

Pesquisa e Ordenação de Dados

Unidade 2.2:

Selection Sort



Selection Sort

- Consiste em **selecionar o menor** valor e trocá-lo com o elemento que está na primeira posição; em seguida, o segundo menor elemento é trocado com o que se encontra na segunda posição, e assim sucessivamente, até que reste apenas 1 elemento.
- O algoritmo divide a lista em:
 - Parte ordenada, à esquerda (inicialmente vazia)
 - Parte não ordenada, à direita (inicialmente contém todos os elementos)

Selection Sort

- São realizadas **$n-1$** iterações
- Cada iteração:
 - Seleciona o menor elemento da parte não ordenada (direita) e troca-o de posição com o primeiro elemento desta mesma parte.
 - Este elemento é então incorporado à parte ordenada (esquerda). Com isso, a parte ordenada aumenta e a parte não ordenada diminui.
- Invariantes:
 - Valores já incorporados à parte ordenada não sofrem mais qualquer movimentação
 - Nenhum elemento da parte não ordenada é menor que qualquer elemento da parte ordenada

Selection Sort

Exemplo (1)

Iteração 1



Selection Sort

Exemplo (2)

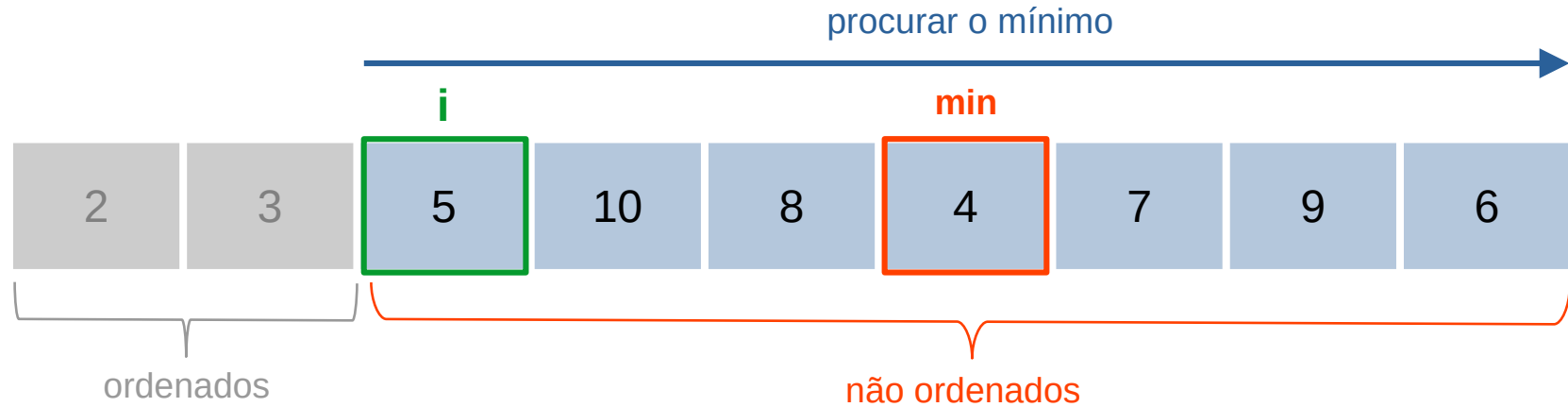
Iteração 2



Selection Sort

Exemplo (3)

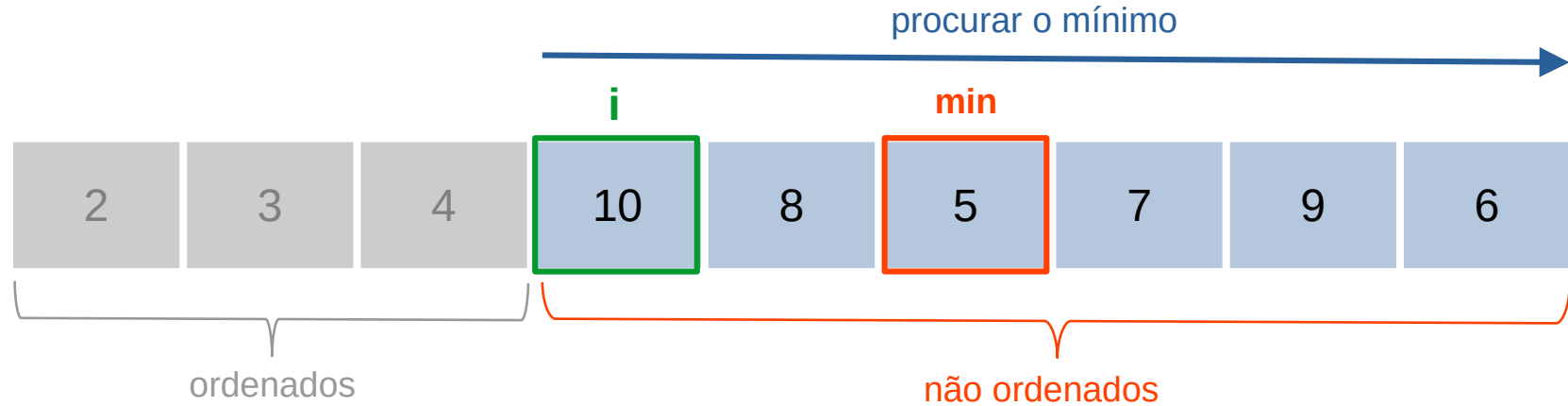
Iteração 3



Selection Sort

Exemplo (4)

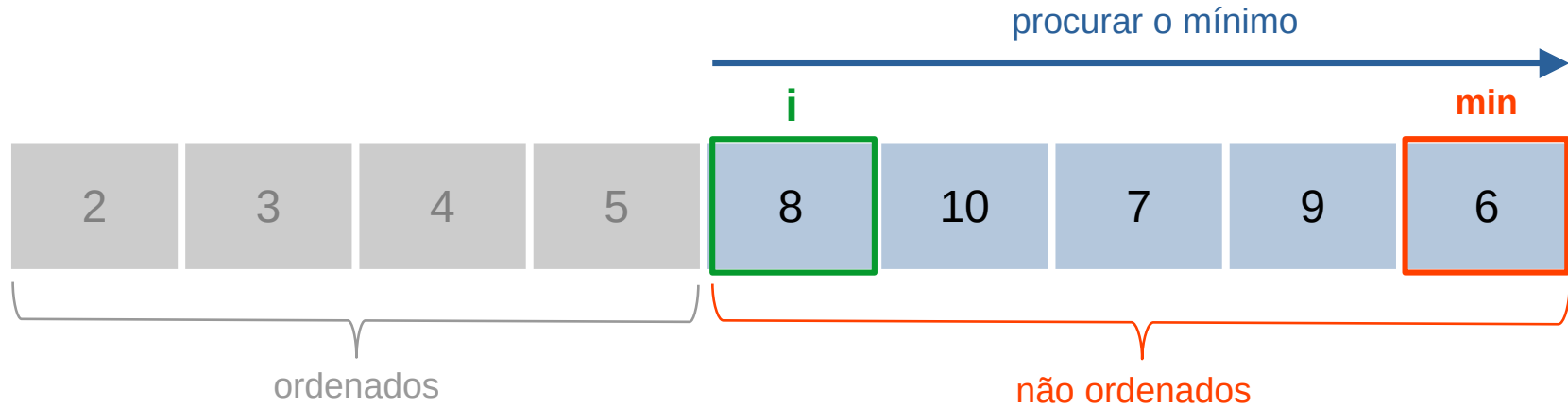
Iteração 4



Selection Sort

Exemplo (5)

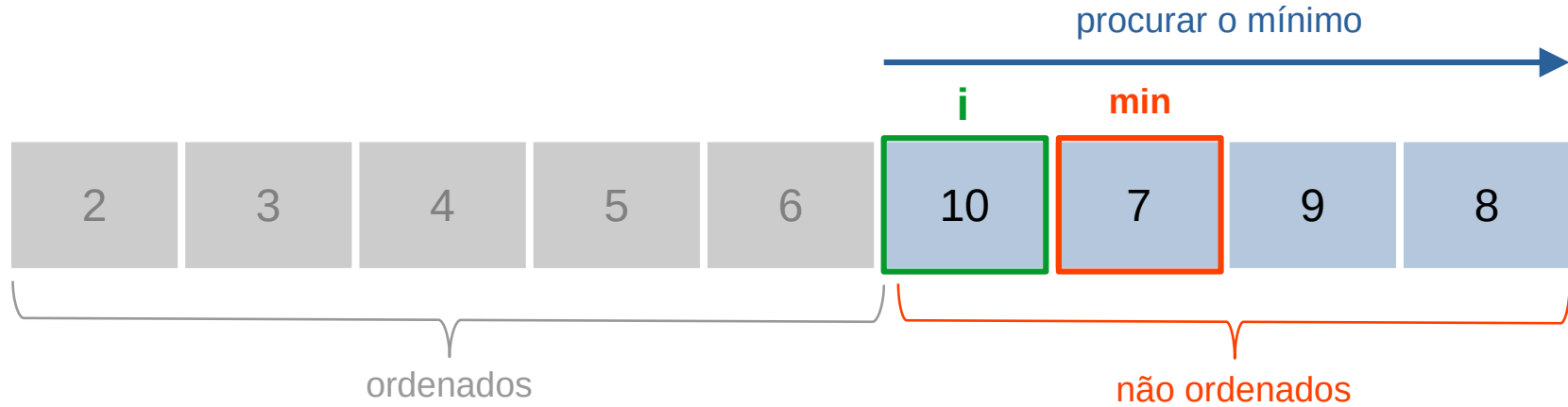
Iteração 5



Selection Sort

Exemplo (6)

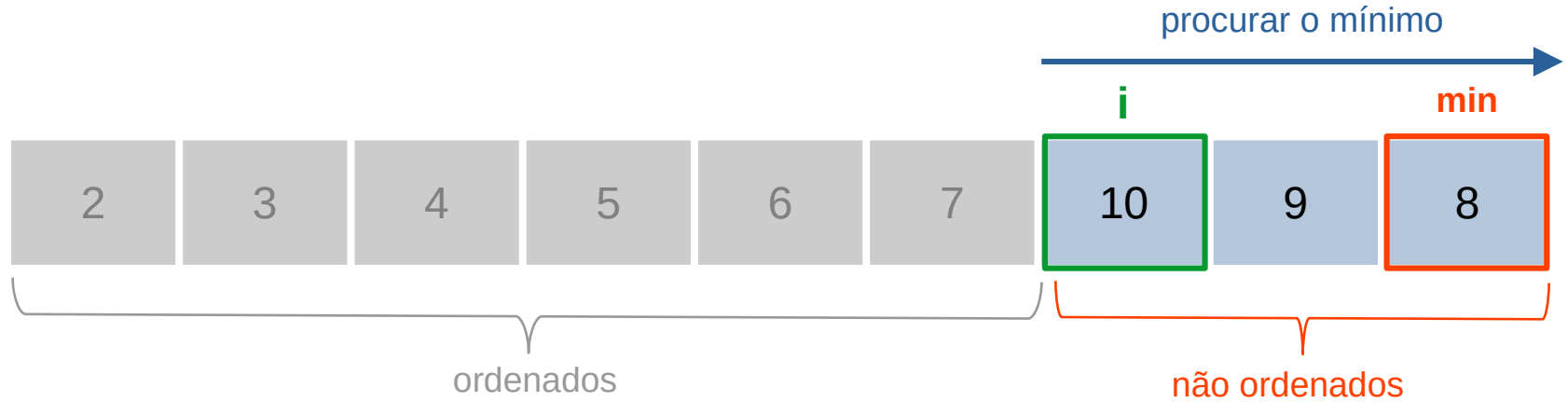
Iteração 6



Selection Sort

Exemplo (7)

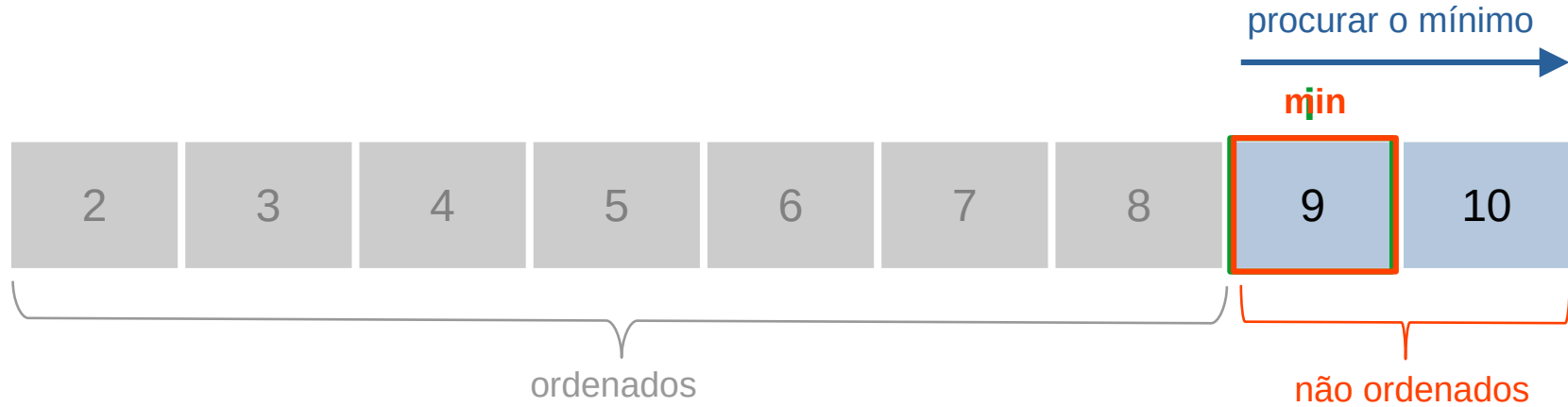
Iteração 7



Selection Sort

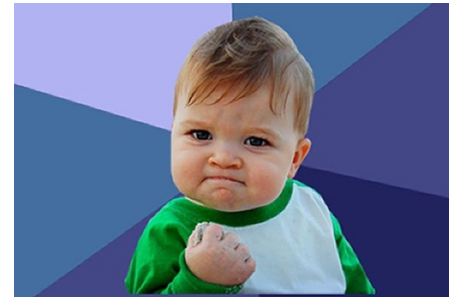
Exemplo (8)

Iteração 8



Selection Sort

Exemplo (9)



Selection Sort

Outro exemplo

- Lista original:

10	4	5	3	<u>10</u>	2
----	---	---	---	-----------	---

	0	1	2	3	4	5	i	vl. min
1ª it.	10	4	5	3	<u>10</u>	2	0	2
2ª it.	2	4	5	3	<u>10</u>	10	1	3
3ª it.	2	3	5	4	<u>10</u>	10	2	4
4ª it.	2	3	4	5	<u>10</u>	10	3	5
5ª it.	2	3	4	5	<u>10</u>	10	4	<u>10</u>
fim	2	3	4	5	<u>10</u>	10	5	-

Observe que o método não é estável: 10 vinha antes de 10, mas terminou depois

Selection Sort

Pseudocódigo

Algoritmo Selection

Início

```
para i de 0 até n-2 faça
```

```
    minPos = i
```

```
    para j de i+1 até n-1 faça
```

```
        se  $A[j] < A[\text{minPos}]$  então
```

```
            minPos = j
```

```
        fimSe
```

```
    fimPara
```

```
    troca( $A[i]$ ,  $A[\text{minPos}]$ )
```

```
    fimPara
```

Iteração interna: procura o mínimo

Iteração do método.

fimAlgoritmo

Selection Sort

Análise

- Complexidade $O(n^2)$ em **todos os casos**
 - Não importa a ordenação inicial da lista, serão feitas sempre $(n^2 - n) / 2$ comparações
- **In place**, isto é, a complexidade de espaço é constante
- **Não é estável**
- Cada item é colocado em sua posição final com apenas 1 troca
 - ou seja, são feitas no máximo N trocas
 - indicado para conjuntos de registros muito grandes (onde o custo da movimentação supera o custo das comparações)