

Estruturas de Dados

Prof. Rodrigo Martins

rodrigo.martins@francomontoro.com.br

Cronograma da Aula

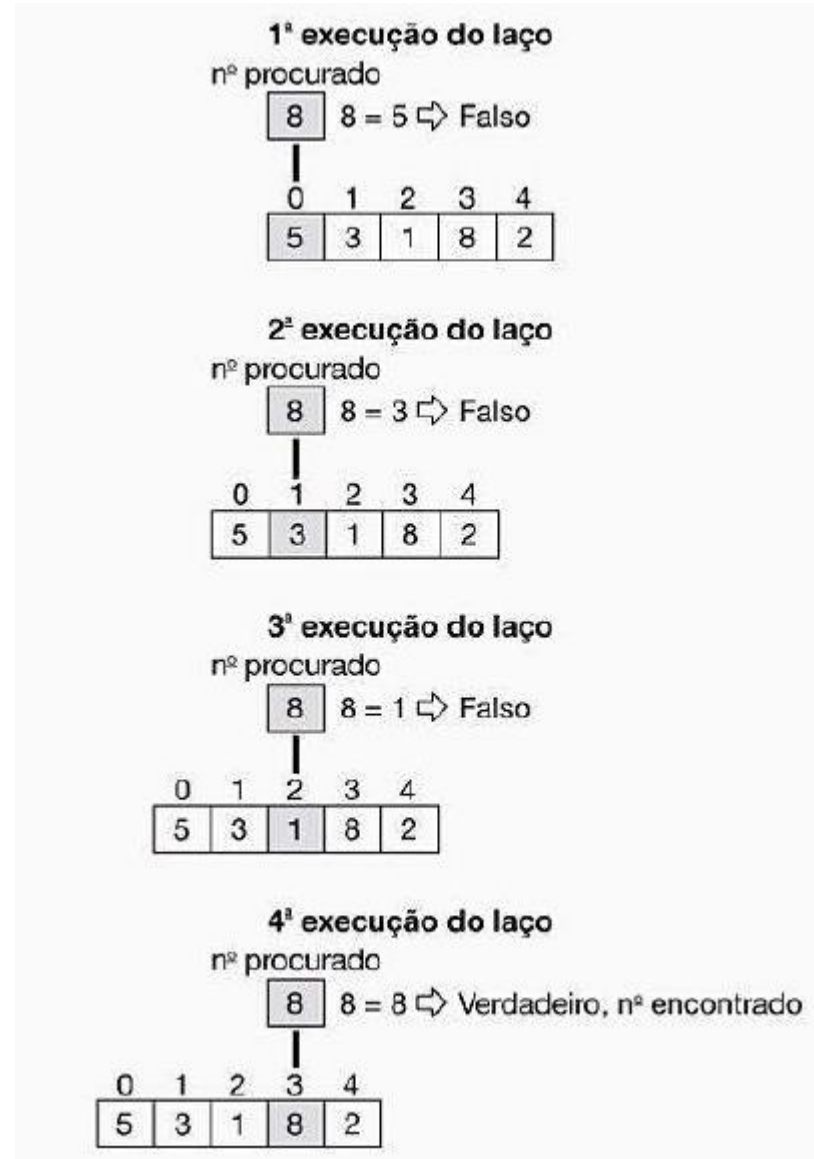
- Algoritmos de busca
 - Sequencial
 - Binária
- Exemplos
- Exercício

Algoritmo de busca sequencial

- O algoritmo de busca sequencial pode ser executado em um vetor não ordenado e em um vetor ordenado.
- Em um vetor não ordenado, será buscado o número até que ele seja encontrado ou até se chegar ao final do vetor.
- E um vetor ordenado, será buscado o número até que ele seja encontrado e enquanto for maior que o número do vetor.

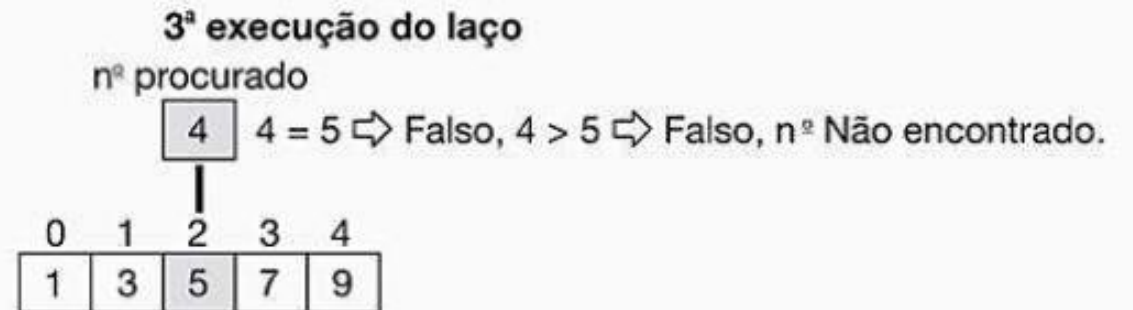
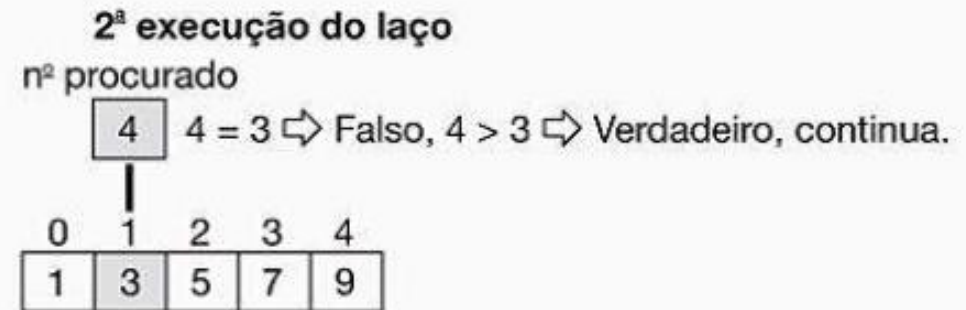
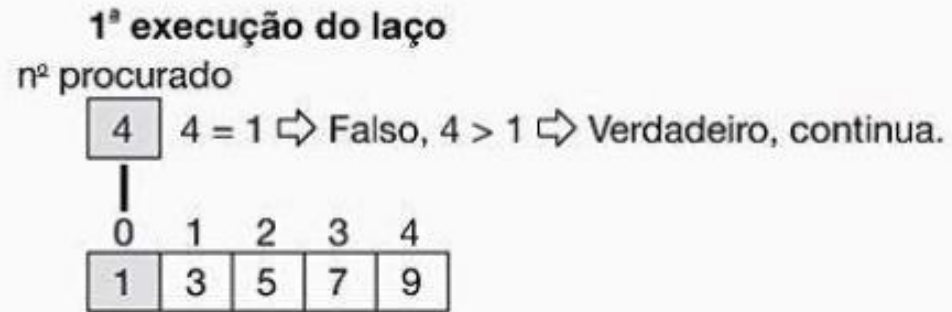
Algoritmo de busca sequencial

- Vetor não ordenado



Algoritmo de busca sequencial

- Vetor ordenado



Algoritmo de busca sequencial

```
*buscaSequencialNaoOrdenad
1  #include <iostream>
2
3  using namespace std;
4
5  void buscar(int x[5])
6  {
7      int achou = 0;
8      int i = 0, num = 0;
9
10     // Digitando o número a ser buscado
11     cout << "Digite o numero a ser buscado no vetor: ";
12     cin >> num;
13
14     while (i <= 5 && achou == 0)
15     {
16
17         if (x[i] == num)
18         {
19             achou = 1;
20         }
21         else
22         {
23             i++;
24         }
25     }
26 }
```

Algoritmo de busca sequencial

```
27     if (achou == 0)
28     {
29         cout << "Numero nao encontrado no vetor" << endl;
30     }
31     else
32     {
33         cout << "Numero encontrado no vetor na posicao " << i + 1;
34     }
35
36
37 }
38
39 int main(int argc, char** argv)
40 {
41     int x[5];
42     //carregando os números no vetor
43     //vetor com números não ordenados
44
45     for (int i = 0; i < 5; i++)
46     {
47         cout << "Digite o numero: " << endl;
48         cin >> x[i];
49     }
50
51     buscar(x);
52     return 0;
53 }
```

Algoritmo de busca sequencial

```
buscaSequencialOrdenado.cpp
1  #include <iostream>
2
3  using namespace std;
4
5  void buscar(int x[5])
6  {
7      int achou = 0;
8      int i = 0, num = 0;
9
10     // Digitando o número a ser buscado
11     cout << "Digite o numero a ser buscado no vetor: ";
12     cin >> num;
13
14     while (i <= 5 && achou == 0 && num >= x[i])
15     {
16         if (x[i] == num)
17         {
18             achou = 1;
19         }
20         else
21         {
22             i++;
23         }
24     }
25 }
```

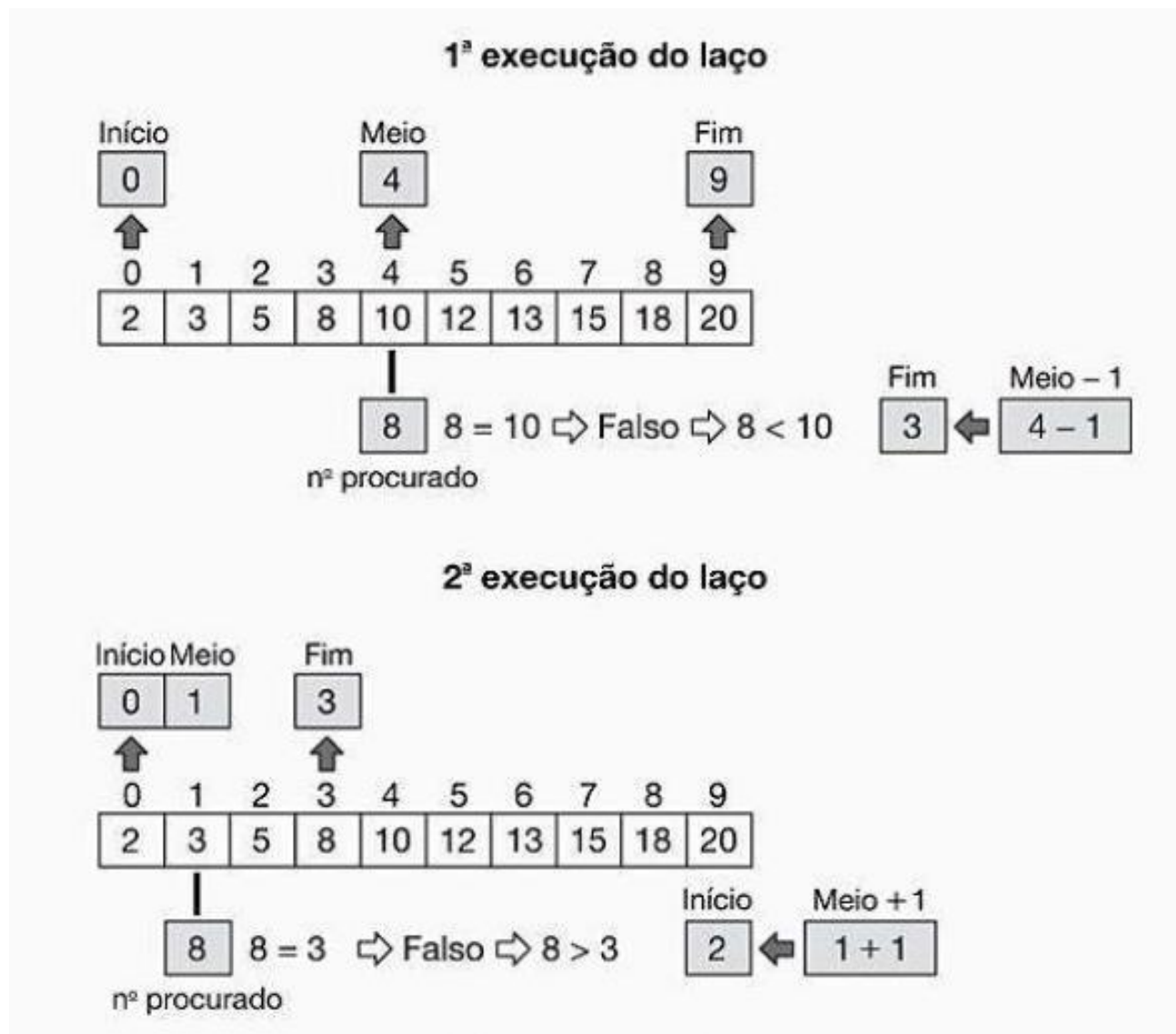

Algoritmo de busca sequencial

```
26     if (achou == 0)
27     {
28         cout << "Numero nao encontrado no vetor" << endl;
29     }
30     else
31     {
32         cout << "Numero encontrado no vetor na posicao " << i + 1;
33     }
34
35
36 }
37
38 int main(int argc, char** argv)
39 {
40     int x[5];
41     //carregando os números no vetor
42     //vetor com números ordenados
43
44     for (int i = 0; i < 5; i++)
45     {
46         cout << "Digite o numero: " << endl;
47         cin >> x[i];
48     }
49
50     buscar(x);
51     return 0;
52 }
```

Algoritmo de busca binária

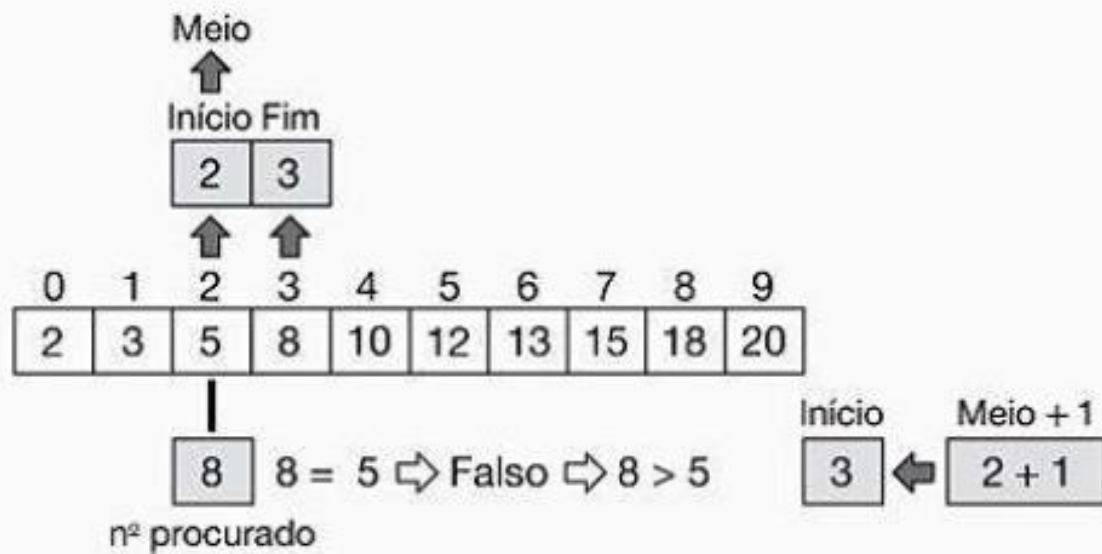
- O algoritmo de busca binária é executado somente em vetores ordenados.
- Nesse algoritmo o vetor com os dados é dividido ao meio e o número do meio é comparado com o número procurado.
- Se estes forem iguais, a busca termina, caso contrário, se o número procurado é menor que o do meio, a busca será realizada no vetor à esquerda ao do meio.
- Se o número procurado é maior do que o do meio, a busca será realizada no vetor à direita ao vetor do meio.
- Esse procedimento de divisão e comparação acontece até que o vetor de dados fique com apenas um elemento ou até o número procurado ser encontrado.

Algoritmo de busca binária



Algoritmo de busca binária

3ª execução do laço



4ª execução do laço



Algoritmo de busca binária

```
buscaBinaria.cpp x
1  #include <iostream>
2
3  using namespace std;
4
5  void buscar(int x[])
6  {
7      int achou = 0, inicio = 0, fim = 9, meio = 0;
8      int num = 0;
9      meio = (inicio + fim) / 2;
10
11     // Digitando o número a ser buscado
12     cout << "Digite o numero a ser buscado no vetor: ";
13     cin >> num;
14
15     fim = 9;
16     meio = (inicio + fim) / 2;
17
```

Algoritmo de busca binária

```
18 while (inicio <= fim && achou == 0)
19 {
20     if (x[meio] == num)
21         achou = 1;
22
23     else
24     {
25         if(num < x[meio])
26             fim = meio - 1;
27         else
28             inicio = meio + 1;
29
30         meio = (inicio + fim) / 2;
31     }
32 }
33
34 if (achou == 0)
35 {
36     cout << "Numero nao encontrado no vetor" << endl;
37 }
38 else
39 {
40     cout << "Numero encontrado no vetor na posicao " << meio + 1;
41 }
42 }
43
```

Algoritmo de busca binária

```
44 int main(int argc, char** argv)
45 {
46     int x[10];
47     //carregando os números no vetor
48     //vetor com números ordenados
49
50     for (int i = 0; i < 10; i++)
51     {
52         cout << "Digite o numero: " << endl;
53         cin >> x[i];
54     }
55
56     buscar(x);
57     return 0;
58 }
59
```

Quiz

Essa atividade vale pontos para a próxima avaliação.

1º colocado – Ganha 2 pontos

2º colocado – Ganha 1 ponto

3º colocado – Ganha 0,5 ponto

BOM JOGO!

<https://kahoot.it/>

Referência desta aula

- <https://www.devmedia.com.br/algoritmos-de-ordenacao-analise-e-comparacao/28261>
- <https://www.devmedia.com.br/algoritmos-de-ordenacao/2622>
- <https://www.treinaweb.com.br/blog/conheca-os-principais-algoritmos-de-ordenacao/>
- <http://www.cplusplus.com/reference/>
- Ascencio, A. F. G. (2010). Estruturas De Dados: ALGORITMOS, ANÁLISE DA COMPLEXIDADE E IMPLEMENTAÇÃO. Brasil: PEARSON BRASIL.

Obrigado

Rodrigo