Algoritmos e Estrutura de Dados II

Aula 05

Vetores Ordenados

Busca Binária

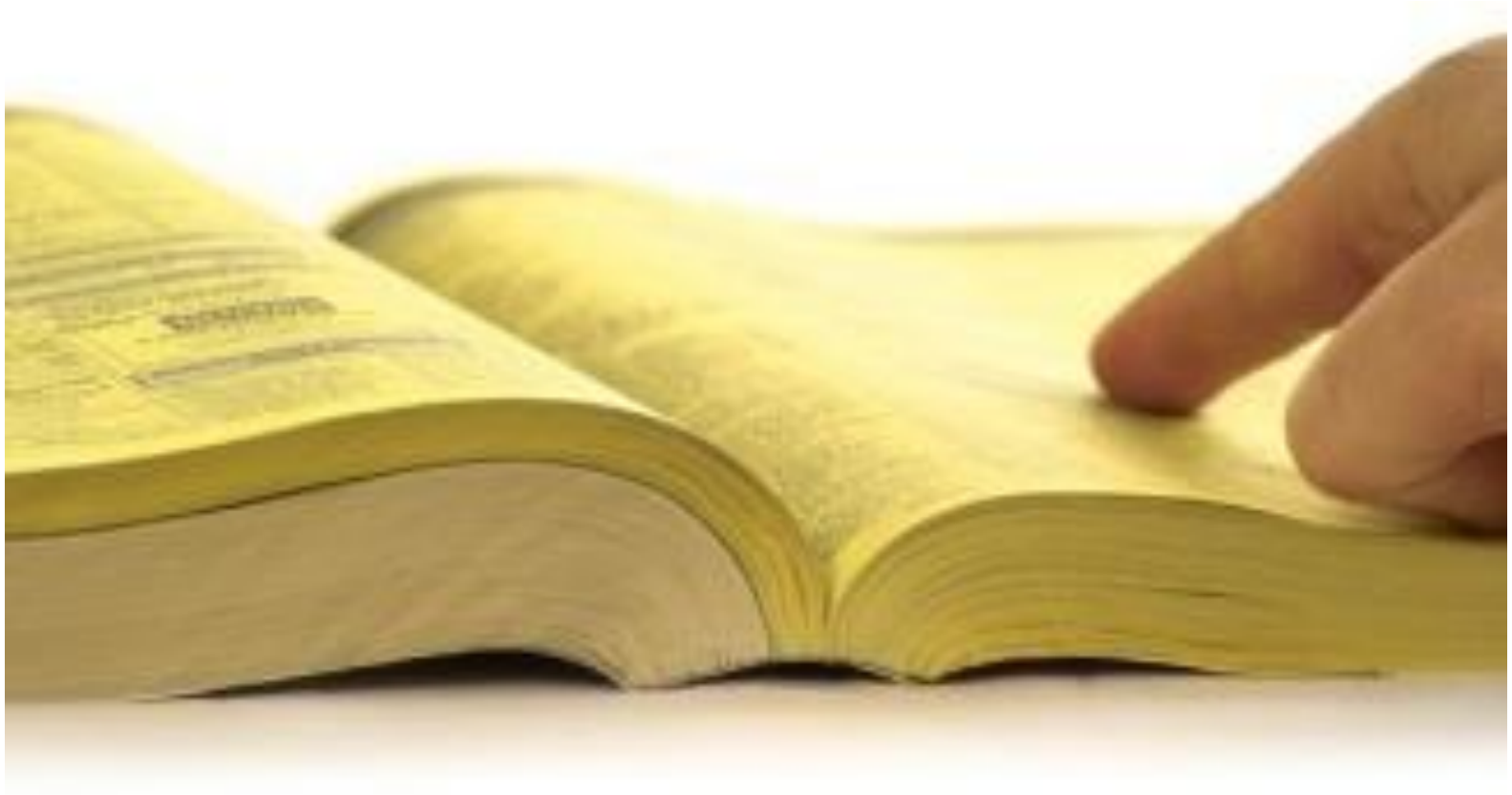
Prof. Dr. Dilermando Piva Jr

2º Semestre - CDN

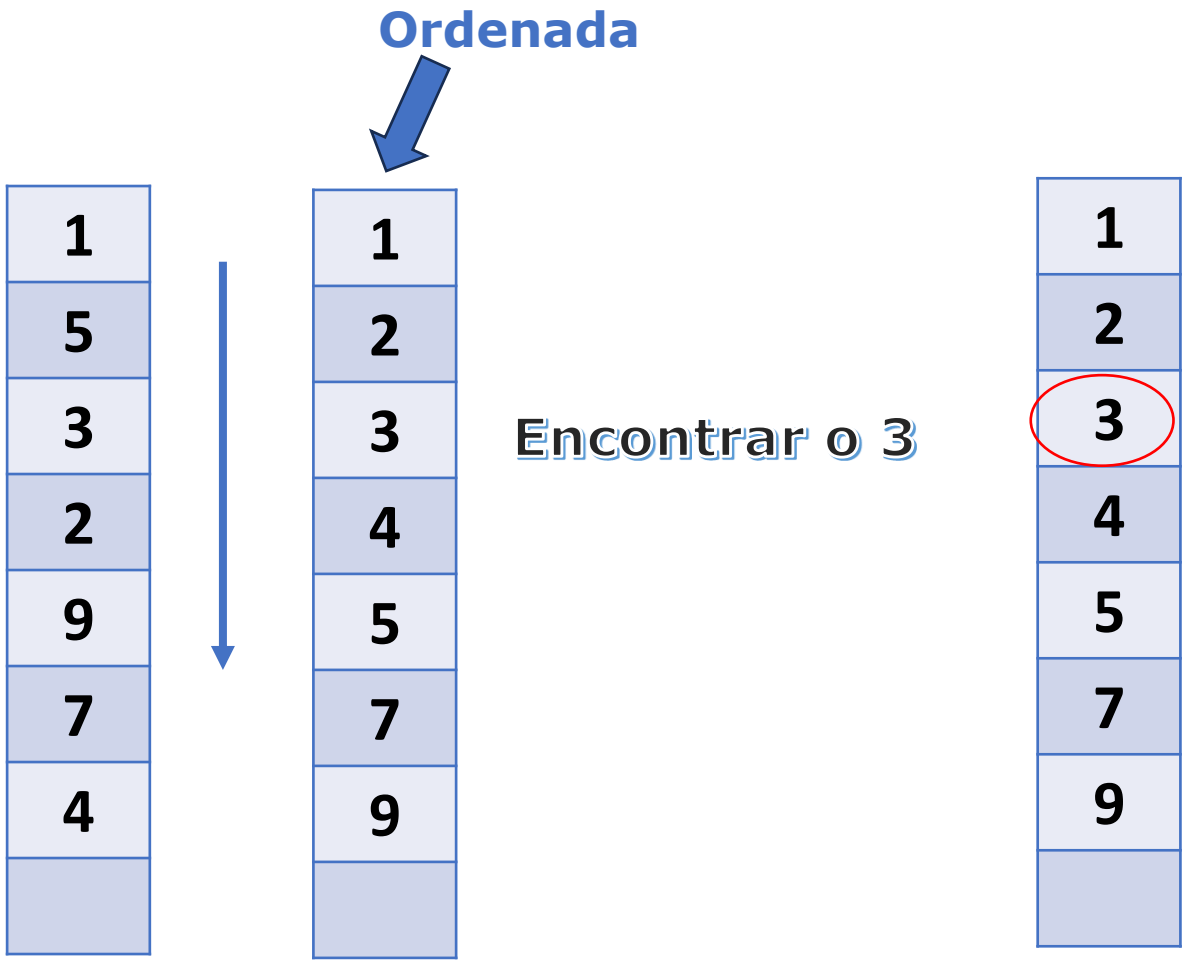
Algoritmos de Pesquisa

Estruturas ordenadas - Busca binária

Busca em uma lista telefônica...



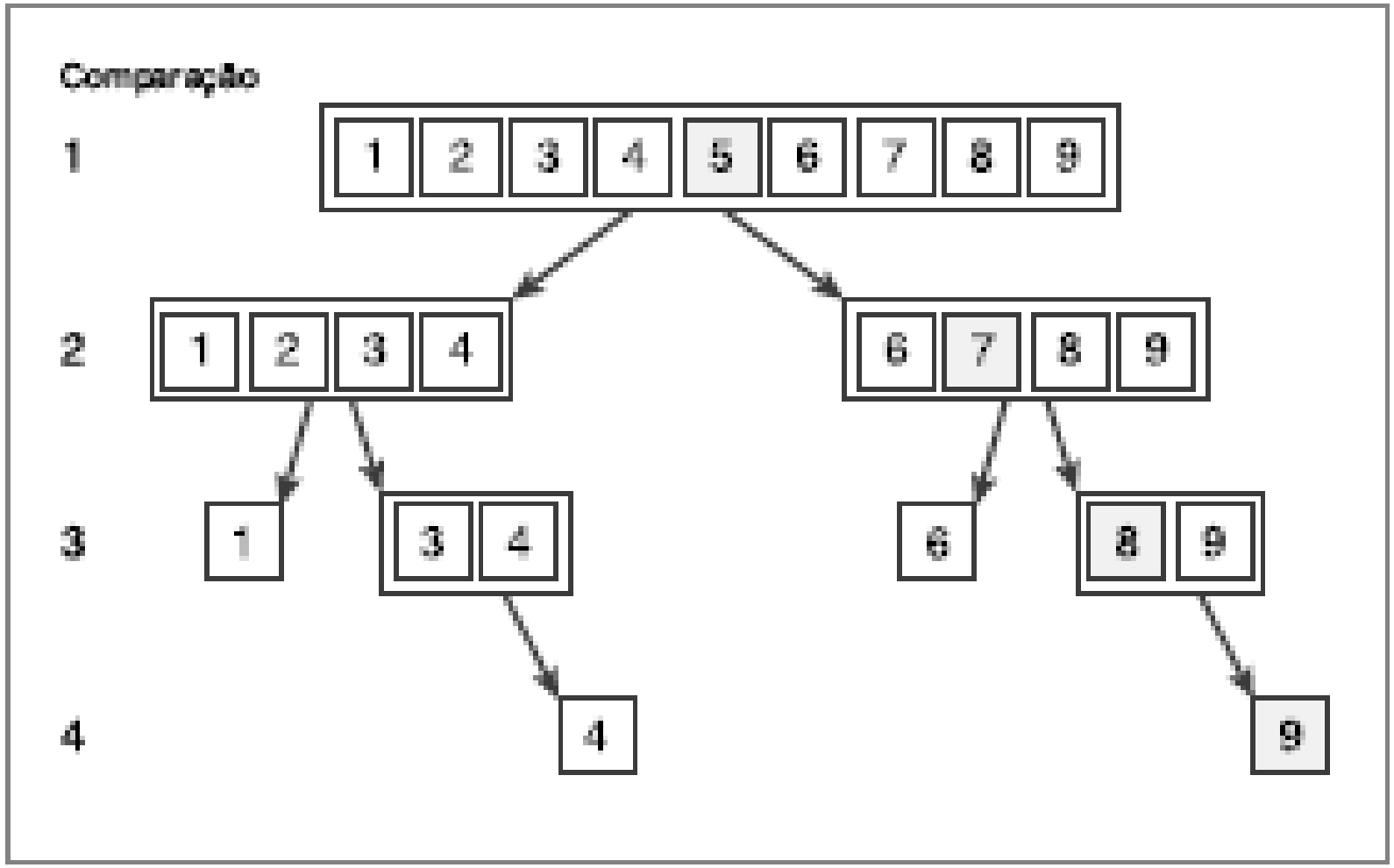
Busca em uma Lista Ordenada



- **Problema:** buscar um valor v não existente em uma lista sequencial ordenada com N elementos
  - o tempo de busca pode ser reduzido significantemente
    - devido a relação entre os elementos da lista
    - mesmo para um conjunto de elementos é grande

Busca Binária em uma Lista Ordenada

- **Problema:** buscar um valor v não existente em uma lista sequencial ordenada com N elementos
  - o tempo de busca pode ser reduzido significantemente
    - devido a relação entre os elementos da lista
    - mesmo para um conjunto de elementos é grande



Pesquisa binária em uma lista ordenada (1 de 2)

Código para a função de pesquisa binária:

```
def busca_binaria(v, lista_ord):
    esquerda = 0
    direita = len(lista_ord) - 1
    while esquerda ≤ direita:
        meio = (esquerda + direita) // 2
        if v == lista_ord[meio]:
            return meio
        elif v < lista_ord[meio]:
            direita = meio - 1
        else:
            esquerda = meio + 1
    return -1
```

Busca Binária... Complexidade

- Número de passos:
  - suponha que n = 2k sem perda de generalidade
    - 1o passo: uma lista de **n** elementos = $n/2^0$ elementos
    - 2o passo: uma lista de **n/2** elementos = $n/2^1$ elementos
    - 3o passo: uma lista de **n/4** elementos = $n/2^2$ elementos
    - ..... k-ésimo passo: uma lista de $\mathbf{n/2^{(k-1)}}$ elementos
    - (k+1)-ésimo passo: uma lista de $\mathbf{n/2^k}$ elementos
  - sendo o último passo, resta na lista somente 1 elemento
  - Logo:
    - $n/2^k = 1 \rightarrow k = \mathbf{\log_2 n}$

Pesquisa binária em uma lista ordenada (1 de 2)

Código para a função de pesquisa binária:

```
def busca_binaria(v, lista_ord):
    direita = 0
    esquerda = len(lista_ord) - 1
    while esquerda ≤ direita:
        meio = (esquerda + direita) // 2
        if v == lista_ord[meio]:
            return meio
        elif v < lista_ord[meio]:
            direita = meio - 1
        else:
            esquerda = meio + 1
    return -1
```

Complexidade?

O (log n)

VAMOS PARA A PRÁTICA ?!!!