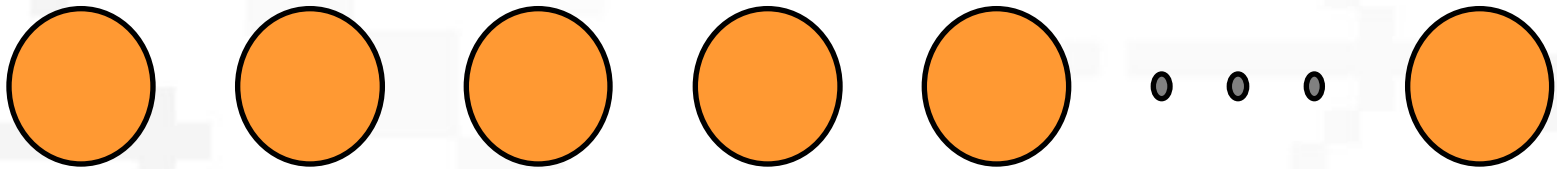


LISTAS LINEARES

Prof. José Luiz de Freitas Júnior

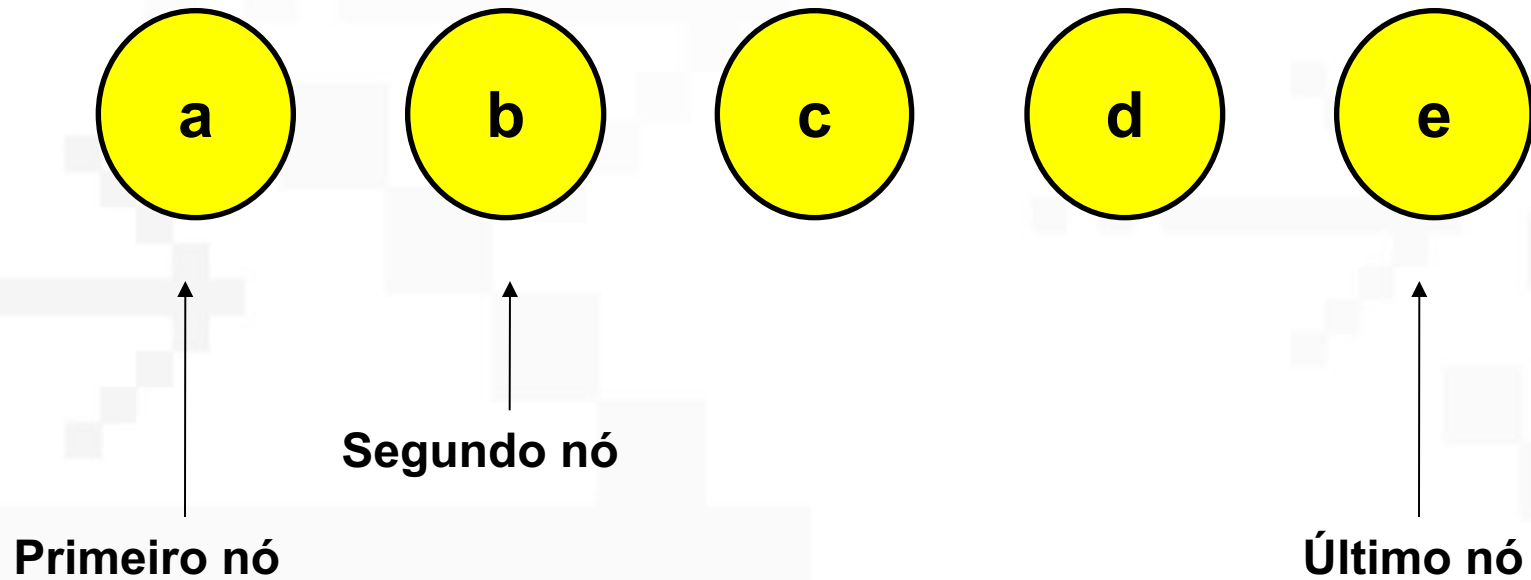
LISTA LINEAR

Uma Lista Linear (LL) é uma **sequência** de nós



- Nós - elementos do mesmo tipo
- Relação de ordem linear

LISTA LINEAR



Estrutura dos nós

- Estrutura interna é abstraída
- Enfatizado o conjunto de relações existente



INFORMAÇÕES				
Número	RG	Nome	Nasc.	Cargo

Exemplos de aplicações com listas

- Notas de alunos
- Cadastro de funcionários de uma empresa
- Itens em estoque em uma empresa
- Dias da semana
- Vagões de um trem
- Letras de uma palavra
- Pessoas esperando ônibus
- Cartas de baralho
- Precipitações pluviométricas em um mês / dia

Definição formal

Uma lista linear é uma coleção de $n \geq 0$ nós x_1, x_2, \dots, x_n , todos do mesmo tipo, cujas propriedades estruturais relevantes envolvem apenas as posições relativas lineares entre nós:

$n = 0$: lista vazia, apresenta zero nós

$n > 0$: x_1 é o primeiro nó

x_n é o último nó

$1 < k < n$: x_k é precedido por x_{k-1} e sucedido por x_{k+1}

- Lista linear : sequência de 0 ou mais nós do mesmo tipo

Operações sobre listas lineares

Exemplos de operações possíveis:

- Criar uma lista linear vazia;
- Inserir um novo item imediatamente após o i -ésimo item;
- Retirar o i -ésimo item;
- Localizar o i -ésimo item para examinar e/ou alterar o conteúdo de seus componentes;
- Combinar duas ou mais listas lineares em uma lista única;
- Partir uma lista linear em duas ou mais listas;
- Fazer uma cópia da lista linear;
- Ordenar os itens da lista em ordem ascendente ou descendente, de acordo com alguns de seus componentes;
- Pesquisar a ocorrência de um item com um valor particular em algum componente;
- Destruir um lista.

Implementação de Listas Lineares

Vamos estudar duas maneiras distintas

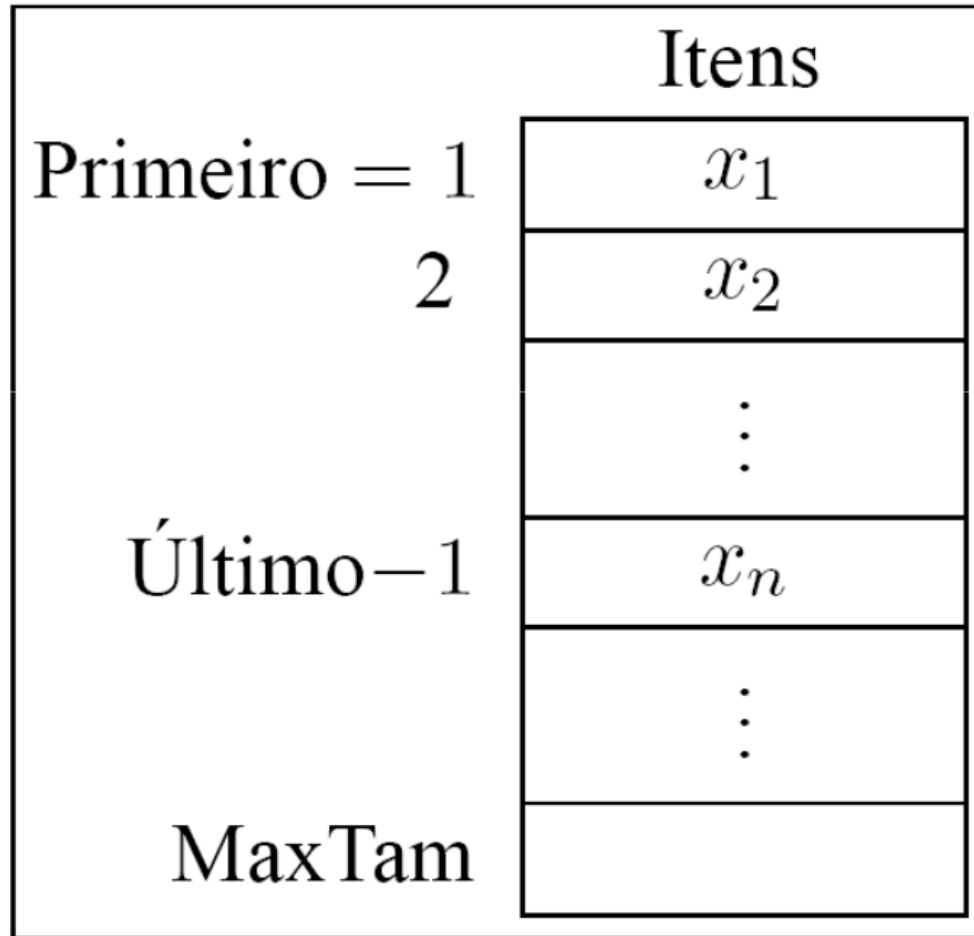
- Usando alocação sequencial e estática (com vetores).
- Usando alocação não sequencial e dinâmica (com ponteiros):

Estruturas Encadeadas.

Implementação com vetores

- Armazena itens em posições contíguas de memória.
- A lista pode ser percorrida em qualquer direção.
- A inserção de um novo item pode ser realizada após o último item com custo constante.
- A inserção de um novo item no meio da lista requer um deslocamento de todos os itens localizados após o ponto de inserção.
- Retirar um item do início da lista requer um deslocamento de itens para preencher o espaço deixado vazio.

Implementação com vetores



Implementação com vetores

- Os itens são armazenados em um **vetor** de tamanho suficiente para armazenar a lista.
- O campo **Último** contém a posição após o último elemento da lista.
- O i -ésimo item da lista está armazenado na i -ésima posição do vetor, $0 \leq i \leq \text{Último}$.
- A constante **MaxTam** define o tamanho máximo permitido para a lista.

Implementação com vetores

Desvantagens

- custo para inserir ou retirar itens da lista, que pode causar um deslocamento de todos os itens, no pior caso;
- em aplicações em que não existe previsão sobre o crescimento da lista, a utilização de arranjos em linguagens como o Pascal e o C++ pode ser problemática pois, neste caso, o tamanho máximo da lista tem de ser definido em tempo de compilação.

Exercício – Implementação de alguns métodos da lista

Lista
<ul style="list-style-type: none">- MaxTam : int {valor > 0}- array[] : int- ultimo : int {=-1}
<ul style="list-style-type: none">+ Lista(tamaho : int)+ ~ Lista()+ ListaVazia() : boolean+ listaCheia() : boolean+ listaTamanho() : int+ insereInicio(elemento : int) : void+ insereFim(elemento : int) : void+ inserePosicao(pos : int, elemento : int) : void+ removeInicio() : void+ removeFim() : void+ removePosicao(pos : int) : void+ removeElemento(elemento : int) : boolean+ getInicio() : int+ getFim() : int+ getPosicao(pos : int) : int+ buscaElemento(elemento : int) : int+ mostraLista() : void+ destroy() : void



EDELWEISS, Nina, GALANTE, Renata.
Estruturas de Dados.
Porto Alegre : Bookman, 2009.

Transparências do Livro