



# ESTRUTURA DE DADOS

Engenharia da Computação

Prof. Renato Matroniani



EDUCAÇÃO  
METODISTA

# ESTRUTURA DE DADOS

- Métodos de pesquisa e ordenação de dados.  
+ análise de complexidade



EDUCAÇÃO  
METODISTA

# MÉTODOS DE PESQUISA (BUSCA) DE DADOS

- Dependendo da literatura: busca ou pesquisa.
- Os algoritmos de busca são alguns dos mais utilizados, sendo usados em bancos de dados, internet, jogos, entre outros.
- A escolha do método a ser utilizado para busca depende muito da **quantidade de dados envolvidos**, do volume de **operações de inclusão e exclusão** a serem realizadas, entre outros fatores que devem ser considerados quando do desenvolvimento da aplicação.
- Busca linear (ou sequencial) e busca binária



EDUCAÇÃO  
METODISTA



# PESQUISA LINEAR (SEQUENCIAL)

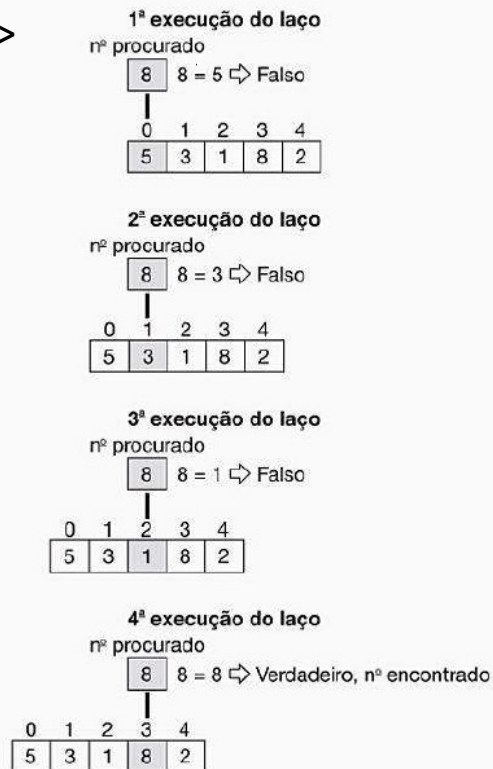
- Um algoritmo de busca sequencial pode ser executado em vetores ordenados ou não ordenados.
- No caso de vetores não ordenados, o número é buscado até que ele seja encontrado **ou quando termina o vetor**.
- No caso de vetores ordenados, o número é buscado até que ele seja encontrado **e/ou enquanto** for menor que o tamanho do vetor.



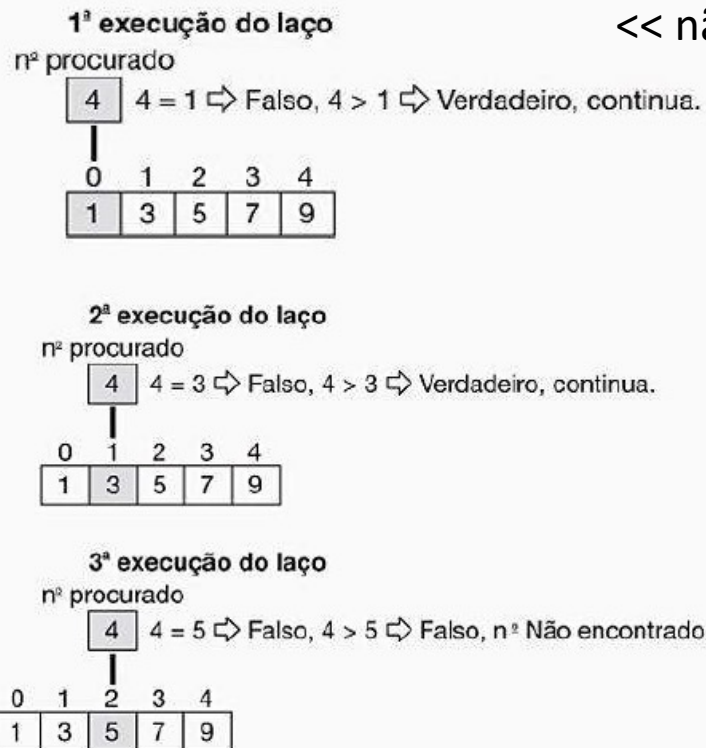
EDUCAÇÃO  
METODISTA

# PESQUISA LINEAR (SEQUENCIAL)

<< Ordenado >>



<< não ordenado >>



# PESQUISA LINEAR (SEQUENCIAL)

```
algoritmo
declare X[10], i, n, achou numérico
// carregando os números no vetor
// vetor com números NÃO ORDENADOS
para i ← 1 até 10 faça
    início
        escreva "Digite o ",i,"º número: "
        leia X[i]
    fim
// digitando o número a ser buscado no vetor
escreva "Digite o número a ser buscado no vetor: "
leia n
// buscando o número digitado no vetor
achou ← 0
i ← 0
enquanto (i <= 9 e achou = 0) faça
    início
        se (X[i] = n)
            então achou ← 1
            senão i ← i + 1
        fim
se (achou = 0)
    então escreva "Número não encontrado no vetor"
    senão escreva "Número encontrado na posição ",i+1
fim_algoritmo.
```



**EDUCAÇÃO  
METODISTA**



# PESQUISA LINEAR (SEQUENCIAL)

```
algoritmo
declare X[10], i, n, achou numérico
// carregando os números no vetor
// vetor com números ORDENADOS
para i ← 1 até 10 faça
    início
        escreva "Digite o ",i,"º número: "
        leia X[i]
    fim
// digitando o número a ser buscado no vetor
escreva "Digite o número a ser buscado no vetor: "
leia n
// buscando o número digitado no vetor
achou ← 0
i ← 0
enquanto (i <= 9 e achou = 0 e n >= X[i]) faça
    início
        se (X[i] = n)
            então achou ← 1
            senão i ← i + 1
        fim
se (achou = 0)
    então escreva "Número não encontrado no vetor"
    senão escreva "Número encontrado na posição ",i+1
fim_algoritmo.
```



**EDUCAÇÃO  
METODISTA**

# PESQUISA LINEAR (SEQUENCIAL)

- Análise de complexidade em algoritmo não ordenado
- Qual seria o **pior caso?**
  - $T(n) = O(n)$
- Qual seria o **melhor caso?**
  - $T(n) = O(1)$

```
1. achou ← 0
2. i ← 0
3. enquanto (i ≤ 9 e achou = 0) faça
4.   início
5.     se {X[i] = n}
6.       então achou ← 1
7.       senão i ← i + 1
8.   fim
```





# PESQUISA LINEAR (SEQUENCIAL)

- Análise de complexidade em algoritmo ordenado
- Qual seria o **pior caso?**
  - $T(n) = O(n)$
- Qual seria o **melhor caso?**
  - $T(n) = O(1)$

```
1. achou ← 0
2. i ← 0
3. enquanto (i ≤ 9 e achou = 0 e n ≥ X[i]) faça
4.   início
5.     se {X[i] = n}
6.       então achou ← 1
7.       senão i ← i + 1
8.   fim
```



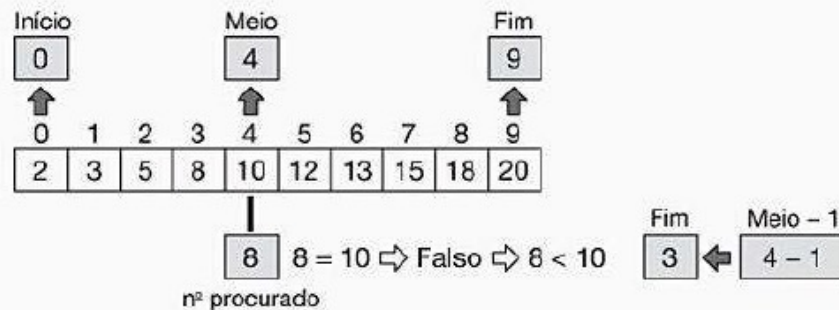
# PESQUISA BINÁRIA

- Um algoritmo de busca binária é executado somente em vetores ordenados.
- Como funciona:
  - o vetor com os dados é dividido ao meio;
  - o número do meio é comparado com o número pesquisado;
  - se forem iguais, a busca termina.
  - senão, se for menor que o número do meio, buscar à esquerda.
  - se for maior que o número do meio, buscar a direita.
  - na busca à esquerda ou à direita, fazer novamente a divisão do vetor.
  - a busca termina quando o vetor ficar com um elemento ou o número ser encontrado.

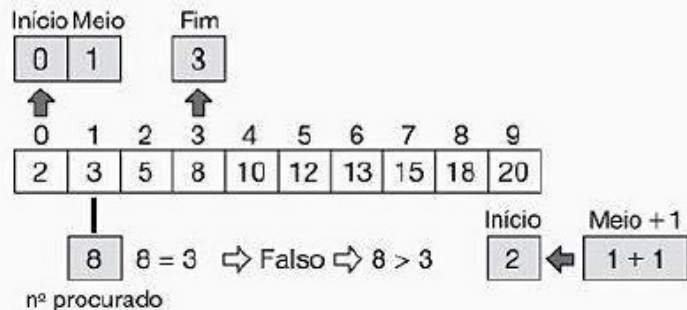


# PESQUISA BINÁRIA

## 1ª execução do laço



## 2ª execução do laço

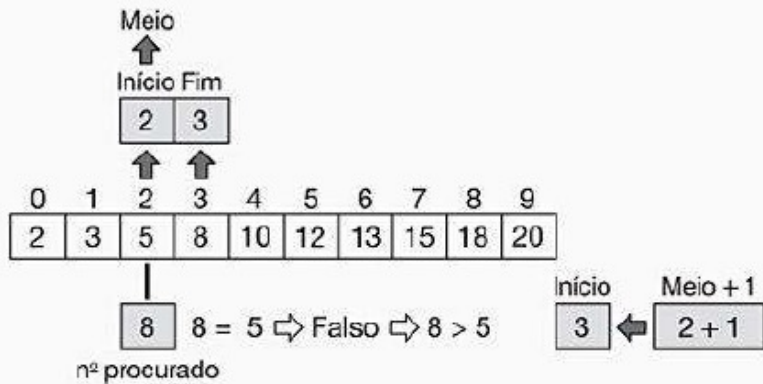


**EDUCAÇÃO  
METODISTA**

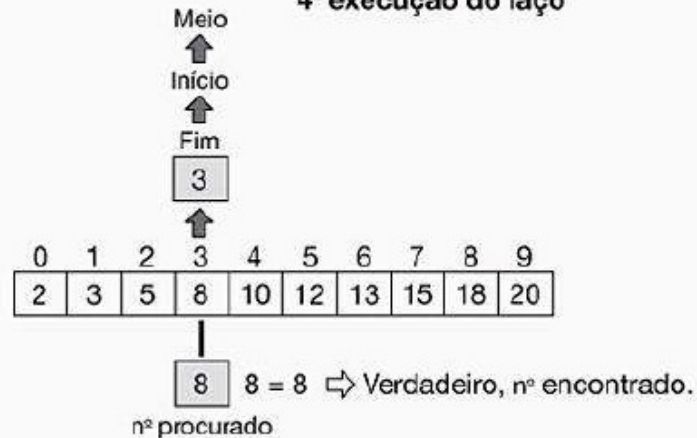


# PESQUISA BINÁRIA

3ª execução do laço



4ª execução do laço



**EDUCAÇÃO  
METODISTA**

# PESQUISA BINÁRIA

```
algoritmo
declare X[10], i, np, achou, início, fim, meio numérico
// carregando os números no vetor
para i ← 1 até 10 faça
    início
        escreva "Digite o ",i,"º número: "
        leia X[i]
    fim
// digitando o número a ser buscado no vetor
escreva "Digite o número a ser buscado no vetor: "
leia np
// buscando o número digitado no vetor
achou ← 0
início ← 0
fim ← 9
meio ← parteinteira((início+fim)/2)
enquanto (início <= fim e achou = 0) faça
    início
        se (X[meio] = np)
            então achou ← 1
            senão início
                se (np < X[meio])
                    então fim ← meio-1
                senão início ← meio+1
                meio ← parteinteira((início+fim)/2)
    fim
```

```
    fim
se (achou = 0)
    então escreva "Número não encontrado no vetor"
    senão escreva "Número encontrado na posição ",meio
fim_algoritmo.
```



**EDUCAÇÃO  
METODISTA**

# PESQUISA BINÁRIA

- Análise de complexidade:
- a cada laço executado, o tamanho do vetor é dividido por 2.
- Se  $n$  é igual ao tamanho do vetor, temos:

$$\frac{n}{2} \rightarrow \frac{n}{4} \rightarrow \frac{n}{8} \rightarrow \dots \rightarrow \frac{n}{2^k} \leq 1$$

para tamanho de vetores = 1  
temos o final da busca

```
1. achou ← 0
2. início ← 0
3. fim ← 9
4. meio ← parteinteira((início+fim)/2)
5. enquanto (início ≤ fim e achou = 0) faça
6.   início
7.       se (X[meio] = np)
8.         então achou ← 1
9.       senão início
10.          se (np < X[meio])
11.            então fim ← meio-1
12.          senão início ← meio+1
13.          meio ← parteinteira((início+fim)/2)
14.        fim
15. fim
```



**EDUCAÇÃO  
METODISTA**



# PESQUISA BINÁRIA

- Análise de complexidade:  
para tamanho de vetores = 1 temos  
o final da busca

$$\text{então: } \frac{n}{2^k} = 1 \rightarrow 2^k = n$$

$$\text{ou } \log_2 2^k = \log_2 n$$

$$\therefore k = \log_2 n$$

- A função complexidade de tempo  
é proporcional a  $\log n$ .
- A busca binária também é  
conhecida como busca  
logarítmica.

```
1. achou ← 0
2. início ← 0
3. fim ← 9
4. meio ← parteinteira((início+fim)/2)
5. enquanto (início ≤ fim e achou = 0) faça
6.   início
7.       se (X[meio] = np)
8.         então achou ← 1
9.       senão início
10.          se (np < X[meio])
11.            então fim ← meio-1
12.            senão início ← meio+1
13.          meio ← parteinteira((início+fim)/2)
14.        fim
15. fim
```



**EDUCAÇÃO  
METODISTA**

# PESQUISA BINÁRIA X LINEAR

- O método de busca linear é o mais adequado quando não se conhece a estrutura em que será realizada essa busca.
- Mas, se o elemento estiver entre os últimos ou não estiver no vetor, a busca pode se tornar demorada.
- No caso das sequências ordenadas, existem os algoritmos de busca binária, que são mais eficientes (pode-se eliminar metade do vetor na primeira iteração).
- A busca binária é conhecida também por busca logarítmica, como foi visto, pois o número de elementos buscados decresce de forma algorítmica a cada iteração.



**EDUCAÇÃO  
METODISTA**

# PESQUISA BINÁRIA X LINEAR

