



Métodos de Ordenação Quadráticos

Prof. Lilian Berton

São José dos Campos, 2018

Introdução: ordenação

- **Ordenar:** processo de rearranjar um conjunto de objetos em uma ordem ascendente ou decendente.
- A ordenação **visa facilitar a recuperação** posterior de itens do conjunto ordenado.
- Um método de ordenação **é estável** se a ordem relativa dos itens com chaves iguais não se altera durante a ordenação.
- **Ordenação interna:** Arquivo a ser ordenado cabe todo na memória principal, qualquer registro pode ser imediatamente acessado.
- **Ordenação externa:** Arquivo não cabe todo na memória principal, sendo necessário acessos a memória secundária.



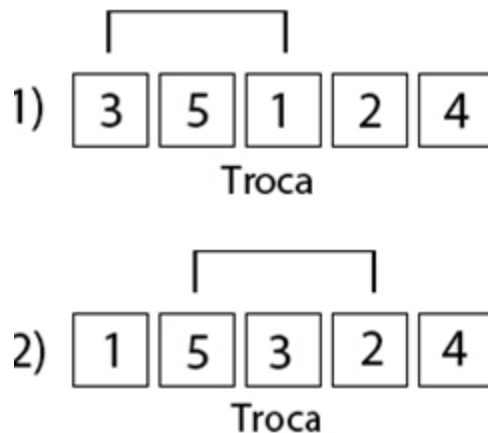
Sobre estabilidade

- Um método de ordenação é **estável** se a **ordem relativa dos itens com chaves iguais não se altera durante a ordenação.**
- Alguns dos métodos de ordenação mais eficientes não são estáveis.

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|----|
| 2 | 7 | 2 | 9 | 5 | 3 | 2 | 1 | 8 | 10 |
| 1 | 7 | 2 | 9 | 5 | 3 | 2 | 2 | 8 | 10 |

Introdução: ordenação

- Sendo n o número de registros no arquivo, as medidas de complexidade relevantes são:
 - Número de comparações $C(n)$ entre chaves.
 - Número de movimentações $M(n)$ de itens do arquivo.



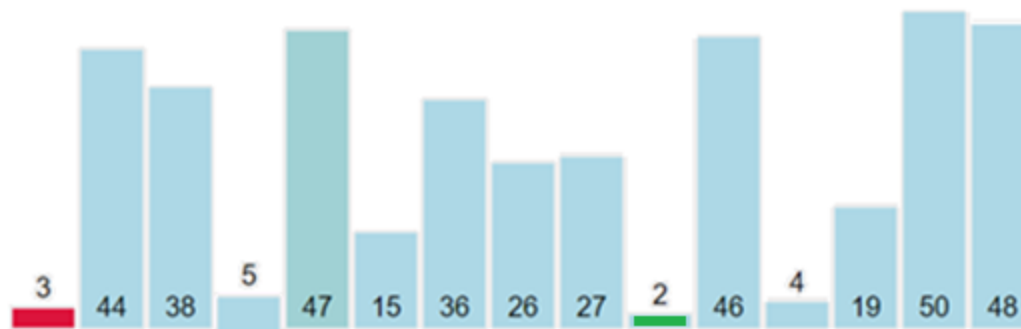
Focaremos a complexidade no número de comparações!

Algoritmos de ordenação em memória interna

- **Quadráticos $O(n^2)$:**
 - Ordenação por Seleção (Selection Sort)
 - Ordenação por Inserção (Insertion Sort)
 - Ordenação por Bolha (BubbleSort)
- **$O(n \log n)$:**
 - Quicksort
 - Heapsort
 - Shellsort
- **Lineares $O(n)$:**
 - Ordenação por contagem
 - Radix sort
 - Bucket sort

Ordenação por seleção


- Algoritmo:
 - Selecione o menor item do vetor.
 - Troque-o com o item da primeira posição do vetor.
 - Repita essas duas operações com os $n - 1$ itens restantes, depois com os $n - 2$ itens, até que reste apenas um elemento.



Exemplo ordenação por seleção


i
0

| | | | | | | | | | |
|----|---|---|---|----|---|----|---|----|----|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 22 | 5 | 9 | 1 | 15 | 3 | 14 | 8 | 19 | 10 |




Exemplo ordenação por seleção

| | | | | | | | | | | |
|---|----|---|---|----|----|---|----|---|----|----|
| i | | | | | | | | | | |
| 0 | 22 | 5 | 9 | 1 | 15 | 3 | 14 | 8 | 19 | 10 |
| 1 | 1 | 5 | 9 | 22 | 15 | 3 | 14 | 8 | 19 | 10 |




Exemplo ordenação por seleção

| | | | | | | | | | | |
|---|----|---|---|----|----|---|----|---|----|----|
| i | | | | | | | | | | |
| 0 | 22 | 5 | 9 | 1 | 15 | 3 | 14 | 8 | 19 | 10 |
| 1 | 1 | 5 | 9 | 22 | 15 | 3 | 14 | 8 | 19 | 10 |
| 2 | 1 | 3 | 9 | 22 | 15 | 5 | 14 | 8 | 19 | 10 |




Exemplo ordenação por seleção

| | | | | | | | | | | |
|---|----|---|---|----|----|---|----|---|----|----|
| i | | | | | | | | | | |
| 0 | 22 | 5 | 9 | 1 | 15 | 3 | 14 | 8 | 19 | 10 |
| 1 | 1 | 5 | 9 | 22 | 15 | 3 | 14 | 8 | 19 | 10 |
| 2 | 1 | 3 | 9 | 22 | 15 | 5 | 14 | 8 | 19 | 10 |
| 3 | 1 | 3 | 5 | 22 | 15 | 9 | 14 | 8 | 19 | 10 |



Exemplo ordenação por seleção

| | | | | | | | | | | |
|---|----|---|---|----|----|---|----|----|----|----|
| i | | | | | | | | | | |
| 0 | 22 | 5 | 9 | 1 | 15 | 3 | 14 | 8 | 19 | 10 |
| 1 | 1 | 5 | 9 | 22 | 15 | 3 | 14 | 8 | 19 | 10 |
| 2 | 1 | 3 | 9 | 22 | 15 | 5 | 14 | 8 | 19 | 10 |
| 3 | 1 | 3 | 5 | 22 | 15 | 9 | 14 | 8 | 19 | 10 |
| 4 | 1 | 3 | 5 | 8 | 15 | 9 | 14 | 22 | 19 | 10 |



Exemplo ordenação por seleção

| | | | | | | | | | | |
|---|----|---|---|----|----|----|----|----|----|----|
| i | | | | | | | | | | |
| 0 | 22 | 5 | 9 | 1 | 15 | 3 | 14 | 8 | 19 | 10 |
| 1 | 1 | 5 | 9 | 22 | 15 | 3 | 14 | 8 | 19 | 10 |
| 2 | 1 | 3 | 9 | 22 | 15 | 5 | 14 | 8 | 19 | 10 |
| 3 | 1 | 3 | 5 | 22 | 15 | 9 | 14 | 8 | 19 | 10 |
| 4 | 1 | 3 | 5 | 8 | 15 | 9 | 14 | 22 | 19 | 10 |
| 5 | 1 | 3 | 5 | 8 | 9 | 15 | 14 | 22 | 19 | 10 |

Exemplo ordenação por seleção

| | | | | | | | | | | |
|---|----|---|---|----|----|----|----|----|----|----|
| i | | | | | | | | | | |
| 0 | 22 | 5 | 9 | 1 | 15 | 3 | 14 | 8 | 19 | 10 |
| 1 | 1 | 5 | 9 | 22 | 15 | 3 | 14 | 8 | 19 | 10 |
| 2 | 1 | 3 | 9 | 22 | 15 | 5 | 14 | 8 | 19 | 10 |
| 3 | 1 | 3 | 5 | 22 | 15 | 9 | 14 | 8 | 19 | 10 |
| 4 | 1 | 3 | 5 | 8 | 15 | 9 | 14 | 22 | 19 | 10 |
| 5 | 1 | 3 | 5 | 8 | 9 | 15 | 14 | 22 | 19 | 10 |
| 6 | 1 | 3 | 5 | 8 | 9 | 10 | 14 | 22 | 19 | 15 |

Exemplo ordenação por seleção

| | | | | | | | | | | |
|---|----|---|---|----|----|----|----|----|----|----|
| i | | | | | | | | | | |
| 0 | 22 | 5 | 9 | 1 | 15 | 3 | 14 | 8 | 19 | 10 |
| 1 | 1 | 5 | 9 | 22 | 15 | 3 | 14 | 8 | 19 | 10 |
| 2 | 1 | 3 | 9 | 22 | 15 | 5 | 14 | 8 | 19 | 10 |
| 3 | 1 | 3 | 5 | 22 | 15 | 9 | 14 | 8 | 19 | 10 |
| 4 | 1 | 3 | 5 | 8 | 15 | 9 | 14 | 22 | 19 | 10 |
| 5 | 1 | 3 | 5 | 8 | 9 | 15 | 14 | 22 | 19 | 10 |
| 6 | 1 | 3 | 5 | 8 | 9 | 10 | 14 | 22 | 19 | 15 |
| 7 | 1 | 3 | 5 | 8 | 9 | 10 | 14 | 22 | 19 | 15 |

Exemplo ordenação por seleção

| | | | | | | | | | | |
|---|----|---|---|----|----|----|----|----|----|----|
| i | | | | | | | | | | |
| 0 | 22 | 5 | 9 | 1 | 15 | 3 | 14 | 8 | 19 | 10 |
| 1 | 1 | 5 | 9 | 22 | 15 | 3 | 14 | 8 | 19 | 10 |
| 2 | 1 | 3 | 9 | 22 | 15 | 5 | 14 | 8 | 19 | 10 |
| 3 | 1 | 3 | 5 | 22 | 15 | 9 | 14 | 8 | 19 | 10 |
| 4 | 1 | 3 | 5 | 8 | 15 | 9 | 14 | 22 | 19 | 10 |
| 5 | 1 | 3 | 5 | 8 | 9 | 15 | 14 | 22 | 19 | 10 |
| 6 | 1 | 3 | 5 | 8 | 9 | 10 | 14 | 22 | 19 | 15 |
| 7 | 1 | 3 | 5 | 8 | 9 | 10 | 14 | 22 | 19 | 15 |
| 8 | 1 | 3 | 5 | 8 | 9 | 10 | 14 | 15 | 19 | 22 |



Exemplo ordenação por seleção

| | | | | | | | | | | |
|---|----|---|---|----|----|----|----|----|----|----|
| i | | | | | | | | | | |
| 0 | 22 | 5 | 9 | 1 | 15 | 3 | 14 | 8 | 19 | 10 |
| 1 | 1 | 5 | 9 | 22 | 15 | 3 | 14 | 8 | 19 | 10 |
| 2 | 1 | 3 | 9 | 22 | 15 | 5 | 14 | 8 | 19 | 10 |
| 3 | 1 | 3 | 5 | 22 | 15 | 9 | 14 | 8 | 19 | 10 |
| 4 | 1 | 3 | 5 | 8 | 15 | 9 | 14 | 22 | 19 | 10 |
| 5 | 1 | 3 | 5 | 8 | 9 | 15 | 14 | 22 | 19 | 10 |
| 6 | 1 | 3 | 5 | 8 | 9 | 10 | 14 | 22 | 19 | 15 |
| 7 | 1 | 3 | 5 | 8 | 9 | 10 | 14 | 22 | 19 | 15 |
| 8 | 1 | 3 | 5 | 8 | 9 | 10 | 14 | 15 | 19 | 22 |
| 9 | 1 | 3 | 5 | 8 | 9 | 10 | 14 | 15 | 19 | 22 |

Algoritmo Seleção

```
procedure Seleccion (var A: Vetor; var n: Indice);  
var i, j, Min: Indice;  
    x      : Item;  
begin  
  for i := 1 to n - 1 do  
    begin  
      Min := i;  
      for j := i + 1 to n do  
        if A[j].Chave < A[Min].Chave  
        then Min := j;  
      x := A[Min]; A[Min] := A[i]; A[i] := x;  
    end;  
  end;
```

$O(n)$

$O(n)$

Total = $O(n^2)$

Análise

1. Qual o número de comparações entre chaves e movimentações no algoritmo de **ordenação por seleção**?

$$C(n) = \sum_{i=2}^n (i-1) = \frac{1}{2} (n-1) n = \frac{1}{2} (n^2 - n) = O(n^2)$$

$$M(n) = 3(n - 1)$$

2. O algoritmo de ordenação por seleção é estável?

O algoritmo de ordenação por seleção **não é estável**.

Selecao (n, A)

```
Para i de 1 até (n - 1)
  repita
    min ← i;
    Para j de (i + 1) até n
      repita
        Se A[j] < A[min]
          Entao min ← j;
    fim-para;
    aux ← A[min];
    A[min] ← A[i];
    A[i] ← aux;
  fim-para.
```

3. Quais as vantagens e desvantagens da ordenação por seleção?

Vantagens: Custo linear no tamanho da entrada para o número de movimentos de registros. É o algoritmo a ser utilizado para arquivos com registros muito grandes.

Desvantagens: O fato de o arquivo já estar ordenado não ajuda em nada, pois o custo continua quadrático.

Algoritmos de ordenação em memória interna

- **Quadráticos $O(n^2)$:**
 - Ordenação por Seleção (Selection Sort)
 - Ordenação por Inserção (Insertion Sort)
 - Ordenação por bolha (BubbleSort)
- **$O(n \log n)$:**
 - Quicksort
 - Heapsort
 - Shellsort
- **Lineares $O(n)$:**
 - Ordenação por contagem
 - Radix sort
 - Bucket sort

Ordenação por inserção

- Algoritmo:
 - em cada passo a partir de $i=2$ faça:
 - Selecione o i -ésimo item da sequência.
 - Coloque-o no lugar apropriado na sequência destino de acordo com o critério de ordenação.

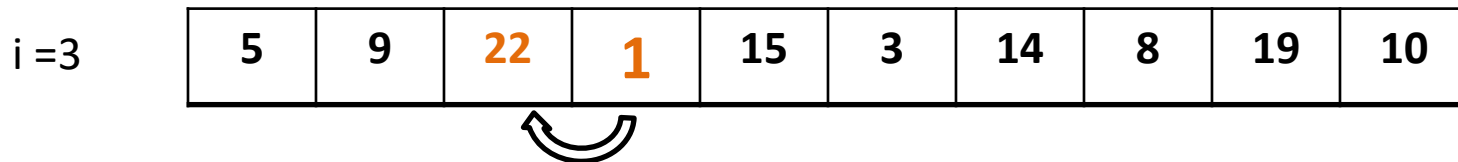
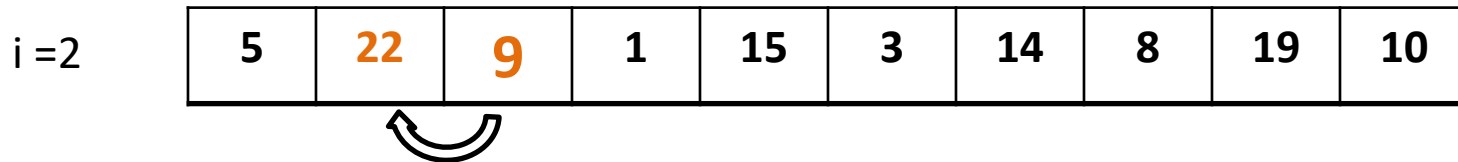
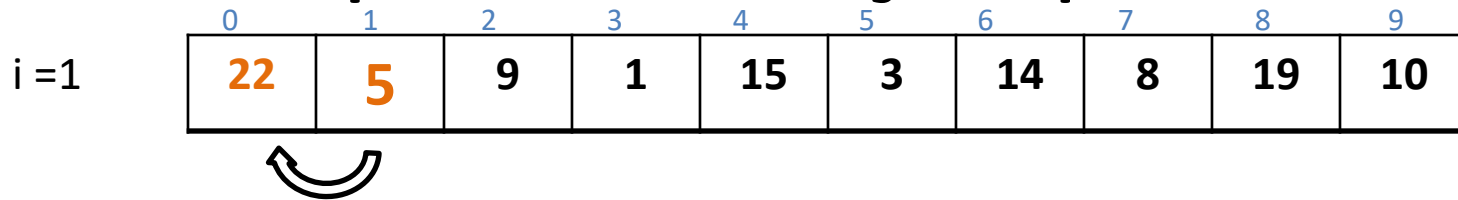
$A < 2 < 3 < \dots < 10 < J < Q < K$

e

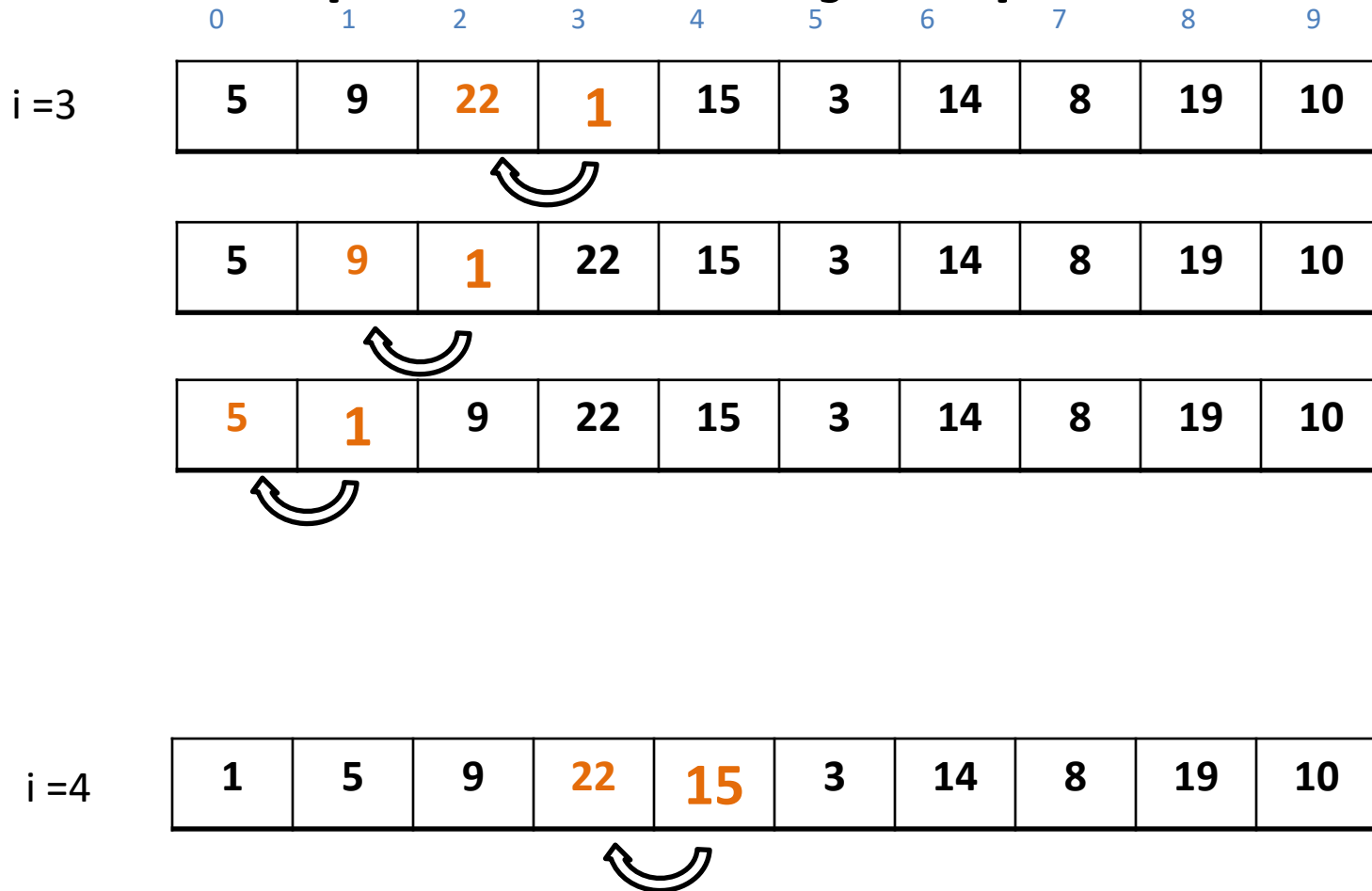
$\clubsuit < \diamond < \heartsuit < \spadesuit$.

Obs: quando jogamos baralhos e organizamos as cartas

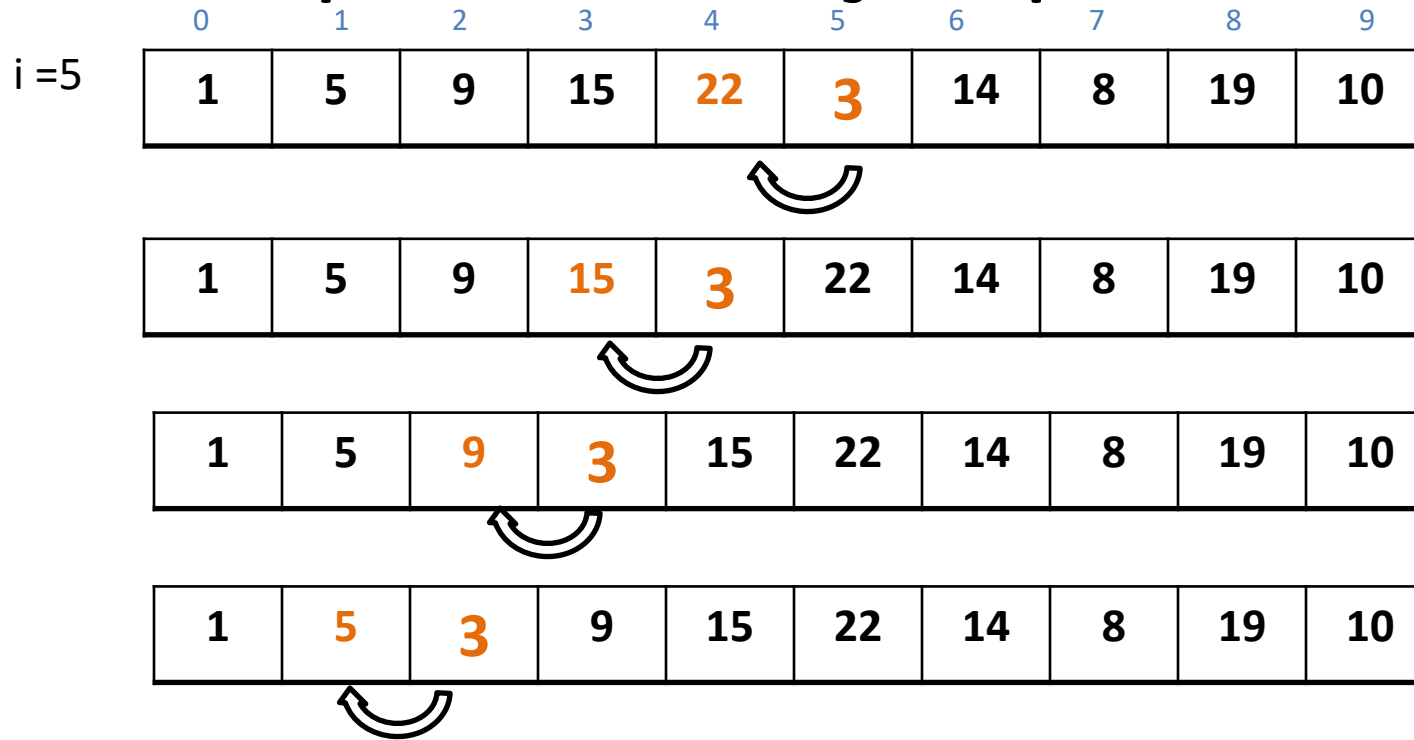
Exemplo ordenação por inserção



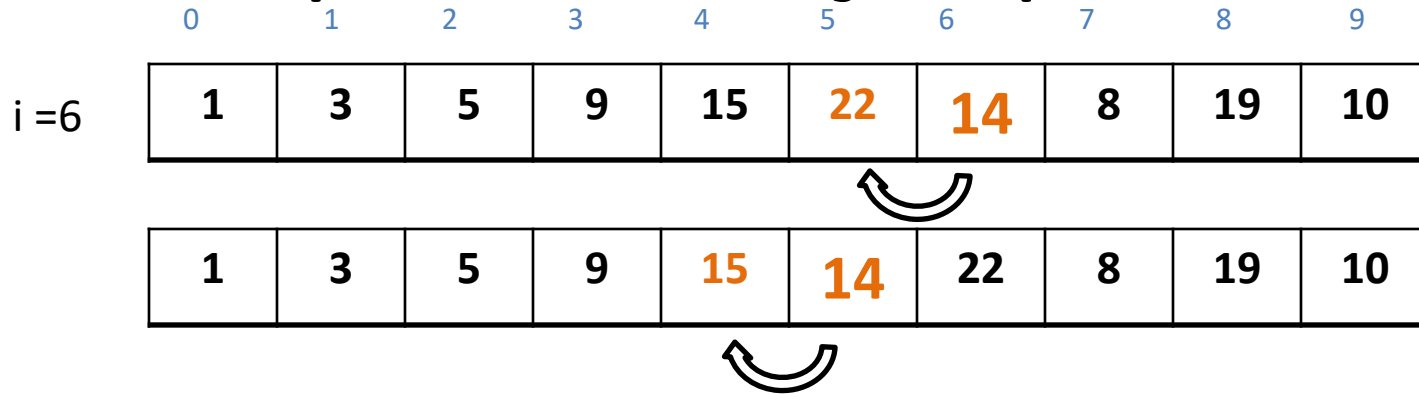
Exemplo ordenação por inserção



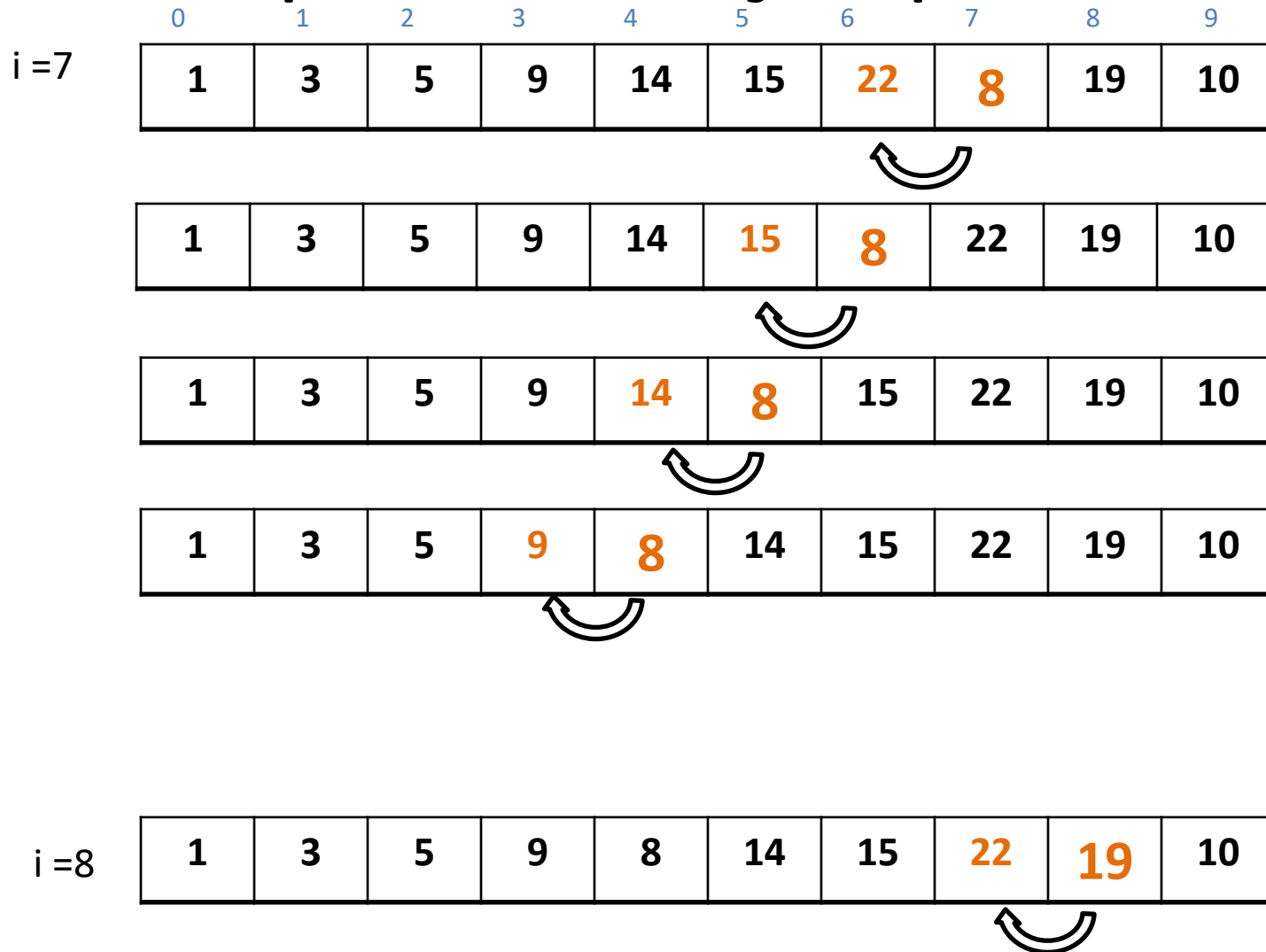
Exemplo ordenação por inserção



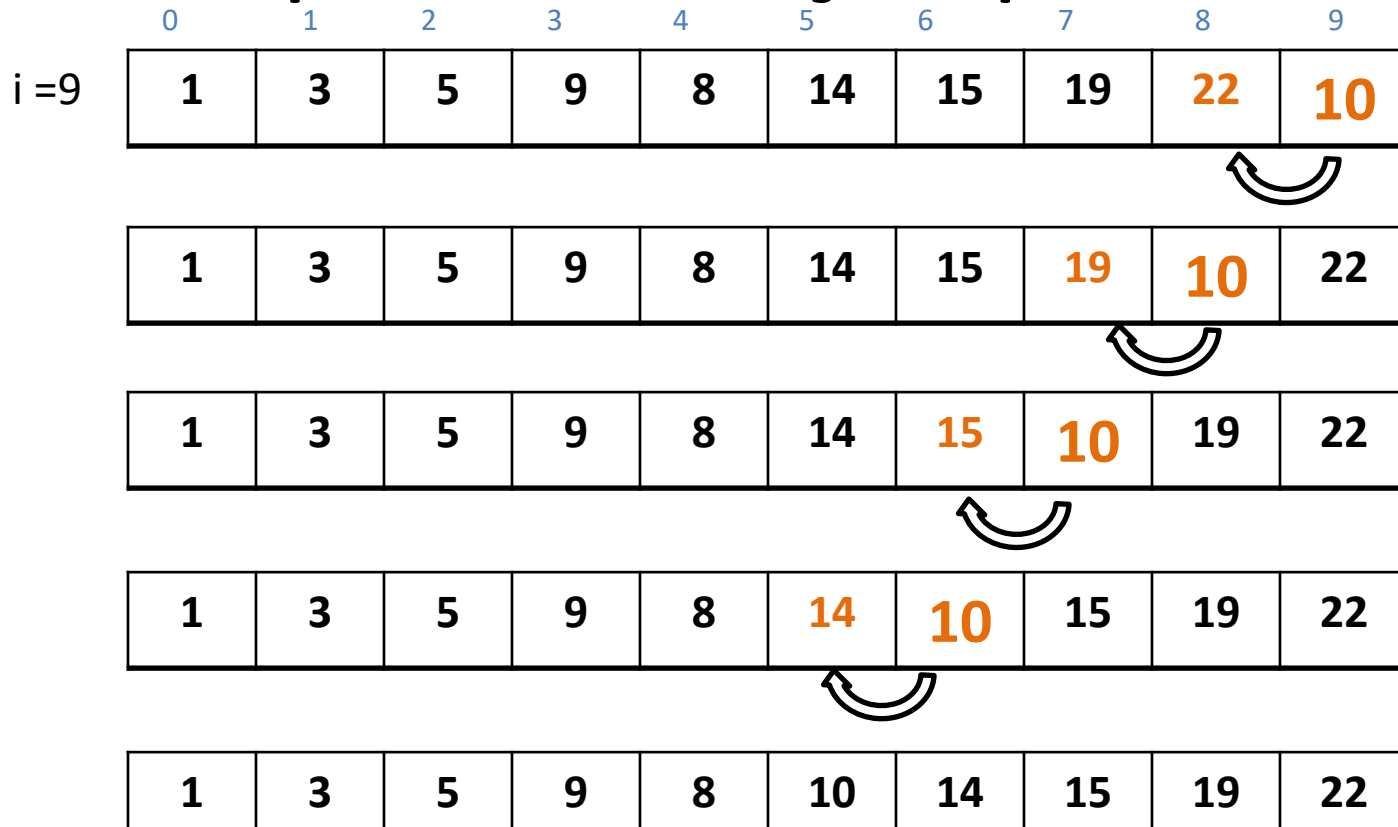
Exemplo ordenação por inserção



Exemplo ordenação por inserção



Exemplo ordenação por inserção



Algoritmo inserção

```
procedure Insercao (var A: Vetor; var n: Indice);  
var i, j: Indice;  
    x : Item;  
begin  
  for i := 2 to n do  
    begin  
      x := A[i];  
      j := i - 1;  
      A[0] := x; { sentinela }  
      while x.Chave < A[j].Chave do  
        begin  
          A[j + 1] := A[j];  
          j := j - 1;  
        end;  
      A[j + 1] := x;  
    end;  
  end;  
end;
```

$O(n)$

$O(n)$

Total = $O(n^2)$

Análise

1. Qual o número de comparações entre chaves e movimentações no algoritmo de **ordenação por inserção**?

melhor caso : $C(n) = (1 + 1 + \dots + 1) = n - 1 = O(n)$

pior caso : $C(n) = (2 + 3 + \dots + n) = n^2/2 + n/2 - 1 = O(n^2)$

caso médio : $C(n) = 1/2(3 + 4 + \dots + n + 1) = n^2/4 + 3n/4 - 1 = O(n^2)$

Número de movimentos:

Melhor caso: $M(n) = (3 + 3 + \dots + 3) = 3(n - 1)$

pior caso : $M(n) = (4 + 5 + \dots + n + 2) = n^2/2 + 5n/2 - 3$

caso médio : $M(n) = 1/2(5 + 6 + \dots + n + 3) = n^2/4 + 11n/4 - 3$

Inserção (A, n)

Para $i=2$ até n faça

$x \leftarrow A[i];$

$j \leftarrow i - 1;$

$A[0] \leftarrow x;$

 Enquanto $x < A[j]$ faça

$A[j+1] \leftarrow A[j];$

$j \leftarrow j - 1;$

 fim-enquanto;

$A[j+1] \leftarrow x;$

fim-para.

2. O algoritmo de ordenação por inserção é estável?

O algoritmo de ordenação por inserção **é estável**.

3. Quais as vantagens e desvantagens da ordenação por inserção?

Vantagens: É um bom método quando se deseja adicionar uns poucos itens a um arquivo ordenado, pois o custo é linear.

Desvantagens: quadrático

Algoritmos de ordenação em memória interna

- **Quadráticos $O(n^2)$:**
 - Ordenação por Seleção (Selection Sort)
 - Ordenação por Inserção (Insertion Sort)
 - Ordenação por Bolha (BubbleSort)
- **$O(n \log n)$:**
 - Quicksort
 - Heapsort
 - Shellsort
- **Lineares $O(n)$:**
 - Ordenação por contagem
 - Radix sort
 - Bucket sort

Ordenação por bolha

Algoritmo:

- Funciona permutando repetidamente elementos adjacentes que estão fora de ordem.

$O(n)$ { $O(n)$ {

```
BUBBLESORT(A)
1 for  $i \leftarrow 1$  to comprimento[A]
2   do for  $j \leftarrow$  comprimento[A] downto  $i + 1$ 
3     do if  $A[j] < A[j - 1]$ 
4       then trocar  $A[j] \leftrightarrow A[j - 1]$ 
```

Total = $O(n^2)$

i=0

j

Exemplo ordenação por bolha

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|----|----|---|---|----|----|----|----|----|----|
| 9 | 22 | 5 | 9 | 1 | 15 | 3 | 14 | 8 | 19 | 10 |
| 8 | 22 | 5 | 9 | 1 | 15 | 3 | 14 | 8 | 10 | 19 |
| 7 | 22 | 5 | 9 | 1 | 15 | 3 | 14 | 8 | 10 | 19 |
| 6 | 22 | 5 | 9 | 1 | 15 | 3 | 8 | 14 | 10 | 19 |
| 5 | 22 | 5 | 9 | 1 | 15 | 3 | 8 | 14 | 10 | 19 |
| 4 | 22 | 5 | 9 | 1 | 3 | 15 | 8 | 14 | 10 | 19 |
| 3 | 22 | 5 | 9 | 1 | 3 | 15 | 8 | 14 | 10 | 19 |
| 2 | 22 | 5 | 1 | 9 | 3 | 15 | 8 | 14 | 10 | 19 |
| 1 | 22 | 1 | 5 | 9 | 3 | 15 | 8 | 14 | 10 | 19 |
| | 1 | 22 | 5 | 9 | 3 | 15 | 8 | 14 | 10 | 19 |

i=1

j

Exemplo ordenação por bolha

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|----|----|---|---|----|----|----|----|----|
| 9 | 1 | 22 | 5 | 9 | 3 | 15 | 8 | 14 | 10 | 19 |
| 8 | 1 | 22 | 5 | 9 | 3 | 15 | 8 | 14 | 10 | 19 |
| 7 | 1 | 22 | 5 | 9 | 3 | 15 | 8 | 10 | 14 | 19 |
| 6 | 1 | 22 | 5 | 9 | 3 | 15 | 8 | 10 | 14 | 19 |
| 5 | 1 | 22 | 5 | 9 | 3 | 8 | 15 | 10 | 14 | 19 |
| 4 | 1 | 22 | 5 | 9 | 3 | 8 | 15 | 10 | 14 | 19 |
| 3 | 1 | 22 | 5 | 3 | 9 | 8 | 15 | 10 | 14 | 19 |
| 2 | 1 | 22 | 3 | 5 | 9 | 8 | 15 | 10 | 14 | 19 |
| | 1 | 3 | 22 | 5 | 9 | 8 | 15 | 10 | 14 | 19 |

i=2

Exemplo ordenação por bolha

| j | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|----|----|---|---|----|----|----|----|
| 9 | 1 | 3 | 22 | 5 | 9 | 8 | 15 | 10 | 14 | 19 |
| 8 | 1 | 3 | 22 | 5 | 9 | 8 | 15 | 10 | 14 | 19 |
| 7 | 1 | 3 | 22 | 5 | 9 | 8 | 15 | 10 | 14 | 19 |
| 6 | 1 | 3 | 22 | 5 | 9 | 8 | 10 | 15 | 14 | 19 |
| 5 | 1 | 3 | 22 | 5 | 9 | 8 | 10 | 15 | 14 | 19 |
| 4 | 1 | 3 | 22 | 5 | 8 | 9 | 10 | 15 | 14 | 19 |
| 3 | 1 | 3 | 22 | 5 | 8 | 9 | 10 | 15 | 14 | 19 |
| | 1 | 3 | 5 | 22 | 8 | 9 | 10 | 15 | 14 | 19 |

i=3

j

Exemplo ordenação por bolha

| | | | | | | | | | | |
|---|---|---|---|----|----|---|----|----|----|----|
| 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| | 1 | 3 | 5 | 22 | 8 | 9 | 10 | 15 | 14 | 19 |
| 8 | 1 | 3 | 5 | 22 | 8 | 9 | 10 | 15 | 14 | 19 |
| 7 | 1 | 3 | 5 | 22 | 8 | 9 | 10 | 14 | 15 | 19 |
| 6 | 1 | 3 | 5 | 22 | 8 | 9 | 10 | 14 | 15 | 19 |
| 5 | 1 | 3 | 5 | 22 | 8 | 9 | 10 | 14 | 15 | 19 |
| 4 | 1 | 3 | 5 | 22 | 8 | 9 | 10 | 14 | 15 | 19 |
| | 1 | 3 | 5 | 8 | 22 | 9 | 10 | 14 | 15 | 19 |

i = 4

Exemplo ordenação por bolha

| j | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|----|----|----|----|----|----|
| 9 | 1 | 3 | 5 | 8 | 22 | 9 | 10 | 14 | 15 | 19 |
| 8 | 1 | 3 | 5 | 8 | 22 | 9 | 10 | 14 | 15 | 19 |
| 7 | 1 | 3 | 5 | 8 | 22 | 9 | 10 | 14 | 15 | 19 |
| 6 | 1 | 3 | 5 | 8 | 22 | 9 | 10 | 14 | 15 | 19 |
| 5 | 1 | 3 | 5 | 8 | 22 | 9 | 10 | 14 | 15 | 19 |
| | 1 | 3 | 5 | 8 | 9 | 22 | 10 | 14 | 15 | 19 |

i=5

j

Exemplo ordenação por bolha

| | | | | | | | | | | |
|---|---|---|---|---|---|----|----|----|----|----|
| 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| | 1 | 3 | 5 | 8 | 9 | 22 | 10 | 14 | 15 | 19 |
| 8 | 1 | 3 | 5 | 8 | 9 | 22 | 10 | 14 | 15 | 19 |
| 7 | 1 | 3 | 5 | 8 | 9 | 22 | 10 | 14 | 15 | 19 |
| 6 | 1 | 3 | 5 | 8 | 9 | 22 | 10 | 14 | 15 | 19 |
| | 1 | 3 | 5 | 8 | 9 | 10 | 22 | 14 | 15 | 19 |

i = 6

Exemplo ordenação por bolha

j

9

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|----|----|----|----|----|
| 1 | 3 | 5 | 8 | 9 | 10 | 22 | 14 | 15 | 19 |

8

| | | | | | | | | | |
|---|---|---|---|---|----|----|----|----|----|
| 1 | 3 | 5 | 8 | 9 | 10 | 22 | 14 | 15 | 19 |
|---|---|---|---|---|----|----|----|----|----|

7

| | | | | | | | | | |
|---|---|---|---|---|----|----|----|----|----|
| 1 | 3 | 5 | 8 | 9 | 10 | 22 | 14 | 15 | 19 |
|---|---|---|---|---|----|----|----|----|----|

| | | | | | | | | | |
|---|---|---|---|---|----|----|----|----|----|
| 1 | 3 | 5 | 8 | 9 | 10 | 14 | 22 | 15 | 19 |
|---|---|---|---|---|----|----|----|----|----|

i=7

j

Exemplo ordenação por bolha

| | | | | | | | | | | |
|---|---|---|---|---|---|----|----|----|----|----|
| 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| | 1 | 3 | 5 | 8 | 9 | 10 | 14 | 22 | 15 | 19 |
| 8 | 1 | 3 | 5 | 8 | 9 | 10 | 14 | 22 | 15 | 19 |
| | 1 | 3 | 5 | 8 | 9 | 10 | 14 | 15 | 22 | 19 |

i=8

j

9

Exemplo ordenação por bolha

| | | | | | | | | | |
|---|---|---|---|---|----|----|----|----|----|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 1 | 3 | 5 | 8 | 9 | 10 | 14 | 15 | 22 | 19 |
| 1 | 3 | 5 | 8 | 9 | 10 | 14 | 15 | 19 | 22 |

Análise

1. Qual o número de comparações entre chaves e movimentações no algoritmo de **ordenação por bolha**?

melhor caso : $C(n) = O(n)$

pior caso : $C(n) = O(n^2)$

caso médio : $C(n) = O(n^2)$

BUBBLESORT(A)

```
1 for i ← 1 to comprimento[A]
2   do for j ← comprimento[A] downto i + 1
3     do if A[j] < A[j - 1]
4       then trocar A[j] ↔ A[j - 1]
```

Número de movimentos é similar ao de comparações.

2. O algoritmo de ordenação por inserção é estável?

O algoritmo de ordenação por inserção **é estável**.

3. Quais as vantagens e desvantagens da ordenação por inserção?

Vantagens: É um bom método quando se deseja adicionar uns poucos itens a um arquivo ordenado, pois o custo é linear.

Desvantagens: quadrático

Prática

- Ver funcionamento do algoritmo **select**, **insert** e **bubble** no site: <https://visualgo.net/sorting>



**Quem come do fruto do conhecimento, é
sempre expulso de algum paraíso.**

Melanie Klein

