, e modifique o programa anterior para remover também o quarto elemento.