LISTAS

CONCEITO
LISTAS ENCADEADAS SIMPLES OU DUPLAS
ORDENADAS OU NÃO

Prof. Dr. Fabio Fernando Kobs

Listas - Tipo Abstrato de Dados - ADT

Listas é outro tipo abstrato de dados (ADT), ou seja, é uma especificação de um conjunto de dados.

Uma lista consiste de um conjunto de dados organizados em ordem linear, ou seja, uma coleção de itens sequenciais.

Os algoritmos que manipulam uma estrutura de dados do tipo listas fazem com que o conjunto de dados cresçam, encolham ou sofram alterações ao longo do tempo.

Uma lista armazena elementos dispostos um após o outro, como em uma lista de nomes, lista com dados de funcionários, lista de produtos etc.

Uma lista pode ser ordenada ou não.

ADT – Listas – Tipos

A estrutura lista pode ser representada em cinco tipos diferentes, e sugere-se que seja implementado de forma dinâmica (encadeada):

- **Lista simplesmente encadeada e não ordenada;**
- **Lista simplesmente encadeada e ordenada;**
- **Lista duplamente encadeada e não ordenada;**
- **Lista duplamente encadeada e ordenada;**
- **Listas circulares.**

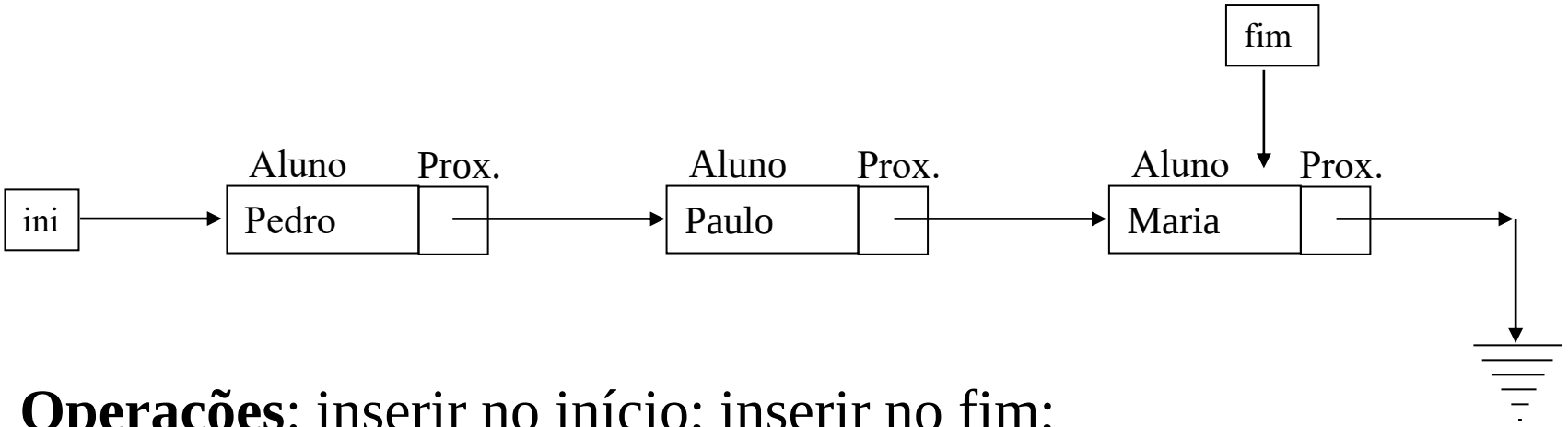
ADT - Listas

Implementação por lista encadeada (características):

- Tamanho da lista é dinâmico;
- Código reaproveitável das outras estruturas de dados;
- Inserção e remoção rápida de elementos na lista.

ADT - Lista Encadeada

Uma lista simplesmente encadeada não ordenada é formada por uma sequência de nós. Cada nó contém a informação relevante e uma referência indicando onde se encontra o próximo nó da lista.



Operações: inserir no início; inserir no fim;
consultar toda a lista; remover um elemento qualquer; esvaziá-la.

Ordenamento de uma Lista

Para certas aplicações será necessário manter os dados em ordem classificada na lista. Uma lista com essa característica é chamada de **lista ordenada**.

Em uma lista ordenada, os itens são organizados na ordem especificada (crescente ou decrescente).

A eliminação está geralmente limitada ao menor item (ou maior) na lista.

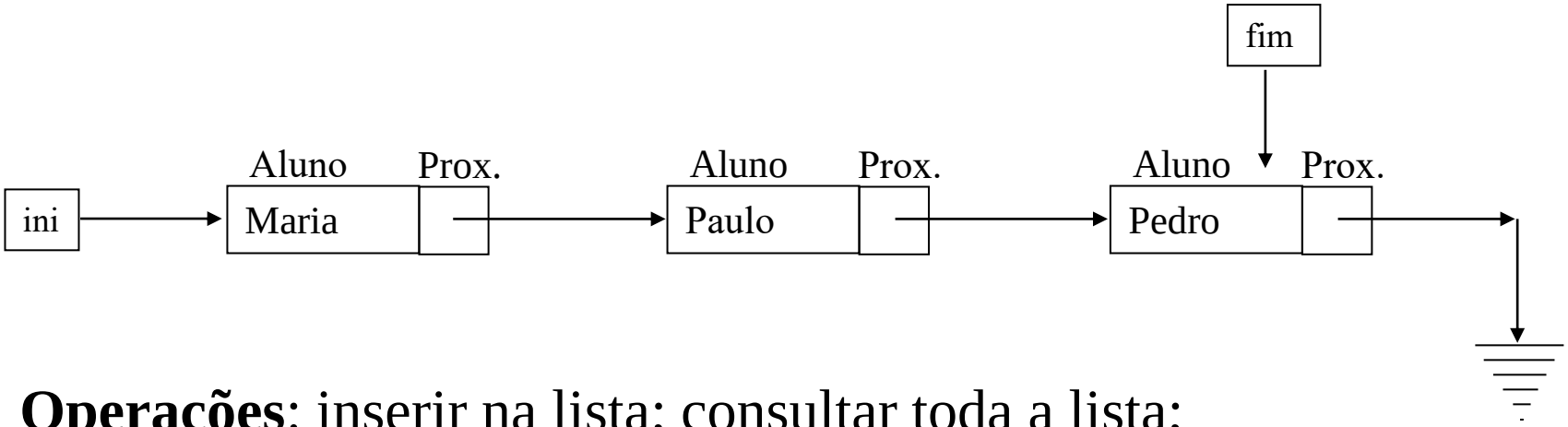
Ordenamento de uma Lista

Não há regras quanto à obrigatoriedade de inserção direta em uma lista ordenada.

Todavia, pode-se executar operações de ordenação antes de um processo específico, como por exemplo, listar os itens da lista de forma ordenada.

ADT - Lista Encadeada

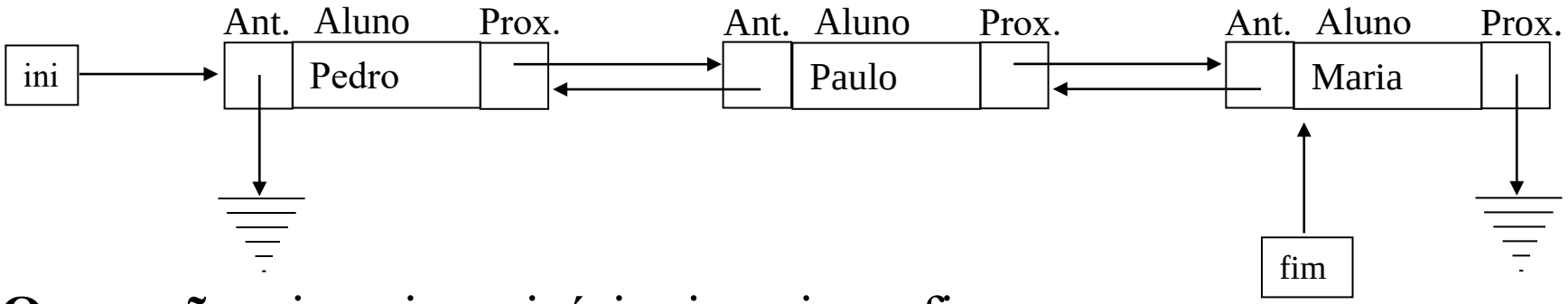
Uma lista simplesmente encadeada ordenada é formada por uma sequência de nós. Cada nó contém a informação relevante e uma referência indicando onde se encontra o próximo nó da lista.



Operações: inserir na lista; consultar toda a lista; remover um elemento qualquer; esvaziá-la.

ADT - Lista Encadeada

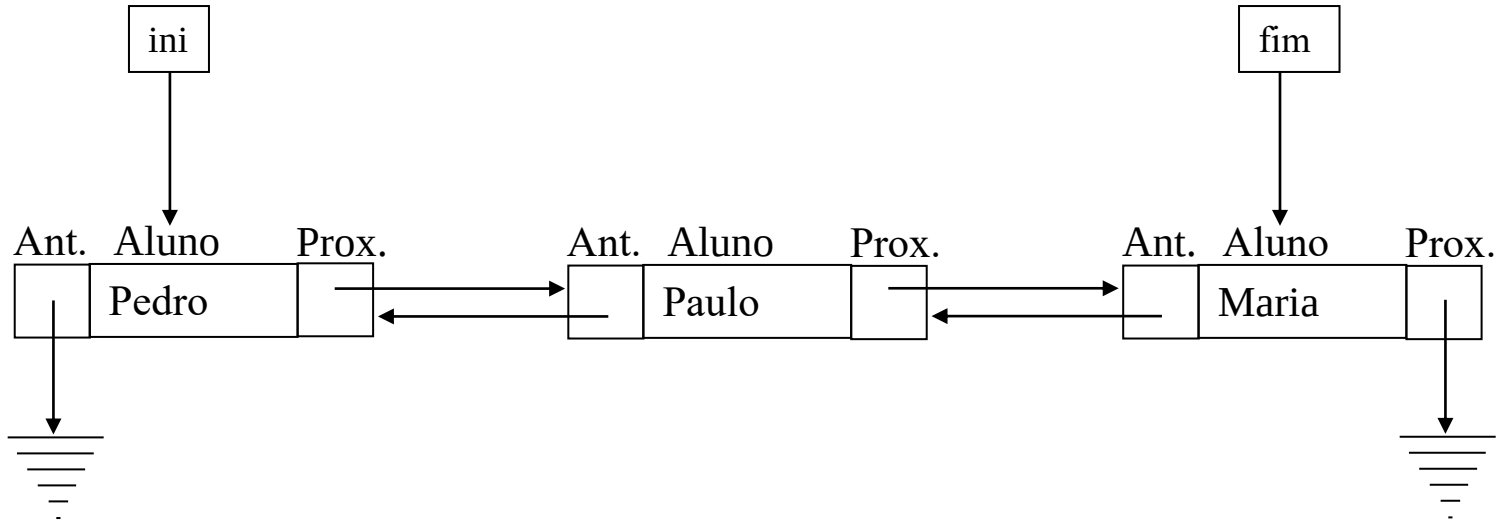
Uma lista duplamente encadeada e não ordenada é formada por uma sequência de nós. Cada nó contém a informação relevante e duas referências. A primeira indica onde se encontra o nó anterior da lista, e a segunda indica onde se encontra o próximo nó da lista.



Operações: inserir no início; inserir no fim;
consultar toda a lista do início ao fim ou do fim ao início;
remover um elemento qualquer; esvaziá-la.

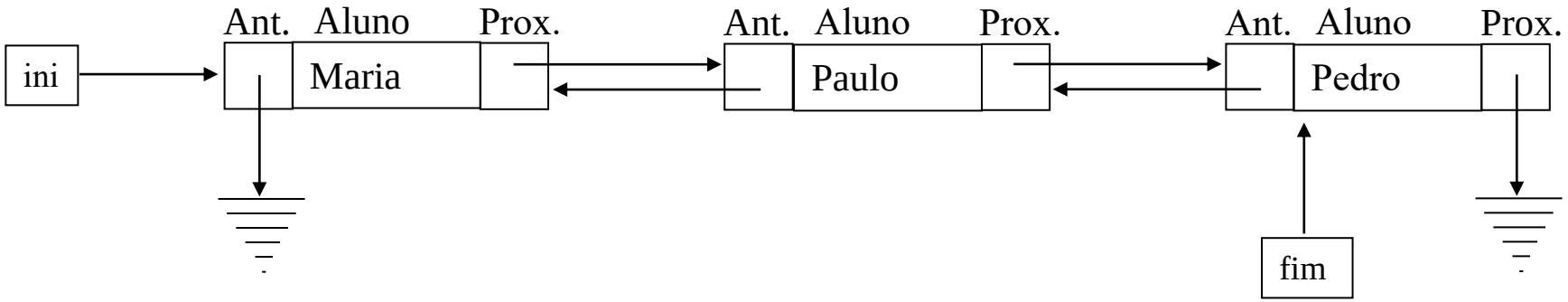
ADT - Lista Encadeada

Os ponteiros (ou referências) que indicam o início e o fim numa lista encadeada são chamados de sentinela ou lista com extremidade dupla.



ADT - Lista Encadeada

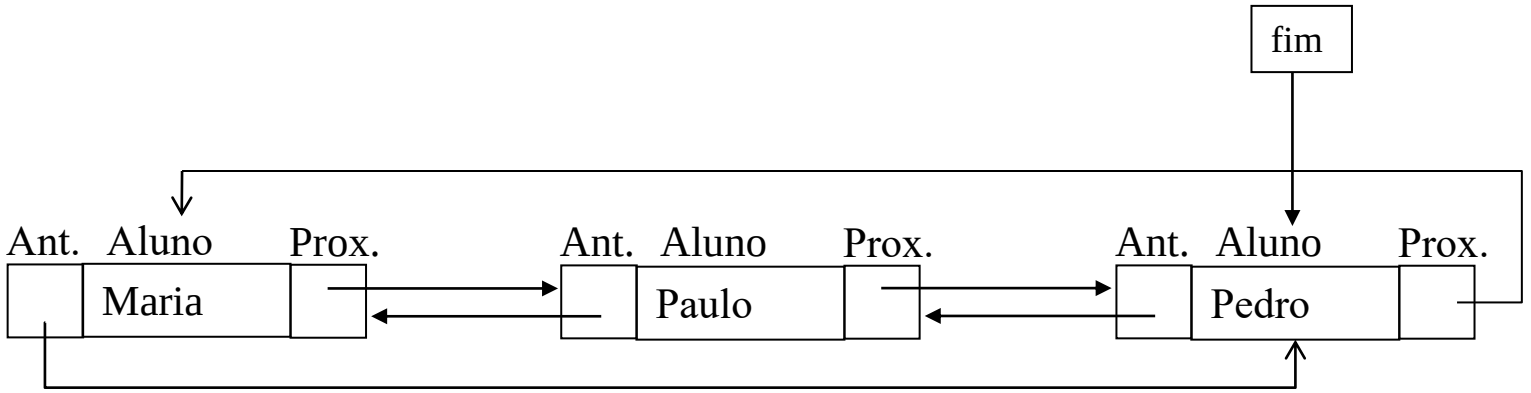
Uma lista duplamente encadeada e ordenada é formada por uma sequência de nós. Cada nó contém a informação relevante e duas referências. A primeira indica onde se encontra o nó anterior da lista, e a segunda indica onde se encontra o próximo nó da lista.



Operações: inserir na lista; consultar toda a lista do início ao fim ou do fim ao início; remover um elemento qualquer; esvaziá-la.

ADT – Listas Circulares

No caso de uma lista duplamente encadeada circular e ordenada (as características da circular, equivalem as vistas para a estrutura de dado fila), a referência final terá como referência o primeiro nó. E no primeiro nó, a referência do anterior receberá o último nó da lista.



Também pode-se implementar listas circulares para listas simplesmente encadeadas, sejam ordenadas ou não.

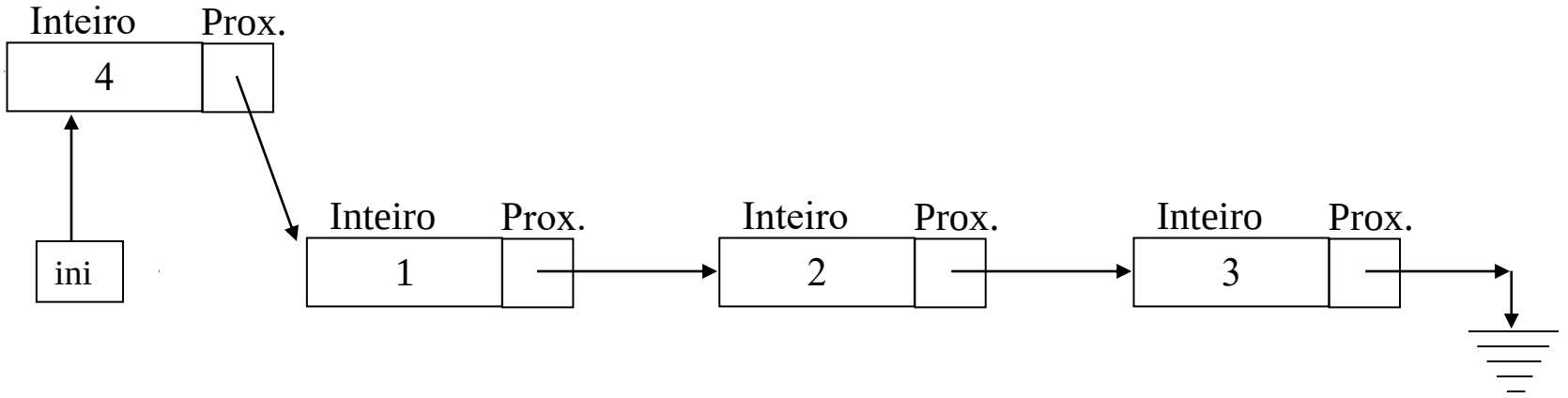
Análise: INSERÇÃO NO INÍCIO DA LISTA

É necessário apenas realizar operações de atribuições atualizando a referência do início (e fim da lista, quando necessário).

Para implementação nas listas:

- Lista simplesmente encadeada e não ordenada;
- Lista duplamente encadeada e não ordenada;
- Lista circular simplesmente encadeada e não ordenada;
- Lista circular duplamente encadeada e não ordenada.

INSERÇÃO NO INÍCIO DA LISTA



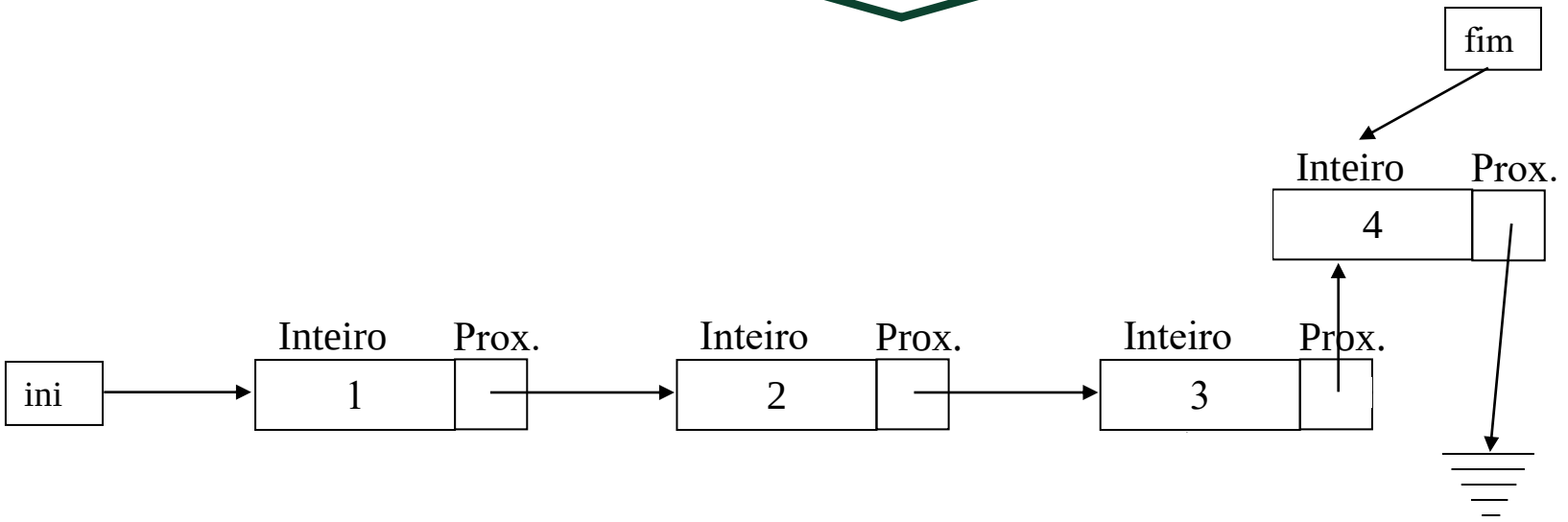
Análise: INSERÇÃO NO FIM DA LISTA

É necessário apenas realizar operações de atribuições atualizando a referência do fim (e início da lista, quando necessário).

Para implementação nas listas:

- Lista simplesmente encadeada e não ordenada;
- Lista duplamente encadeada e não ordenada;
- Lista circular simplesmente encadeada e não ordenada;
- Lista circular duplamente encadeada e não ordenada.

INSERÇÃO NO FIM DA LISTA



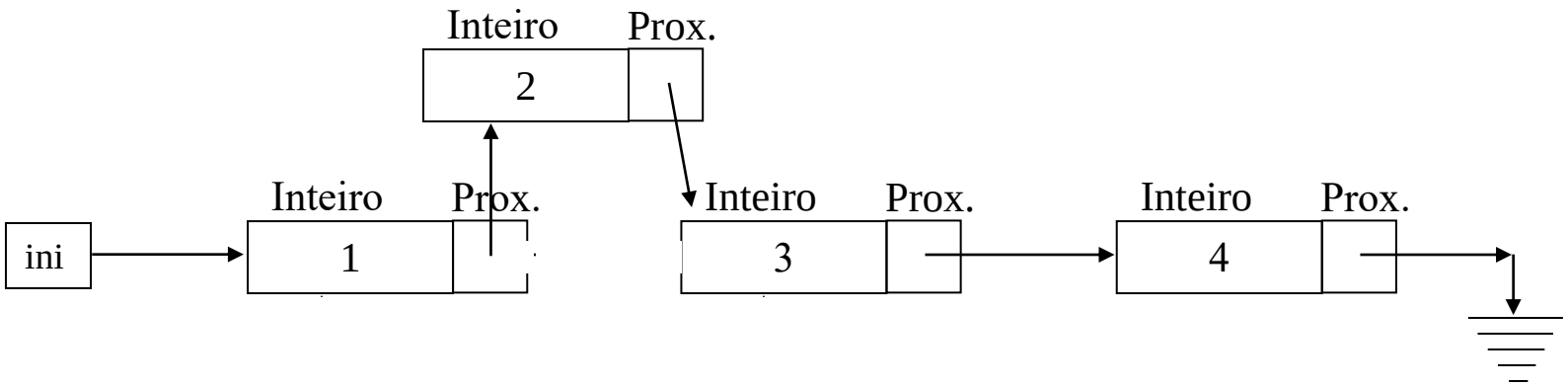
Análise: INSERÇÃO ORDENADA

O elemento a ser inserido pode ser, no pior caso, o maior de todos os já existentes na lista. Com isso, percorre-se toda a lista, comparando o novo elemento com os demais, para encontrar a posição correta e inseri-lo.

Para implementação nas listas:

- Lista simplesmente encadeada e ordenada;
- Lista duplamente encadeada e ordenada;
- Lista circular simplesmente encadeada e ordenada;
- Lista circular duplamente encadeada e ordenada.

INSERÇÃO ORDENADA

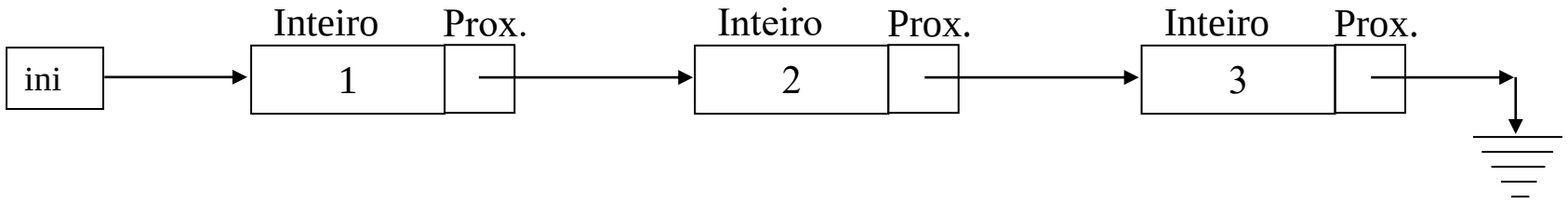


Análise: CONSULTAR TODA A LISTA

Em todos os tipos de listas, para realizar a operação para consultar todos os itens da lista, se faz necessário acessar cada elemento.

No caso da lista duplamente encadeada, ordenada ou não, é possível percorrer a lista do início ao fim ou vice-versa.

CONSULTAR TODA A LISTA

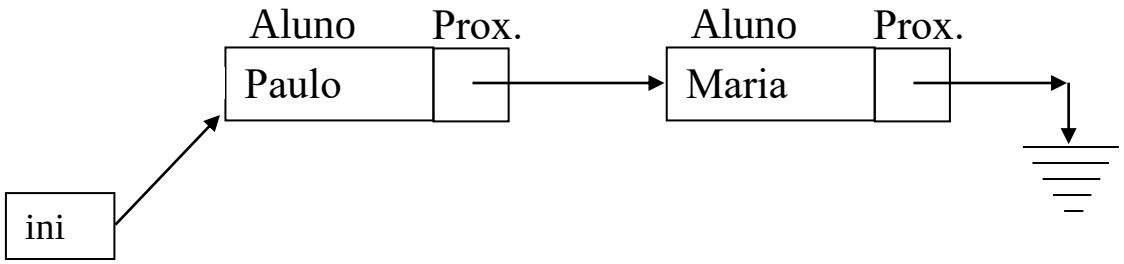


Análise: REMOÇÃO

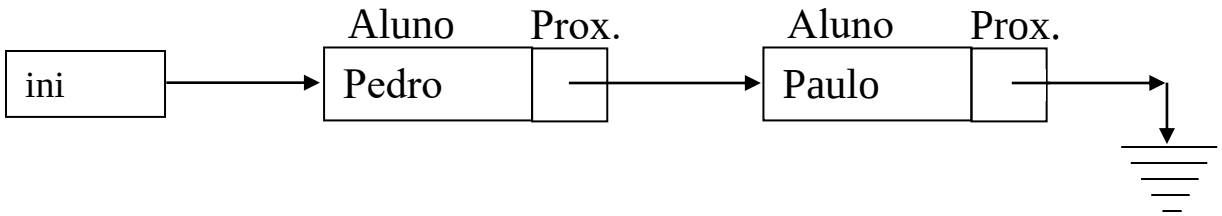
A remoção de qualquer tipo de lista, consiste em procurá-lo e depois acertar as referências para que a lista continue sendo acessada após a remoção.

Na busca pelo elemento a ser removido, percorre-se, no pior caso, todos os elementos da lista.

REMOÇÃO NO INÍCIO



REMOÇÃO NO FIM



REMOÇÃO NO MEIO

Análise: ESVAZIAR A LISTA

A operação de esvaziamento da lista consiste em remover todos os elementos dela.

O tempo gasto nessa operação depende da linguagem de programação que está sendo utilizada.

No caso da linguagem PYTHON, não é necessário realizar a remoção de cada um dos elementos, apenas atualizar a referência início (e ou fim) da lista para *None*, e as memórias alocadas serão liberadas por um procedimento de coleta de lixo da linguagem.

Exercício 01

Construa um programa para manipular inteiros em uma lista simplesmente encadeada e não ordenada:

1 – Inserir no final da lista
2 – Inserir no início da lista
3 – Excluir um elemento
4 – Consultar a lista
5 – Esvaziar a lista

Exercício 02

Construa um programa para manipular inteiros em uma lista simplesmente encadeada e ordenada:

1 – Inserir um inteiro na lista
2 – Excluir um inteiro na lista
3 – Consultar a lista
4 – Esvaziar a lista

Exercício 03

Construa um programa para manipular inteiros em uma lista duplamente encadeada e não ordenada:

1 – Inserir no final da lista
2 – Inserir no início da lista
3 – Excluir um elemento
4 – Consultar a lista
5 – Esvaziar a lista

Exercício 04

Construa um programa para manipular inteiros em uma lista duplamente encadeada e ordenada:

1 – Inserir um inteiro na lista
2 – Excluir um inteiro na lista
3 – Consultar a lista
4 – Esvaziar a lista

Exercício 05

Refaça os exercícios 01, 02, 03 e 04, mas agora para listas encadeadas circulares.

Exercício 06

Construa um programa que, a partir de lista encadeada L desordenada, tenha uma função que cria uma nova lista K ordenada, **com os mesmos nós da lista L**.
A função remove os elementos da lista L, e insere-os na lista K, que dessa forma torna-se uma lista ordenada (em ordem crescente).

Observações:

- Não devem ser criados nós extras, deve-se utilizar os mesmos nós alocados para a lista L.
- No final do processo, a lista L estará vazia e a lista K conterá os nós anteriormente alocados para a lista L.

Exercício 07

Construa um programa com uma função para concatenar duas listas encadeadas L1 e L2 em uma única lista encadeada ordenada. Depois, inclua uma função para remover os elementos repetidos da lista encadeada ordenada.

Não devem ser alocados (criados) novos nós. Os nós serão religados para compor a nova lista ordenada.

Exercício 08

Fazer um programa para criar um gerenciamento automático de **estoque de alimentos**.
O sistema permitirá incluir alimentos em uma lista. A lista organizará os alimentos por tipo/nome seguido da data de validade.

Nome Qtd D.Validade
Couve 1 05/12/2021
Couve 3 08/12/2021
Tomate 3 04/12/2021
Tomate 5 05/12/2021
Tomate 5 07/12/2021
Batata 10 01/01/2022
Orégano 50 18/12/2022
Ervilha 100 12/12/2021

As opções de menu permitem:
1) Incluir novo alimento comprado.
2) Consumir/remover alimento (todo ou apenas reduzir uma quantidade) – sempre o com validade mais próxima.
3) Verificar itens disponíveis (agrupados por tipo – couve = 4 , ou separados por lotes conforme cadastro – couve 1 05/12 couve 3 8/12).
4) Conferir quantidade mínima cadastrada (definição do estoque de emergência).
5) Alterar quantidade mínima de determinado item.

Exercício 08

Observações:

Ao escolher consumir um alimento:

- Verificar quantidade desejada do alimento e comparar com a quantidade disponível.
- Se disponível, reduzir quantidade do estoque. Iniciando sempre pelos itens com menor prazo de validade.
- Se após a baixa, o item entrar no estoque de emergência, notificar a necessidade de nova compra.
    - Uma lista a parte conterá os tipos/nomes dos alimentos e quantidade mínima.
    - Sempre que um novo produto for adicionado na lista, deverá ser solicitado a quantidade mínima. Arroz, por exemplo."""

Exercício 09 – BÔNUS

Examine a seguinte variação do problema de Josephus. Um grupo de pessoas faz um círculo e cada uma escolhe um inteiro positivo. Um de seus nomes e um inteiro positivo n são escolhidos. Começando com a pessoa cujo nome é escolhido, elas serão contadas ao redor do círculo, no sentido horário, e a enésima pessoa será eliminada. O inteiro positivo que essa pessoa escolheu será, então, usado para continuar a contagem. Toda vez que uma pessoa for eliminada, o número que ela escolheu será usado para determinar a próxima pessoa eliminada. Por exemplo, suponha que as cinco pessoas sejam A, B, C, D e E, que elas tenham escolhido os números 3, 4, 6, 2 e 7, respectivamente, e que o inteiro 2 seja inicialmente escolhido. Sendo assim, se começarmos a partir de A, a sequência na qual as pessoas serão eliminadas do círculo será B, A, E, C, deixando D como o último no círculo.
Escreva um programa que leia um grupo de linhas de entrada. Cada linha de entrada, exceto a primeira e a última, contém um nome e um inteiro positivo escolhidos por essa pessoa. A ordem dos nomes nos dados é a sequência em sentido horário das pessoas no círculo, e a contagem deve começar com o primeiro nome na entrada. A primeira linha de entrada contém o número de pessoas no círculo. A última linha de entrada contém somente um único inteiro positivo representando a contagem inicial. O programa imprime a sequência na qual as pessoas são eliminadas do círculo.