



Algoritmo e Estrutura de Dados II

CTCO02

Ordenação por Seleção
Ordenação por Inserção

Vanessa Souza



Ordenação



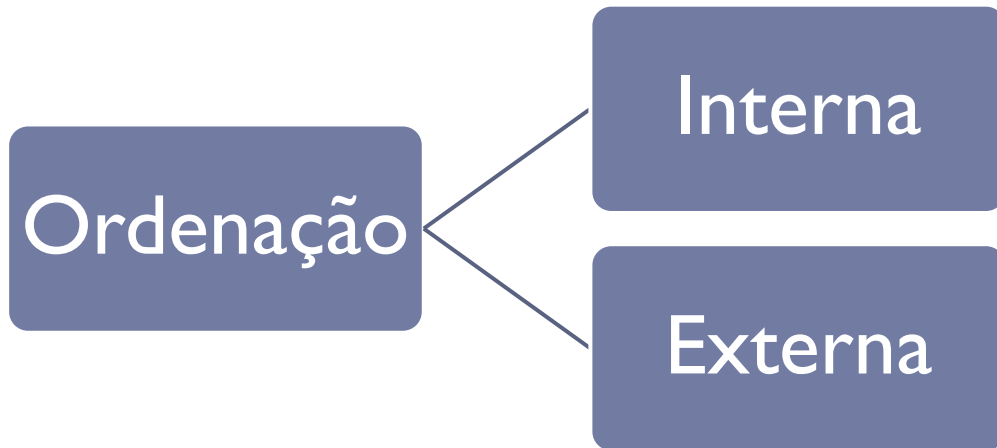
Ordenação

- ▶ Processo bastante utilizado na computação de uma estrutura de dados
- ▶ Ordenar significa colocar em ordem, segundo algum critério
- ▶ Alterar a ordem na qual os elementos de uma estrutura de dados aparece nessa estrutura
 - ▶ Rearranjar a estrutura



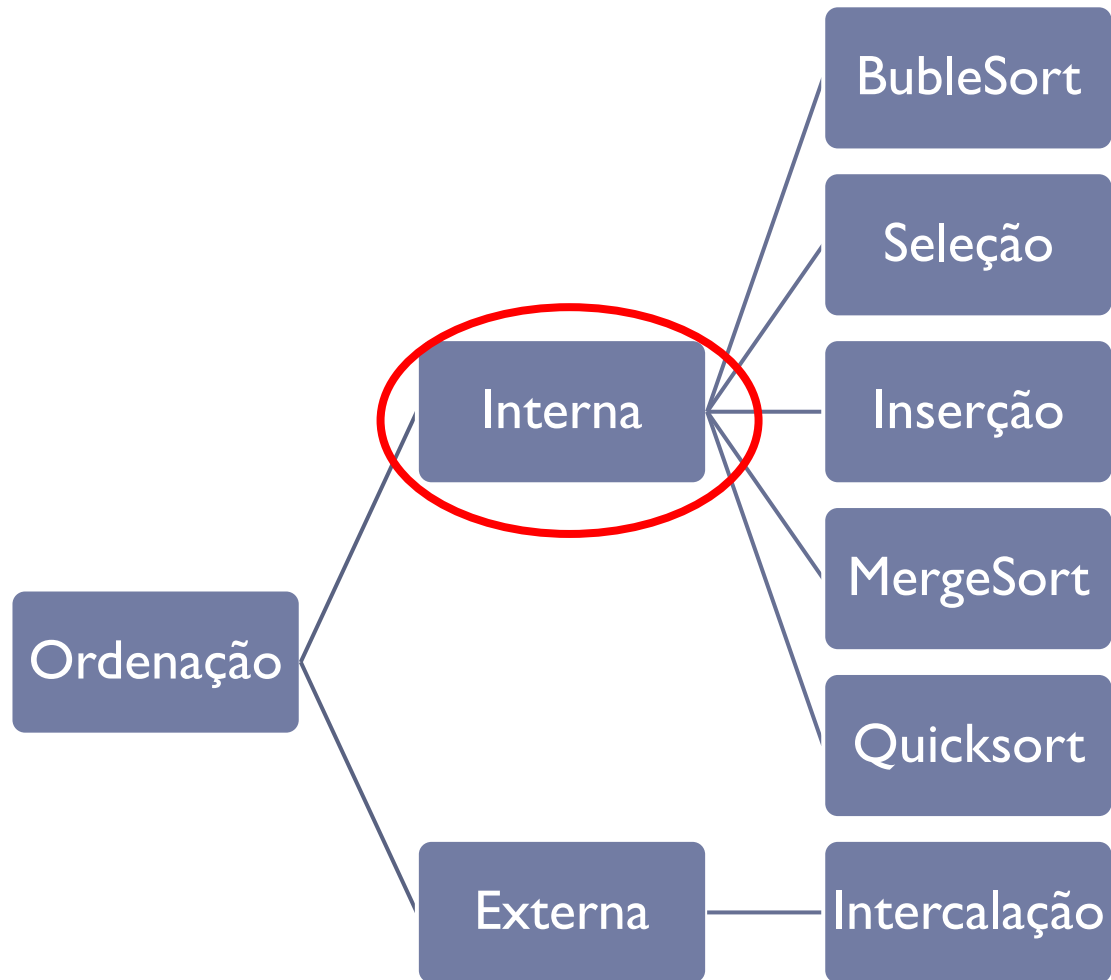


Classificação dos Métodos de Ordenação



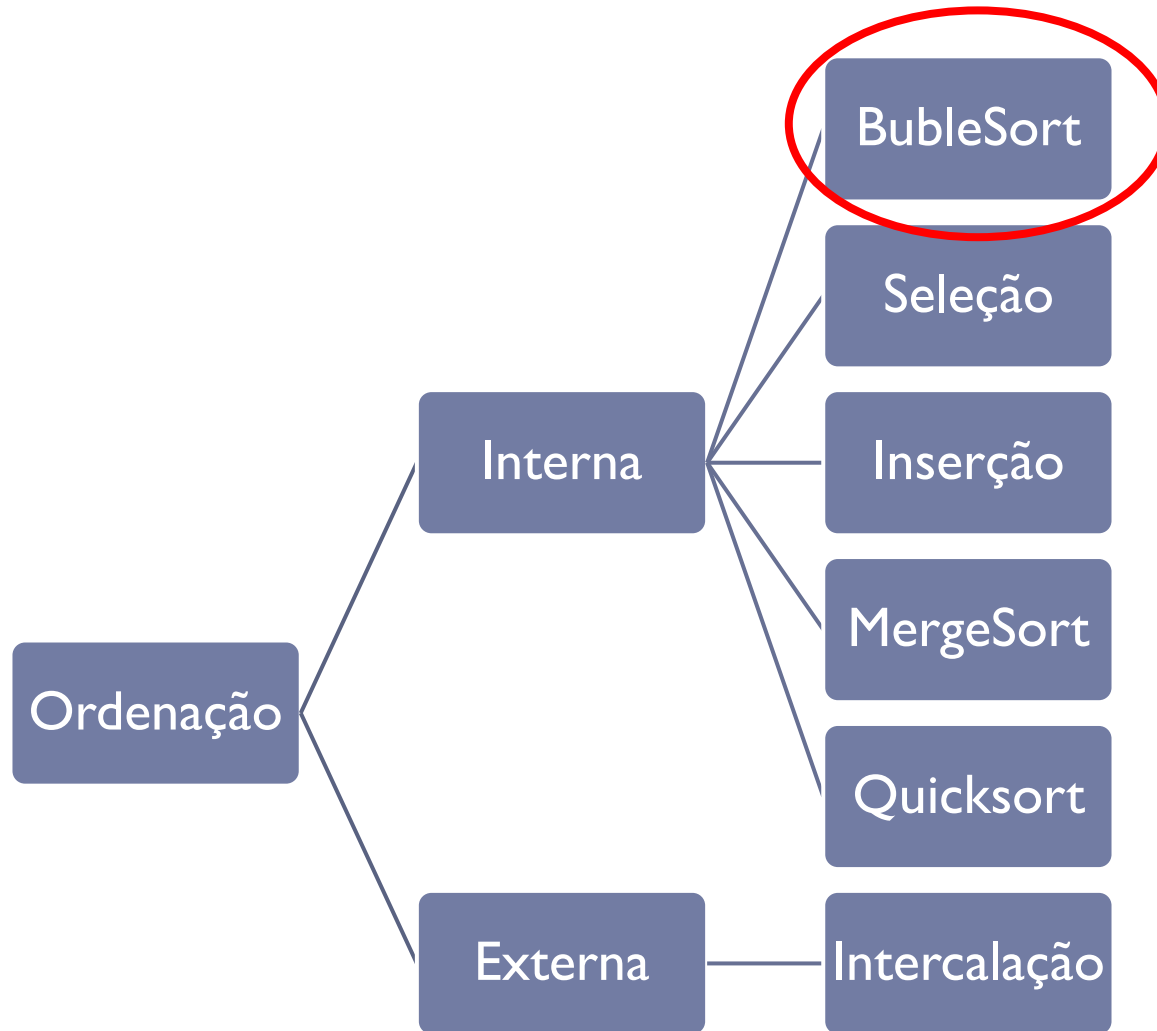


Classificação dos Métodos de Ordenação



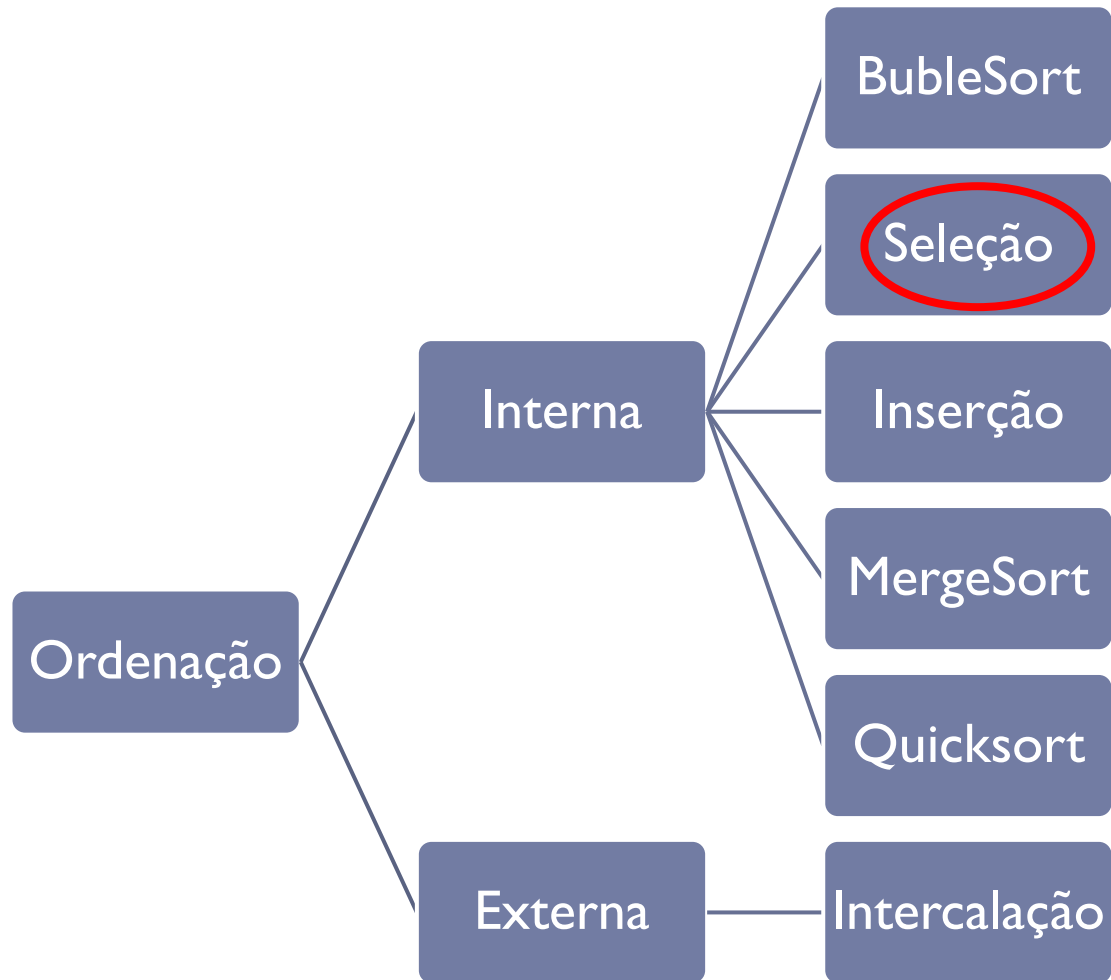


Classificação dos Métodos de Ordenação





Classificação dos Métodos de Ordenação





Seleção



Ordenação por Seleção

► Ideia

- Este algoritmo usa um marcador para dividir as partes ordenada (à esquerda) e desordenada (à direita) do vetor.
- Procura-se na parte desordenada pelo menor elemento e troca-se esse elemento com o elemento sob o marcador.
- Em seguida, avança-se o marcador. O processo se repete até que exista apenas um elemento a partir do marcador.





Ordenação por Seleção





Ordenação por Seleção



Ordenado Desordenado

Procura pelo menor elemento





Ordenação por Seleção



Ordenado

Desordenado

Troca com a posição do marcador





Ordenação por Seleção



Ordenado

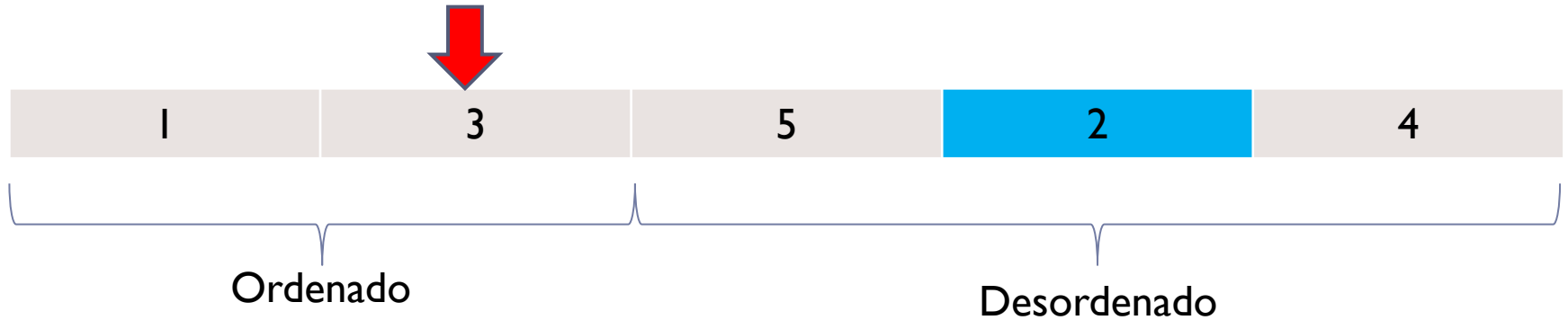
Desordenado

Anda com o marcador





Ordenação por Seleção

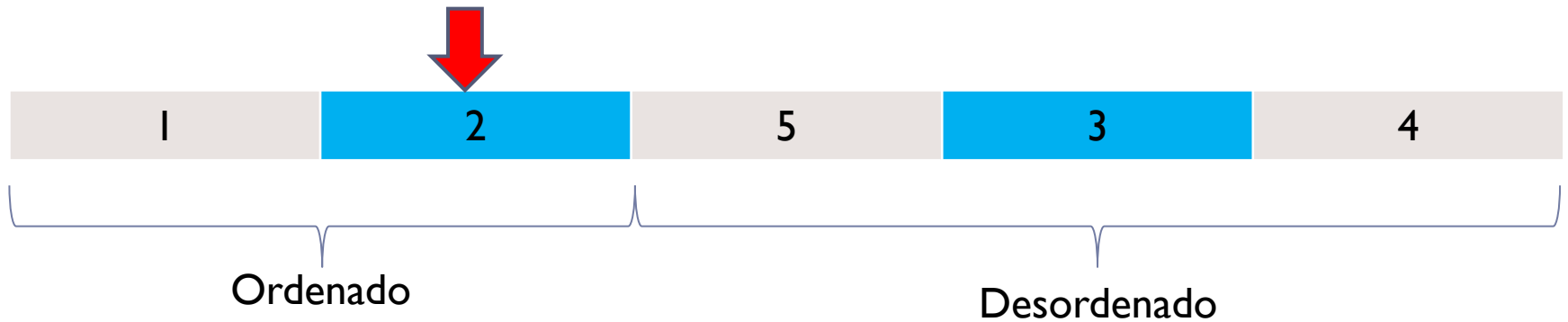


Repete o processo





Ordenação por Seleção



Repete o processo





Ordenação por Seleção



Ordenado

Desordenado

Repete o processo





Ordenação por Seleção



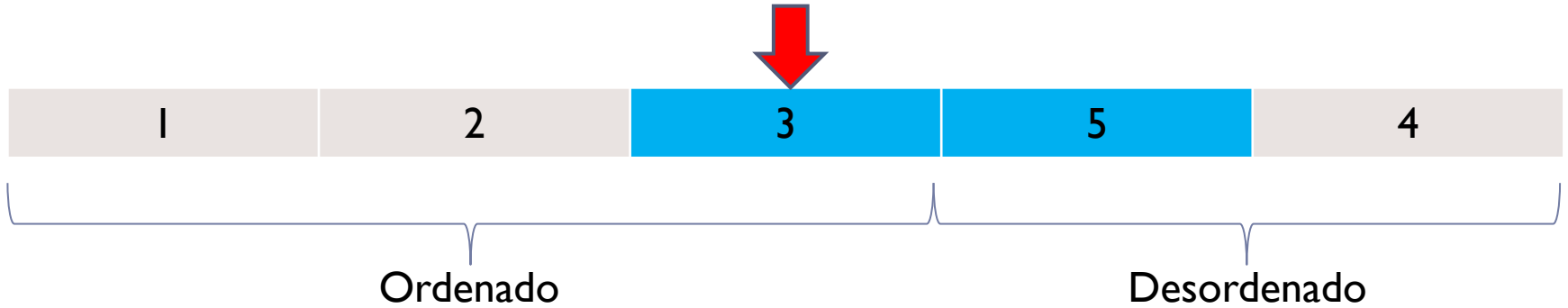
Ordenado

Desordenado





Ordenação por Seleção

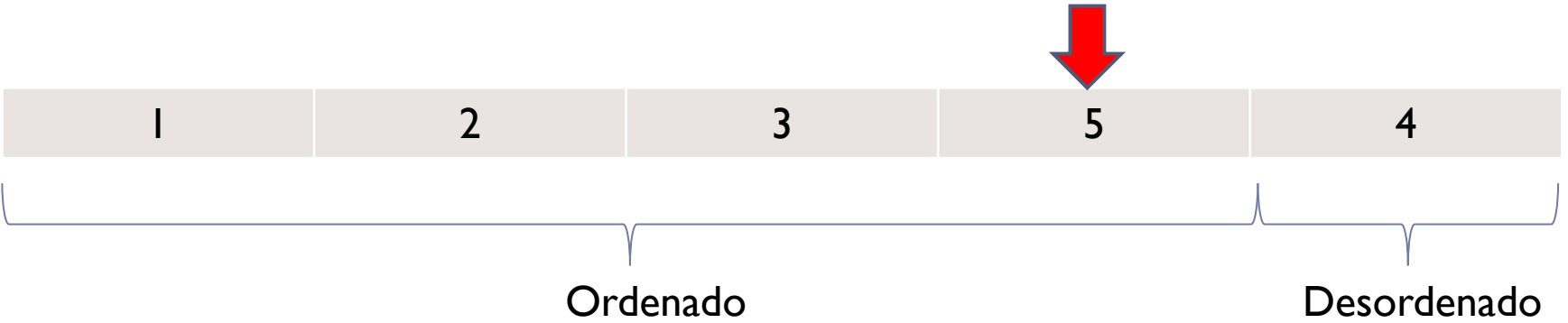


Repete o processo





Ordenação por Seleção

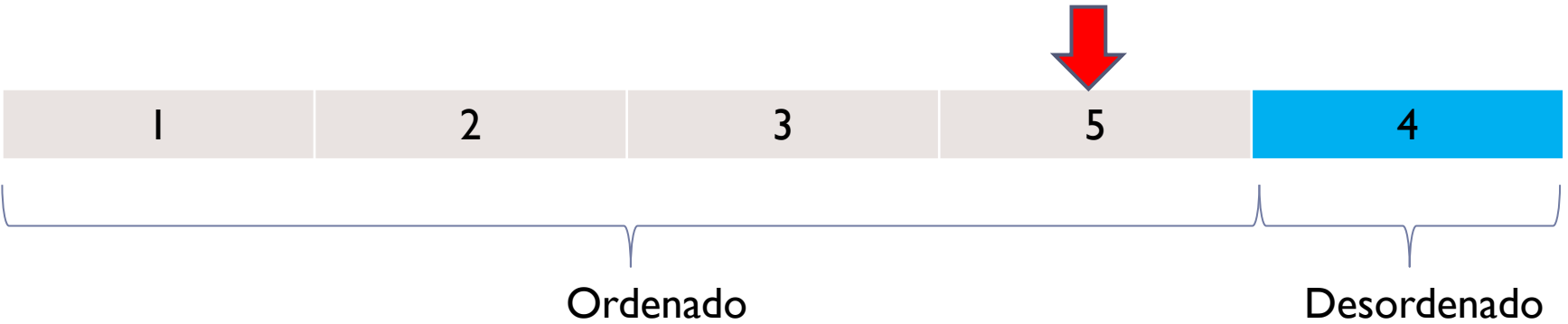


Repete o processo





Ordenação por Seleção

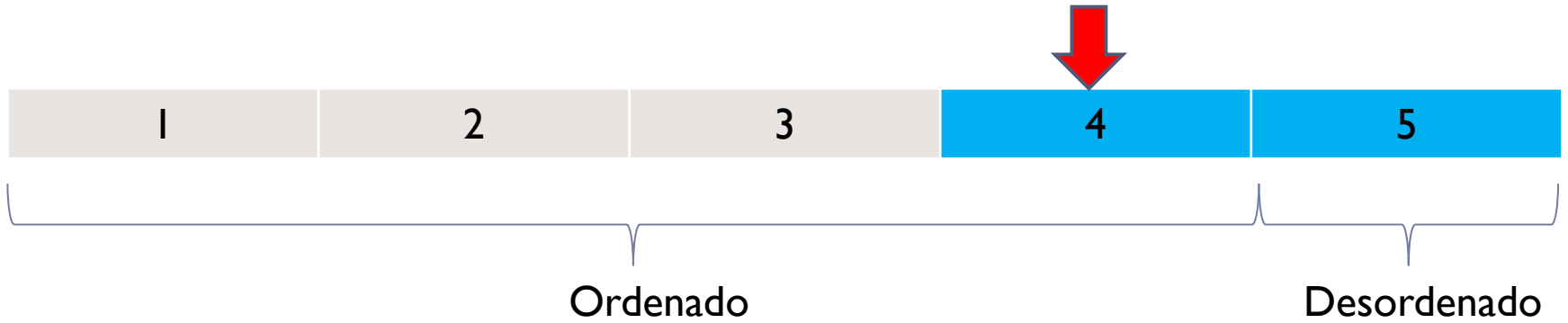


Repete o processo





Ordenação por Seleção



Repete o processo





Ordenação por Seleção





Ordenação por Seleção

► Algoritmo

Algorithm 1 Ordenação por Seleção

```
procedure SELECAO(V, tamVet) ▷ V é um vetor numérico
    marcador  $\leftarrow$  0
    menor  $\leftarrow$  0
    while (marcador < tamVet - 1) do
        menor  $\leftarrow$  índice do menor elemento da parte desordenada do vetor
        if (vet[menor] < vet[marcador]) then
            troque vet[marcador] com vet[menor]
        end if
        marcador  $\leftarrow$  marcador + 1
    end while
end procedure
```





Ordenação por Seleção

▶ Algoritmo

Algorithm 1 Ordenação por Seleção

procedure SELECAO(V , tamVet)

▷ V é um vetor numérico

 marcador $\leftarrow 0$

 menor $\leftarrow 0$

 while ($\text{marcador} < \text{tamVet} - 1$) do

 → menor \leftarrow índice do menor elemento da parte desordenada do vetor

 if ($\text{vet}[\text{menor}] < \text{vet}[\text{marcador}]$) then

 → troque $\text{vet}[\text{marcador}]$ com $\text{vet}[\text{menor}]$

 end if

 marcador \leftarrow marcador + 1

 end while

end procedure





Ordenação por Seleção

- ▶ Exercício
 - ▶ Ordenar por seleção os seguintes vetores
 - ▶ Contar
 - ▶ Número de comparações
 - ▶ Número de trocas

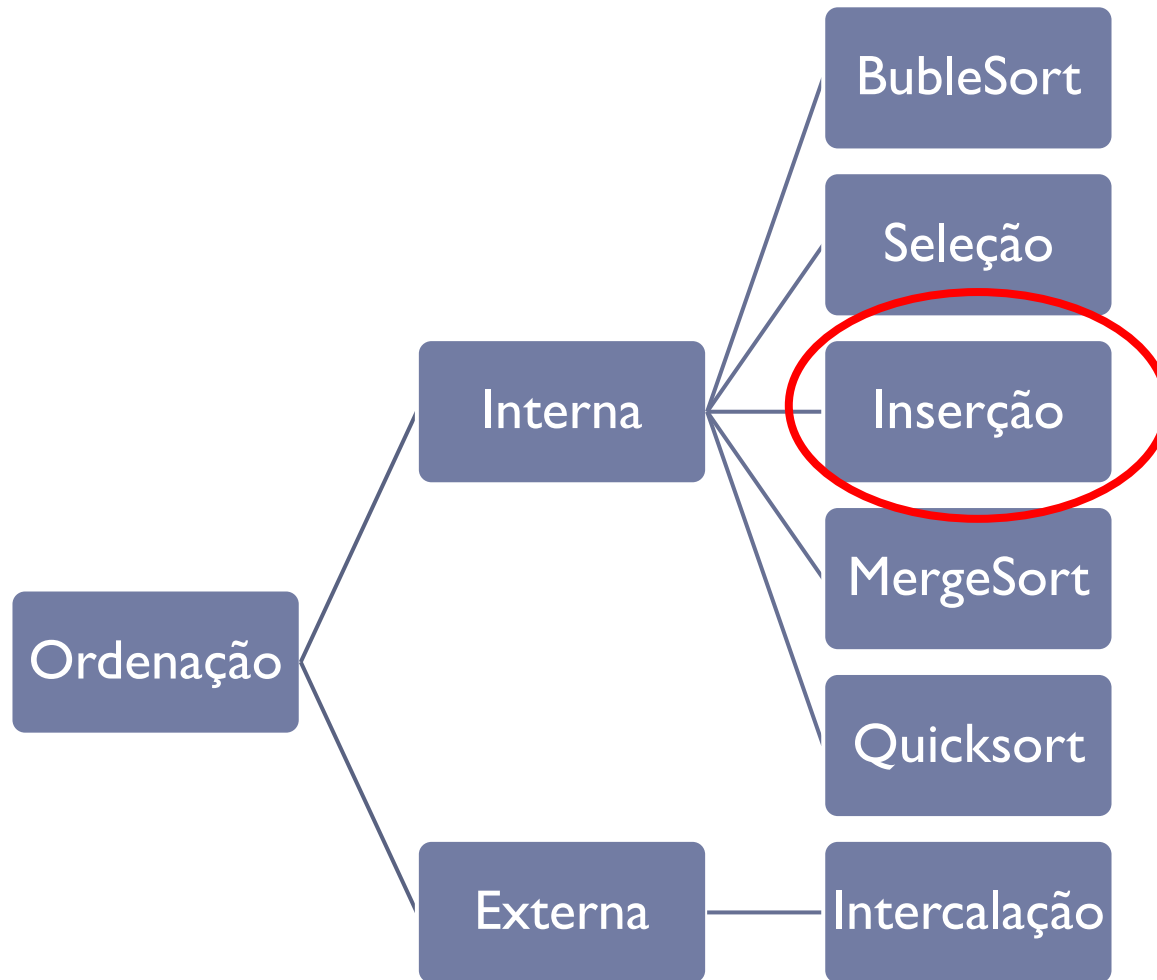
5	4	3	2	1
1	2	3	5	4



Ordenação por Inserção



Classificação dos Métodos de Ordenação





InsertionSort

- ▶ Ideia: Também usa marcador mas, inicialmente, marcador = 1. Seja x o primeiro elemento da parte desordenada. Troca-se x de posição com os elementos que aparecem à esquerda até que x esteja em sua posição correta, e avança-se o marcador.
- ▶ O processo se repete até que a parte desordenada do vetor esteja vazia.
- ▶ Insere $\text{vet}[\text{marcador}]$ na posição correta da parte ordenada do vetor





InsertionSort



Ordenado

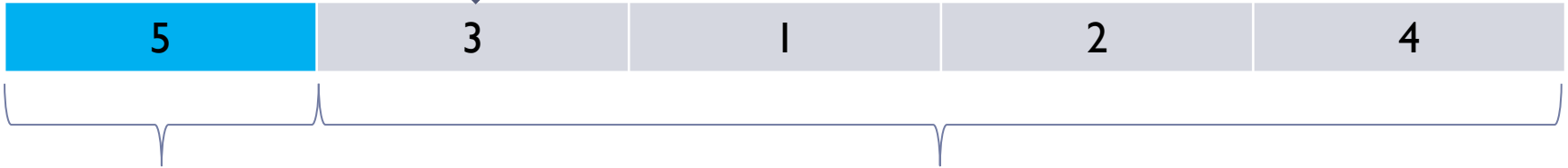
Desordenado

Procura na parte ordenada do vetor, o local correto de 3





InsertionSort



Ordenado

Desordenado

Procura na parte ordenada do vetor, o local correto de 3





InsertionSort



Ordenado

Desordenado

Inserir o 3 no lugar do 5
Andar com o marcador





InsertionSort



Ordenado

Desordenado

Procura na parte ordenada do vetor, o local correto de 1





InsertionSort



Ordenado

Desordenado

Local correto de 1 : posição 0





InsertionSort



Ordenado

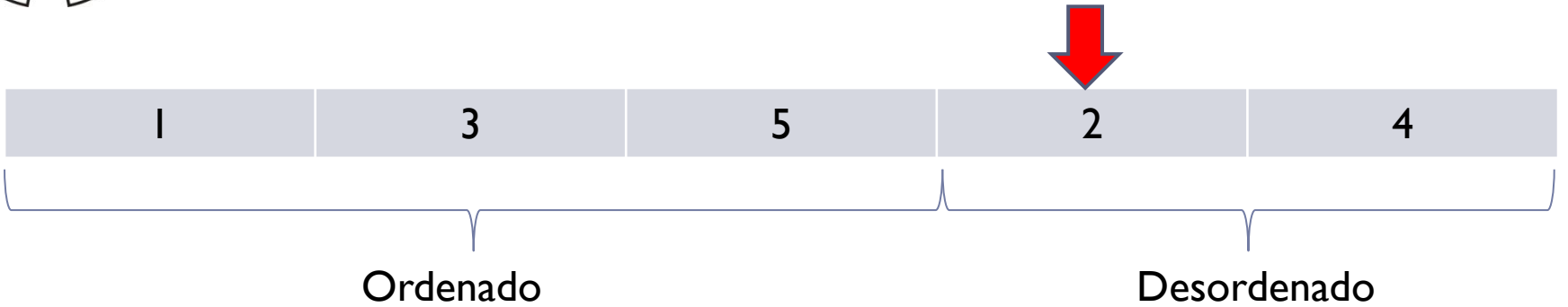
Desordenado

Inserir o 1 no seu local correto, movimentando as demais posições





InsertionSort

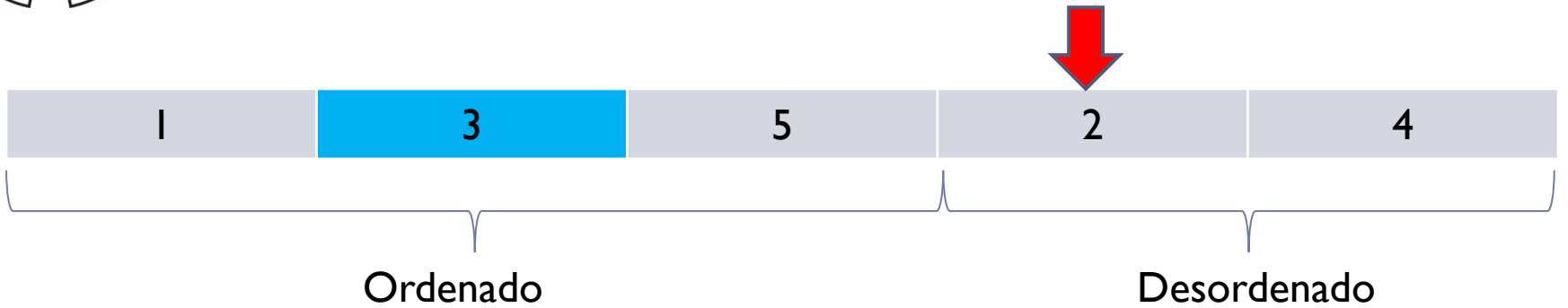


Repete o processo





InsertionSort

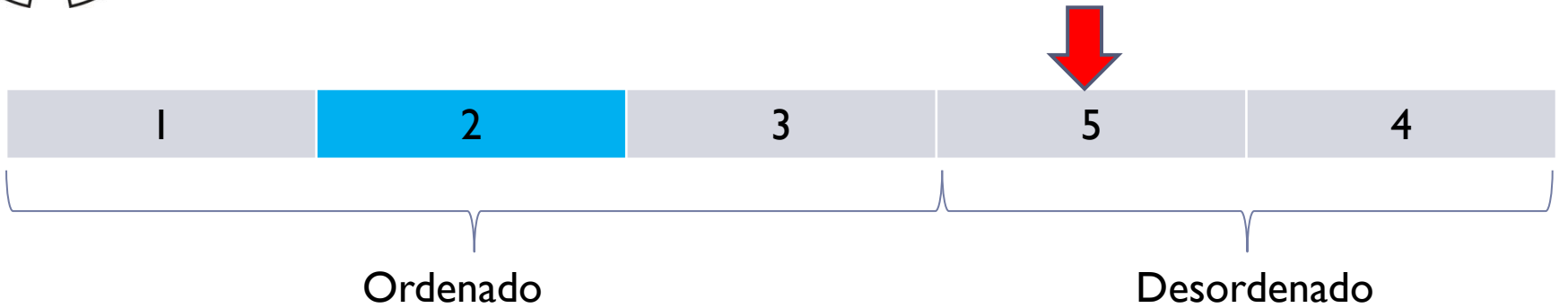


Repete o processo





InsertionSort

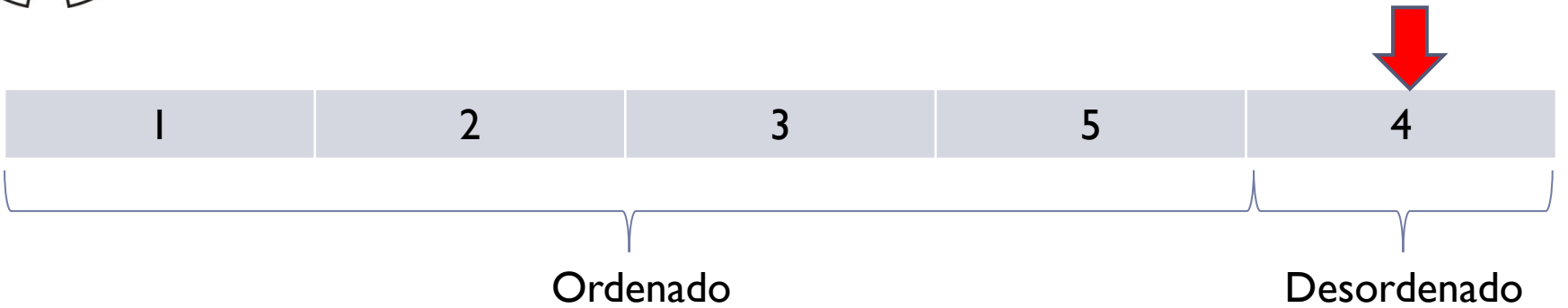


Repete o processo





InsertionSort



Repete o processo





InsertionSort

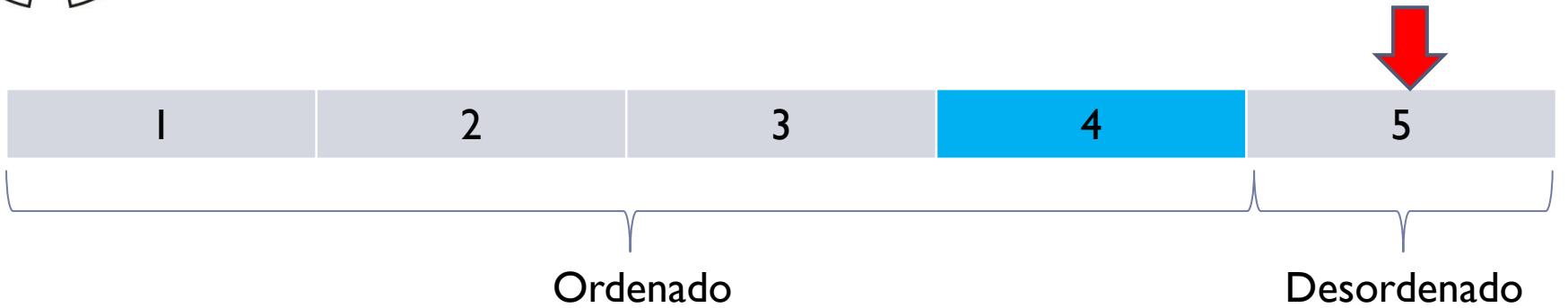


Repete o processo





InsertionSort



Repete o processo





InsertionSort



Ordenado

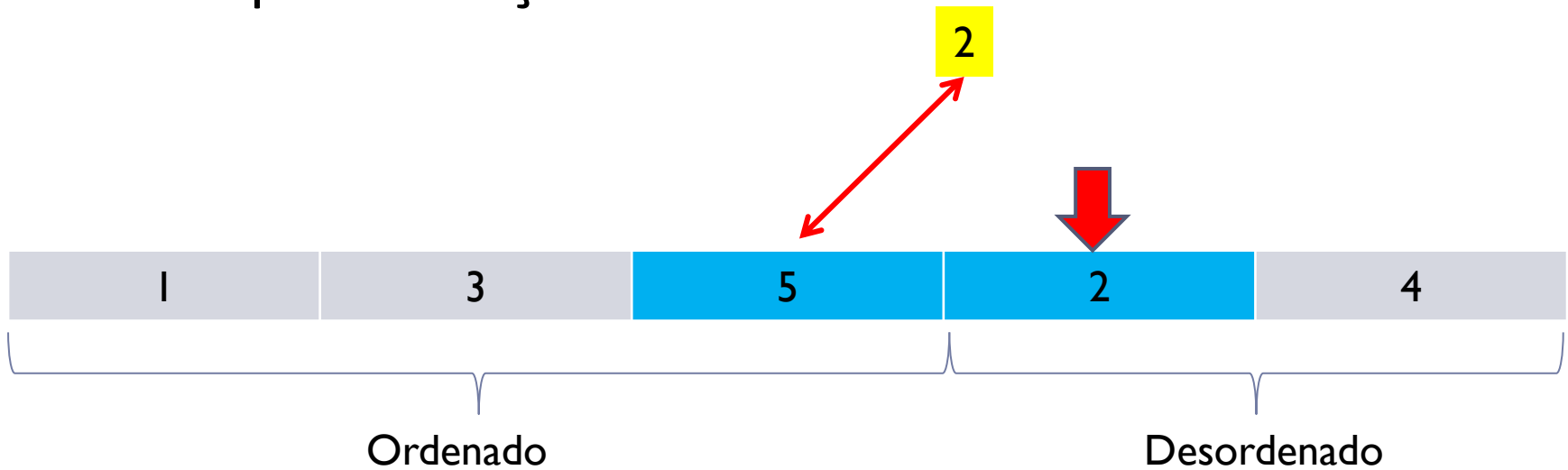
Vetor ordenado





InsertionSort

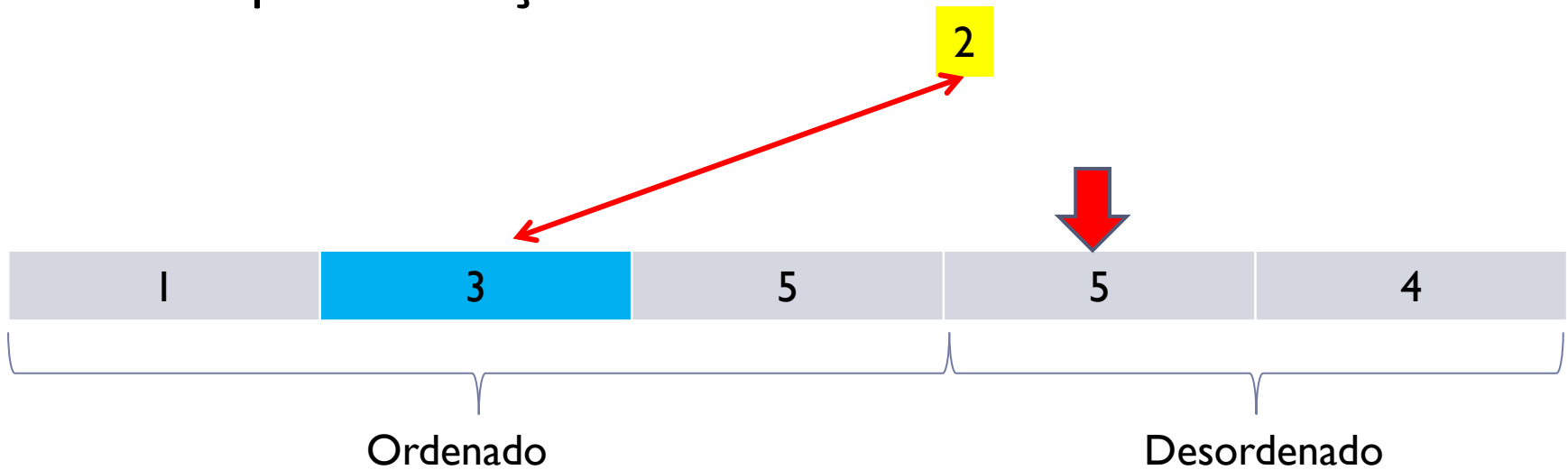
► Na implementação...





InsertionSort

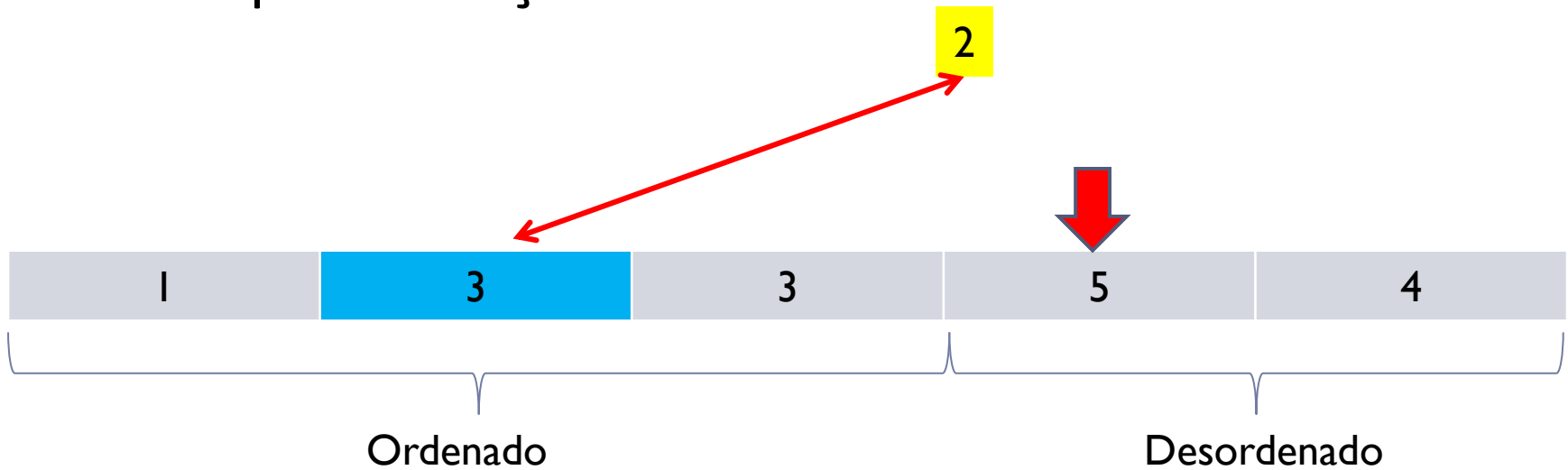
► Na implementação...





InsertionSort

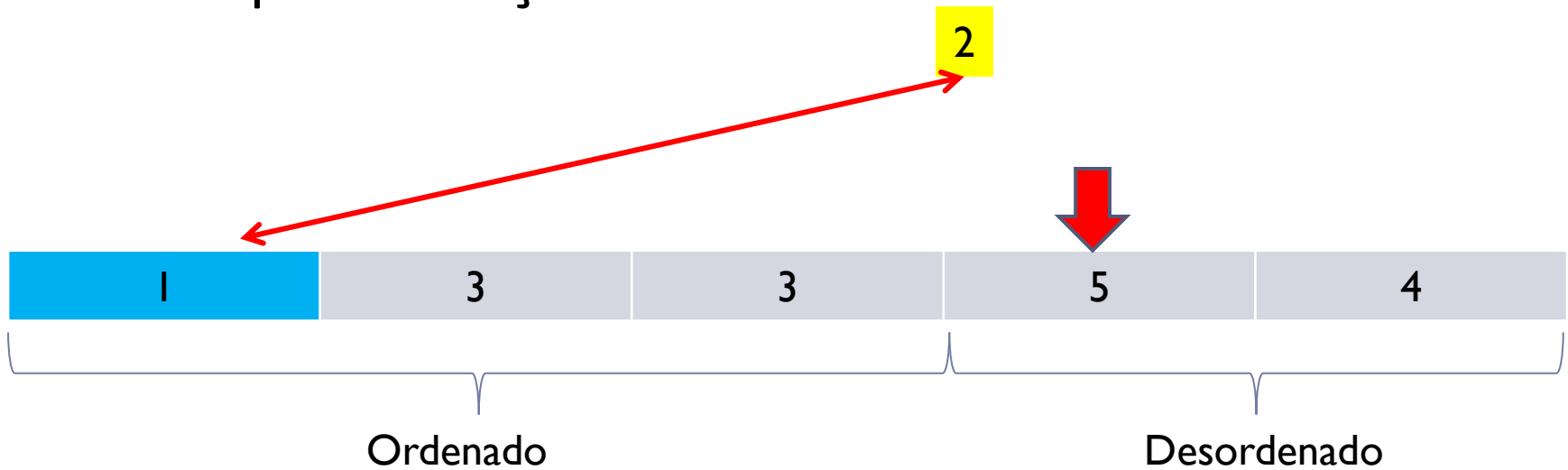
► Na implementação...





InsertionSort

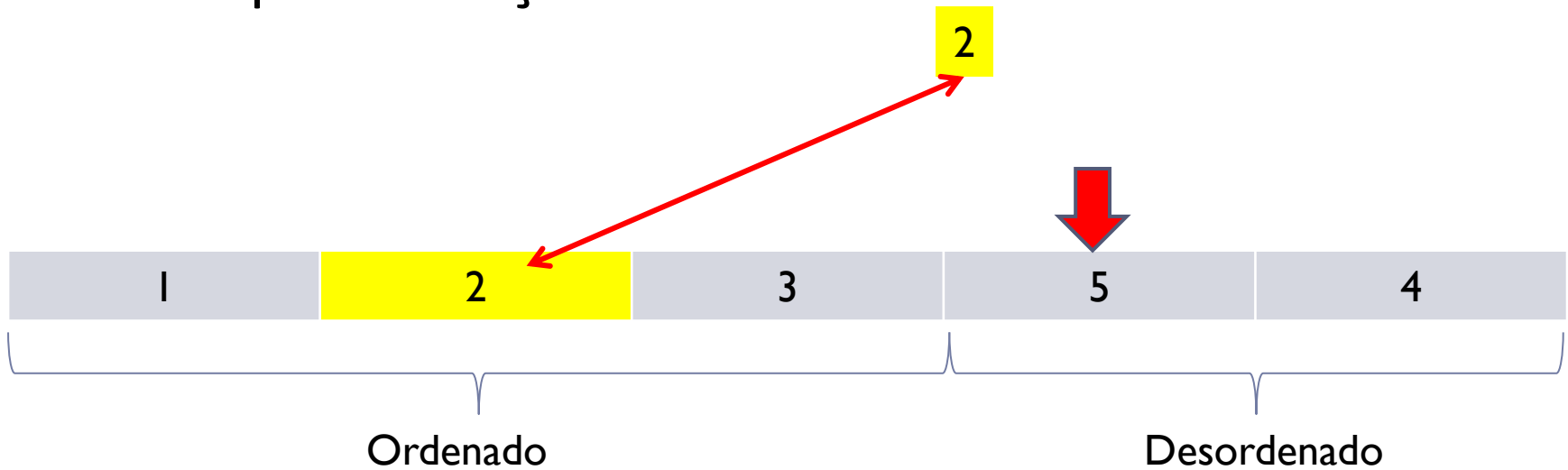
► Na implementação...





InsertionSort

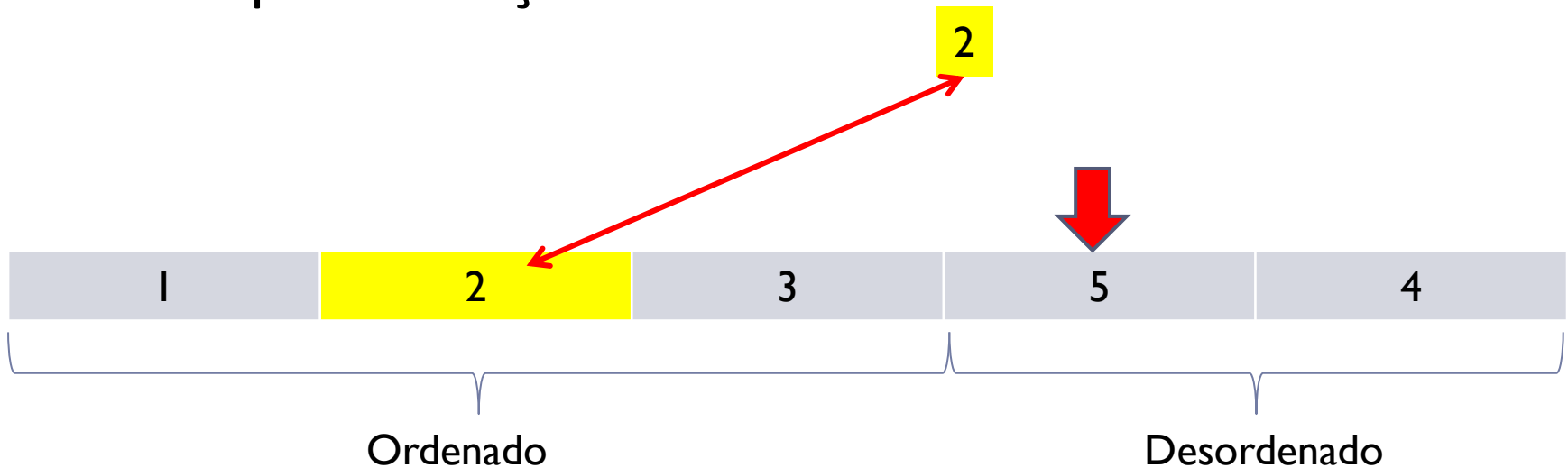
► Na implementação...





InsertionSort

► Na implementação...



Condição de parada : enquanto $\text{pos} \geq 0$ e enquanto $\text{aux} < \text{vet}[\text{pos}]$
pos : variável inicializada em $\text{marcador} - 1$;
aux : variável que guarda $\text{vet}[\text{marcador}]$





InsertionSort

► Algoritmo

Algorithm 1 Ordenação por Inserção

```
procedure INSECAO(V, tamVet) ▷ V é um vetor numérico
  for (marcador = 1; marcador < tamVet) do
    pos ← marcador - 1
    aux ← V[marcador]

    ...Inserindo V[marcador] na parte ordenada...
    while (pos ≥ 0) ∧ (aux < V[pos]) do
      V[pos + 1] ← V[pos]
      pos ← pos - 1
    end while
    V[pos + 1] ← aux

  end for
end procedure
```





Ordenação por Inserção

▶ Exercício

- ▶ Ordenar por inserção os seguintes vetores
- ▶ Contar
 - ▶ Número de comparações
 - ▶ Número de trocas

5	4	3	2	1
---	---	---	---	---

1	2	3	5	4
---	---	---	---	---





InsertionSort

- ▶ Complexidade assintótica do algoritmo de Inserção
 - ▶ Qual o pior caso?

