

LISTAS DUPLAMENTE ENCADEADAS

DCE792 - AEDs II (Prática)

Atualizado em: 11 de setembro de 2024

Iago Carvalho

Departamento de Ciência da Computação



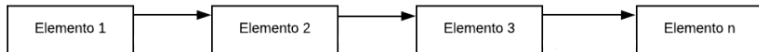
LISTAS OU VETORES

Estrutura de dados mais básicas existente em qualquer linguagem de programação

- Estrutura de dados linear
 - Seus elementos são salvos em pedaços contíguos de memória
 - Acesso pode ser realizado de forma aleatória (e rápida) utilizando índices
 - É possível guardar elementos homogêneos, i.e., do mesmo tipo
-
- | | |
|---------------------------------|---|
| ○ Acesso aleatório ↑ | ○ Tamanho fixo ↓ |
| ○ Fácil de ordenar elementos ↑ | ○ Difícil de inserir ou remover elementos ↓ |
| ○ Simples de trocar variáveis ↑ | ○ Desperdício de memória ↓ |

Uma lista implementada utilizando ponteiros

- Tamanho dinâmico
- Não possui desperdício de memória



O código completo de listas encadeadas está disponível no Github



► Link

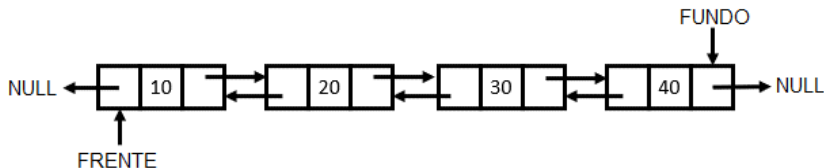
LISTAS DUPLAMENTE ENCADEADAS

Uma lista duplamente encadeada possui ponteiros "indo" e "vindo"

- Ponteiro *next*
- Ponteiro *prev*

Mais fácil navegar pela lista

- Facilita o uso de algoritmos de ordenação



LISTAS DUPLAMENTE ENCADEADAS

```
1 struct node {  
2     int data;  
3     struct node *next, *prev;  
4 };  
5  
6 struct lista_dupla {  
7     struct node *frente, *fundo;  
8 };
```

Struct **node**:

- *data*: valor armazenado
- *next* e *prev*: ponteiros para frente e para trás

Struct **lista_dupla**:

- *frente* e *fundo*: ponteiros para início e fim da lista

ATIVIDADE PRÁTICA

Modificar algum de seus códigos anteriores para implementar uma lista duplamente encadeada

- Incluir ponteiros *prev* e *next*
- Talvez o mais fácil seja trabalhar em cima do código da lista encadeada

Necessário alterar todos os métodos de inserção e remoção

PRÓXIMA AULA:

ÁRVORES BINÁRIAS (03/10)