

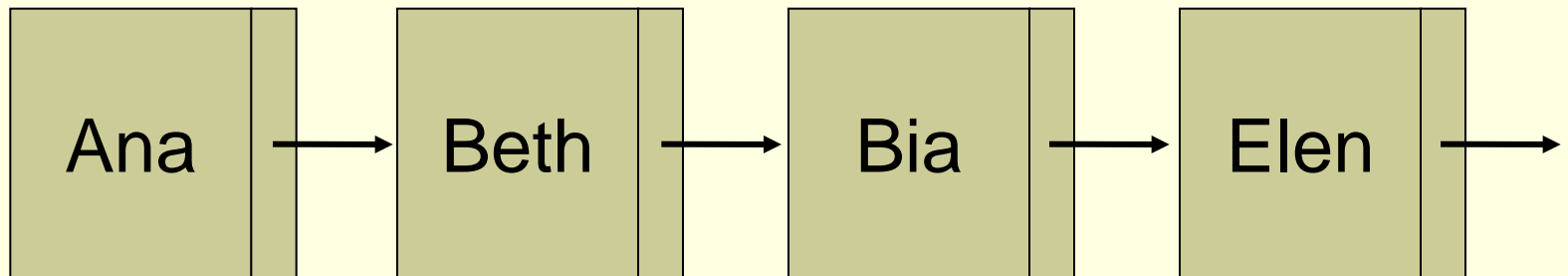
Listas ordenadas

SCC-502 – Algoritmos e Estruturas de
Dados I

Listas ordenadas

■ Definição

- *Listas em que os elementos estão ordenados por algum critério*
 - Em geral, por ordem alfabética



Listas ordenadas

■ Situações

- Cadastro de funcionários por ordem alfabética
- Lista de passageiros em um voo
- Lista de deputados presentes no congresso
- Telefones da cidade

Listas ordenadas

- Nessas três **situações os acessos** (inserção, eliminação e consulta) são **direcionados a um elemento** específico, e não mais ao “primeiro” (ou último) a entrar na lista (fila ou pilha). Exemplos:
 - “Carla Peres” foi despedida (retire seu nome do cadastro)
 - “Edmundo” é funcionário? Verifique se seu nome consta do cadastro
 - Qual o salário do funcionário “Pedro Malan”?
 - “Sandra Bulloc” foi contratada; inclua-a no cadastro
- Nos exemplos acima, os registros (as “pastas”) de cada funcionário são (ordenados e) procurados pelo nome. O nome, neste caso, é a chave de busca, ou simplesmente **chave**.

Listas ordenadas

■ Operações

- cria(lista)
- IsEmpty(lista)
- IsFull(lista)
- esta_na_lista(lista,x)
- inserir(lista,x)
- remover(lista,x)
- imprimir_todos_da_lista(lista)
- Etc.

Listas ordenadas

- Implementações

- Sequencial
- Encadeada

- Implicam mudanças na forma de manipulação da lista

Lista ordenada sequencial

- O que acontece se quero incluir na lista a funcionária “Alice do País das Maravilhas”?

...	...
Zorro	$i+1$
Zoroastro	i
Zico	$i-1$
...	...
Antônio	3
André	2
Ana	1

Lista ordenada sequencial

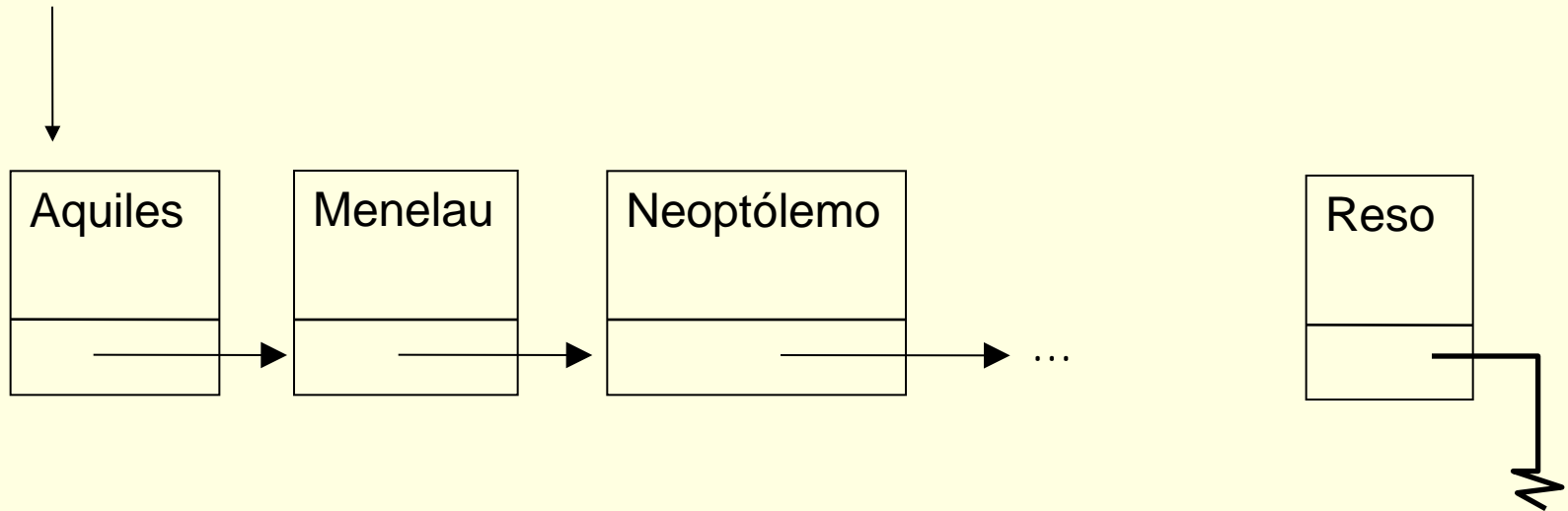
- O que acontece se quero incluir na lista a funcionária “Alice do País das Maravilhas”?

Tem que se deslocar todos os elementos da lista para inserir o novo elemento na posição correta

...	...
Zorro	$i+1$
Zoroastro	i
Zico	$i-1$
...	...
Antônio	3
André	2
Ana	1

Lista ordenada encadeada

- Lista de Gregos, Troianos e Tantáidas



- O que deve acontecer na lista quando
 - Nasce Filoctetes?
 - Nasce Agamêmnon?
 - Morre Menelau?

Lista ordenada encadeada

- Vantagem sobre a alocação sequencial
 - Os nós podem ser inseridos e eliminados na posição correta, sem realocação dos demais elementos
- **Complexidade de algoritmos**
 - O que implica?
 - A diferença é significativa?

Lista ordenada estática e encadeada

■ Declaração da estrutura

```
#define TAM ...
```

```
typedef struct bloco {  
    char nome[20];  
    int prox;  
} no
```

```
typedef struct {  
    int ini, primeiro_vazio;  
    no v[TAM];  
} ListaOrd;
```

Lista ordenada dinâmica e encadeada

■ Declaração da estrutura

```
typedef struct bloco {  
    char nome[20];  
    struct bloco *prox;  
} no
```

```
typedef struct {  
    no *ini;  
} ListaOrd;
```

Lista ordenada dinâmica e encadeada

- Exercício: identificar os possíveis casos para a **inserção** (caso 1, caso 2, ...)
- Supondo que o elemento a ser inserido não está na lista

Inserção: caso 1

- `insere(L,x,erro)`

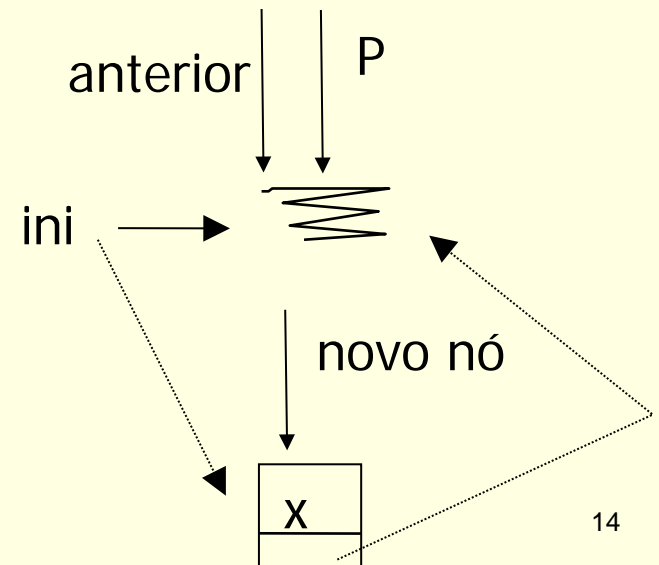
- **Lista vazia**

- se possível, aloca novo nó e `erro` \leftarrow `false`

- `nó` \leftarrow `x`

- `próximo de nó` \leftarrow `nada`

- `ini` \leftarrow `nó`



Inserção: caso 2

- `insere(L,x,erro)`

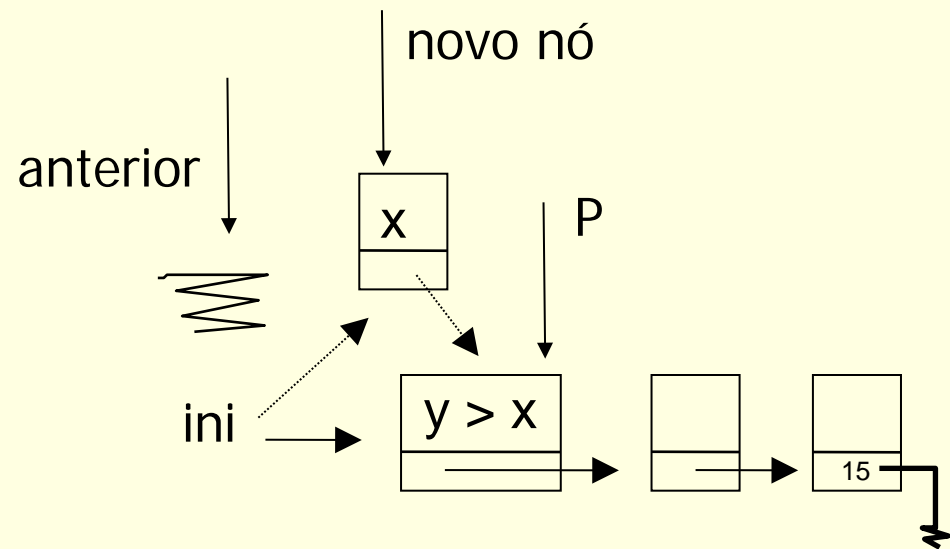
- $x < 1^{\circ}$ da lista

se possível, aloca novo nó e $\text{erro} \leftarrow \text{false}$

$\text{nó} \leftarrow x$

$\text{próximo de nó} \leftarrow P$

$\text{ini} \leftarrow \text{nó}$



Inserção: caso 3

- `insere(L,x,erro)`

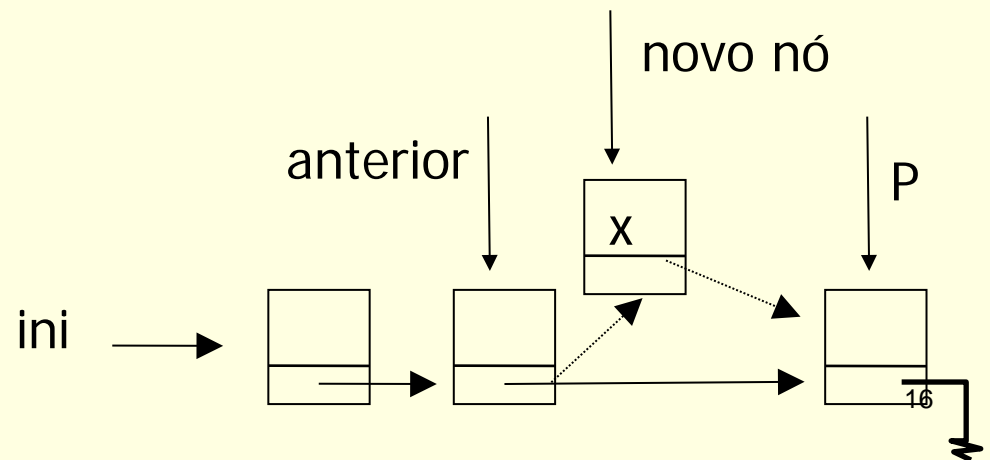
- `x > 1º da lista`

- se possível, aloca novo nó e `erro ← false`

- `nó ← x`

- `próximo de nó ← P`

- `próximo de anterior ← nó`



Inserção

- Exercício: implementar a função de inserção

Questão

- O que é melhor: inserir em uma desordenada ou em uma lista ordenada?

Remoção

- Quais seriam os casos?