

# ESTRUTURA DE DADOS

Prof.<sup>a</sup> Priscilla Abreu

[priscilla.braz@rj.senac.br](mailto:priscilla.braz@rj.senac.br)



# Estrutura de dados



## Roteiro de Aula

- Objetivo da aula
- Métodos de busca
  - Busca sequencial
  - Busca binária
- Exercícios

# Estrutura de dados



## Objetivo da aula

Conhecer e aplicar os métodos de busca de elementos em um vetor.



## **Indicador:**

Elaborar sistemas implementando métodos de ordenação e pesquisa.

# BUSCA

# Estrutura de dados



## Introdução

Pense em sistemas que costumamos acessar, tais como sistemas de compra (online ou lojas físicas), sistemas de agendamento de serviços, Redes Sociais, entre outros.

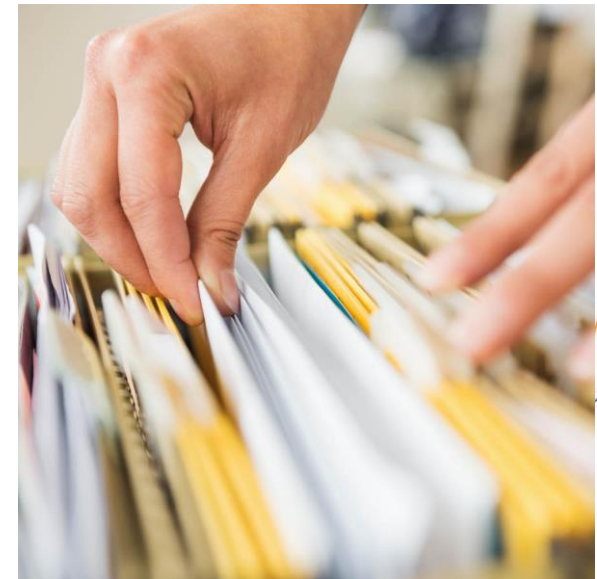
**Quais são as principais operações em sistemas que geralmente utilizamos?**

# Estrutura de dados



## Introdução

Uma importante operação é a **busca**. Todos fazemos buscas, consciente ou inconscientemente, ao longo do dia.



# Estrutura de dados



## Introdução

Grande parte dos sistemas computacionais utilizam operações de busca.

The screenshot shows the Saraiva website interface. The search bar contains 'estrutura de dados'. Below the search bar, there are search results for 'Estrutura de Dados e Algoritmos Em C++ - Tradução da 4ª...' and 'Estrutura de Dados Descomplicada - Em Linguagem C'. The website header includes the Saraiva logo and navigation links.

The screenshot shows a 'Consulta Produtos' window. It contains a table of products with columns: Código, Descrição, Referência, Und, Custo, and Marca. The table lists several products, including 'BOBINA BFF 0.90MM', 'BOBINA BFF 1.50MM', and 'BOBINA BFF 1.80MM'. Below the table, there is a section for 'Estoque' (Stock) and 'Preços' (Prices).

Código	Descrição	Referência	Und	Custo	Marca
603	BOBINA BFF 0.90MM	BFF90	KG	.	BFF 0.90MM
657	BOBINA BFF 1.50MM	BFQ150	KG	.	BFQ 1.50MM
605	BOBINA BFF 1.80MM	BFQ180	KG	.	BFQ 1.80MM
651	BOBINA BFF 2.00MM	BFQ200	KG	.	BFQ 2.00MM
658	BOBINA BFF 2.16MM	BFQ216	KG	.	BFQ 2.16MM
652	BOBINA BFF 2.25MM	BFQ225	KG	.	BFQ 2.25MM
650	BOBINA BFF 2.40MM	BFQ240	KG	.	BFQ 2.40MM
649	BOBINA BFF 2.50MM	BFQ250	KG	.	BFQ 2.50MM

Local	Descrição	Sub-item	Estoque Físico
1	ESTOQUE 1	0	0.000
7	ESTOQUE 7	0	40.105.000

Tab	Descrição	Preço + IPI
6	TABELA 6	1.00
7	TABELA 7	0.00

Total: 40.105.000

mas

# Uma situação mais simples...



# Como efetuar uma busca em um conjunto de valores, tais como em vetores?

# Estrutura de dados



## Introdução – busca

**Considere um vetor com 10 números inteiros.**

4	5	98	32	41	22	7	65	18	57
---	---	----	----	----	----	---	----	----	----

**Suponha que você queira verificar se o número 41 está no vetor. Como seria feita essa verificação?**

**Como seria um algoritmo para representar esse processo?**

# Estrutura de dados



## Introdução – busca

Dada uma coleção de  $n$  elementos, pretende-se saber se um determinado elemento/valor está presente nesta coleção.

Neste curso, vamos considerar que essa coleção de elementos é implementada como um vetor de  $n$  elementos:  $\text{vet}[0] \dots \text{vet}[n-1]$ .

# Estrutura de dados



## Busca

- Busca Sequencial ou linear
  - Não ordenada
  - Ordenada
- Busca Binária

# Estrutura de dados



## Busca Sequencial

Exemplo: Como verificar se o elemento 49 está no vetor?

14	5	21	15	8	49	90	36
----	---	----	----	---	----	----	----

# Estrutura de dados



## Busca Sequencial

Exemplo: Como verificar se o elemento 49 está no vetor?

14	5	21	15	8	49	90	36
----	---	----	----	---	----	----	----




# Estrutura de dados



## Busca Sequencial

Exemplo: Como verificar se o elemento 49 está no vetor?

14	5	21	15	8	49	90	36
----	---	----	----	---	----	----	----

A blue arrow points upwards from below the second cell of the table, which contains the value 5.


# Estrutura de dados



## Busca Sequencial

Exemplo: Como verificar se o elemento 49 está no vetor?

14	5	21	15	8	49	90	36
----	---	----	----	---	----	----	----

A blue arrow points upwards from the bottom center of the slide towards the third cell of the vector, which contains the number 21.




# Estrutura de dados



## Busca Sequencial

Exemplo: Como verificar se o elemento 49 está no vetor?

14	5	21	15	8	49	90	36
----	---	----	----	---	----	----	----

A blue arrow points upwards from below the table to the cell containing the number 15.

# Estrutura de dados



## Busca Sequencial

Exemplo: Como verificar se o elemento 49 está no vetor?

14	5	21	15	8	49	90	36
----	---	----	----	---	----	----	----



# Estrutura de dados



## Busca Sequencial

Exemplo: Como verificar se o elemento 49 está no vetor?

14	5	21	15	8	49	90	36
----	---	----	----	---	----	----	----



# Estrutura de dados



## Busca Sequencial

- Método bastante simples;
- Utilizado quando os dados não estão ordenados;
- Percorre o vetor, elemento por elemento, verificando se o elemento desejado está presente no vetor e termina:

# Estrutura de dados



## Busca Sequencial

- Método bastante simples;
- Utilizado quando os dados não estão ordenados;
- Percorre o vetor, elemento por elemento, verificando se o elemento desejado está presente no vetor e termina:
  - **Com sucesso:** valor pesquisado é encontrado;
  - **Sem sucesso:** todos os registros são pesquisados, mas o valor não é encontrado.

# Estrutura de dados



## Busca Sequencial

```
int buscaSeq(int n, int valor) {  
    int i;  
    for (i=0; i<n; i++){  
        if (vet[i] == valor)  
            return i;  
    }  
    return -1;  
}
```

# Estrutura de dados



## Busca Sequencial

### Tratamento da Informação:

```
pos= buscaSeq(n, valor);  
if (pos !=-1)  
    printf(" %d foi encontrado na posição %d",  
valor, pos);  
else  
    printf(" %d não foi encontrado na lista.", valor);
```

# Estrutura de dados



## Busca Sequencial

Quantas vezes a comparação `valor == vetor[i]` é executada?

Caso o elemento não esteja no vetor,  $n$  vezes;

Caso o elemento esteja no vetor:

- 1 vez, no melhor caso;
- $N$  vezes, no pior caso;
- $N/2$  vezes, no caso médio;



# Estrutura de dados



## Busca Ordenada

Suponha que o vetor utilizado esteja ordenado de forma crescente.

Como poderíamos realizar a busca de forma mais eficiente?

12	25	33	37	48	57	86	92
----	----	----	----	----	----	----	----

# Estrutura de dados



## Busca Ordenada

Considere que estamos procurando o valor 40.

12	25	33	37	48	57	86	92
----	----	----	----	----	----	----	----




# Estrutura de dados



## Busca Ordenada

Considere que estamos procurando o valor 40.

12	25	33	37	48	57	86	92
----	----	----	----	----	----	----	----

A blue arrow points upwards from the bottom of the slide towards the second cell of the array, which contains the value 25.

# Estrutura de dados



## Busca Ordenada

Considere que estamos procurando o valor 40.

12	25	33	37	48	57	86	92
----	----	----	----	----	----	----	----




# Estrutura de dados



## Busca Ordenada

Considere que estamos procurando o valor 40.

12	25	33	37	48	57	86	92
----	----	----	----	----	----	----	----

A blue arrow points upwards from the bottom center of the slide towards the value 37 in the fourth cell of the array.

# Estrutura de dados



## Busca Ordenada

Considere que estamos procurando o valor 40.

12	25	33	37	48	57	86	92
----	----	----	----	----	----	----	----



**48 <= 40 ?**

# Estrutura de dados



## Busca Ordenada

Logo, quando o valor em uma determinada posição ultrapassar o valor procurado, não precisamos continuar a busca.

# Estrutura de dados



## Busca Ordenada

```
int buscaOrd(int n, int valor) {  
    int i;  
    for (i=0; i<n; i++){  
        if (vet[i] == valor)  
            return i;  
        if (vet[i] > valor)  
            return -1;  
    }  
    return -1;  
}
```



# Estrutura de dados



## Busca Binária

Considerando um vetor ordenado de forma crescente podemos aplicar outro método de busca: a busca binária.

# Estrutura de dados



## Busca Binária

Exemplo: Busca pelo valor 86.

Inf = 0

Sup = 7

12	25	33	37	48	57	86	92
----	----	----	----	----	----	----	----

↑  
meio

**86 == 37 ?**

# Estrutura de dados



## Busca Binária

Exemplo: Busca pelo valor 86.

Inf = 0

Sup = 7

12	25	33	37	48	57	86	92
----	----	----	----	----	----	----	----



meio

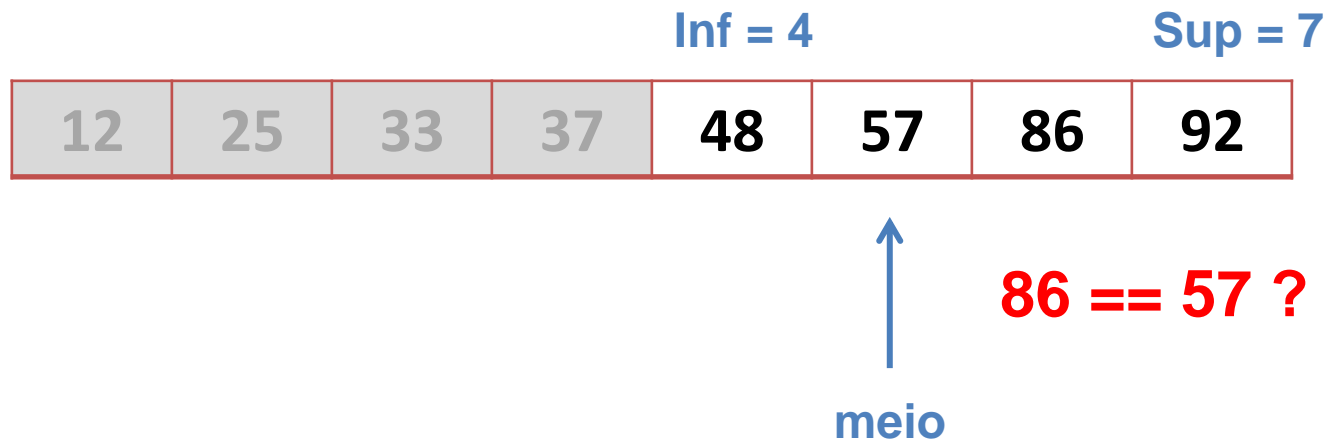
**86 < 37 ?**

# Estrutura de dados



## Busca Binária

Exemplo: Busca pelo valor 86.

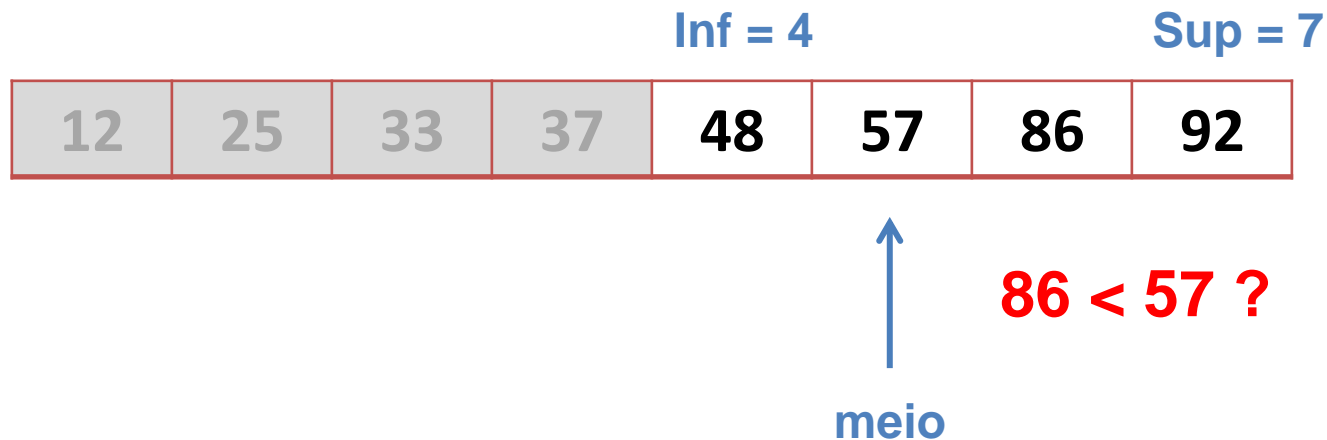


# Estrutura de dados



## Busca Binária

Exemplo: Busca pelo valor 86.

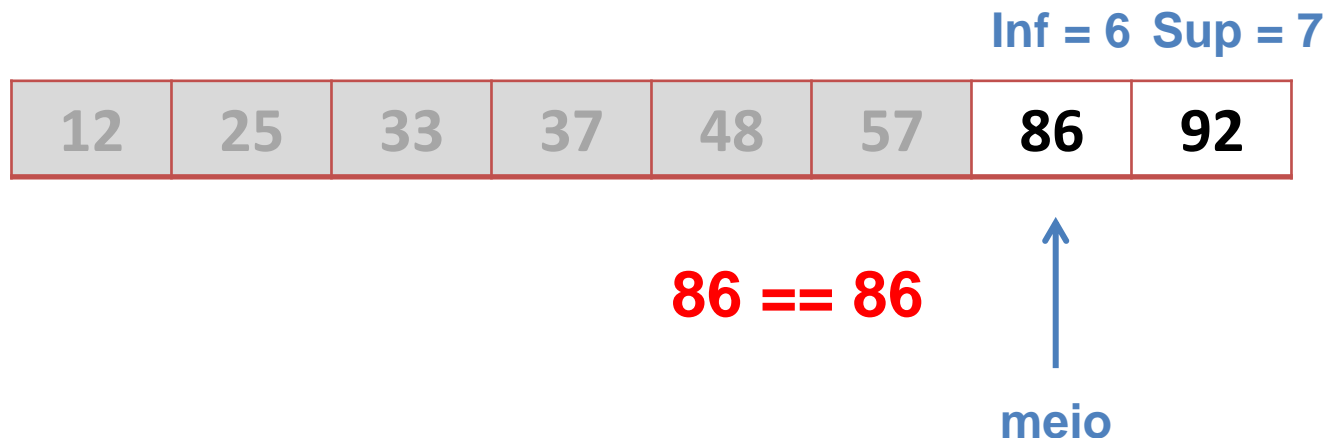


# Estrutura de dados



## Busca Binária

Exemplo: Busca pelo valor 86.



Em cada passo, o tamanho do vetor é reduzido pela metade.

# Estrutura de dados



## Busca Binária

Ao comparar um determinado elemento do vetor com o valor desejado é possível identificar:

- Se o elemento procurado é o elemento comparado;

- Se o elemento procurado está antes ou depois do elemento comparado.

# Estrutura de dados



## Busca Binária

Compara sempre o valor desejado com o elemento do meio da lista;

A cada comparação, a lista será dividida em duas;

Se após diversas divisões, a lista possuir tamanho 0 (zero), então o elemento não está na lista.



# Estrutura de dados



## Busca Binária

O elemento buscado é comparado ao elemento do meio da lista:

- Se igual, busca bem-sucedida;
- Se menor, busca-se na metade inferior da lista;
- Se maior, busca-se na metade superior da lista.

# Estrutura de dados



## Busca Binária

```
int pesqBin (int valor, int n){  
    int inf, sup, meio;  
    inf= 0; sup=n-1;
```

## Busca Binária

```
while (inf <= sup) {  
    meio = floor( (inf + sup) / 2 );  
    if (valor == v[meio])  
        return(meio);  
    else  
        if (valor > v[meio])  
            inf = meio + 1  
        else  
            sup= meio -1;  
}  
if (inf>sup)    return -1;  
}
```

# Estrutura de dados



## Busca – exercícios

Simule a busca binária para os seguintes casos:

- $V = [33, 45, 57, 87, 99, 120, 132, 135, 145]$ ,  
chave = 135
- $V = [1, 4, 7, 9, 15, 18, 19, 26, 33]$ , chave = 7

# DÚVIDAS?