

Listas

emanoelim@utfpr.edu.br

Listas

Uma lista é um tipo de estrutura de dados que guarda uma sequência de itens:



- x_1 é o primeiro item da lista;
- x_2 é o segundo item da lista;
- x_n é o último item da lista;
- n é o número de itens da lista.

Listas

Em uma lista dizemos que:

- x_i é o i -ésimo item da lista.
- x_i é o item que:
 - precede x_{i+1}
 - sucede x_{i-1}

Listas

- É uma das estruturas mais simples para guardar itens;
- É flexível: dependendo da implementação, seu tamanho pode aumentar ou diminuir durante a execução do programa. Por isso, listas são adequadas para aplicações nas quais não é possível prever a demanda por memória;
- Aplicações: cadastros em geral, armazenar dados que precisam estar ordenados (mais adiante veremos como ordenar os itens de uma lista).

TAD Lista

Um TAD Lista é formado pela estrutura que representa a lista e por um conjunto de operações para manipular esta estrutura. Geralmente, uma lista precisa das seguintes operações:

- Criar uma lista vazia;
- Inserir um item;
- Remover um item;
- Acessar um item;
- Verificar se a lista está vazia;
- Verificar se a lista está cheia;
- Imprimir a lista;
- Liberar a lista.

Tipos de implementação

Os métodos mais comuns de implementar listas são por meio de arranjos (vetores) ou por meio de ponteiros (listas encadeadas). Primeiramente estudaremos a implementação por arranjos ou vetores.

Implementação por arranjos

Seguindo a lógica de vetores, os itens da lista são armazenados em posições contíguas de memória. Considere, por exemplo, uma lista de 6 itens:

1000	X ₁
1001	X ₂
1002	X ₃
1003	X ₄
1004	X ₅
1005	X ₆

Implementação por arranjos

Ainda seguindo a ideia de vetores, é preciso definir um tamanho máximo para a lista e deve-se respeitar este tamanho.

Estrutura Lista

Um item de uma lista normalmente é uma estrutura que contém uma chave (um identificador para o item) e os campos desejados para o item:

```
typedef struct item Item;  
struct item {  
    int chave;  
    // demais campos desejados  
};
```

Estrutura Lista

Uma lista geralmente é representada por uma estrutura que contém um vetor de itens e um campo que guarda a quantidade de itens da lista (para depois poder percorrer os itens). Assim, temos:

Estrutura Lista

```
#define MAXTAM 100
typedef struct item Item;
typedef struct lista Lista;

struct item {
    int chave;
    // demais campos desejados
};

struct lista {
    Item item[MAXTAM];
    int qtde; // das 100 posições reservadas, quantas foram ocupadas?
};
```