

# Unidade V:

## Ordenação Interna - Algoritmo da Bolha




**PUC Minas**

Instituto de Ciências Exatas e Informática  
Departamento de Ciência da Computação

# Agenda

- Funcionamento básico
- Algoritmo em C#
- Análise dos número de movimentações e comparações

- **Funcionamento básico** 
- Algoritmo em C#
- Análise dos número de movimentações e comparações

Legenda:    - menor elemento em vermelho  
              - parte ordenada está de azul

101    115    30    63    47    20

101    115    30    63    47    20

Bolha

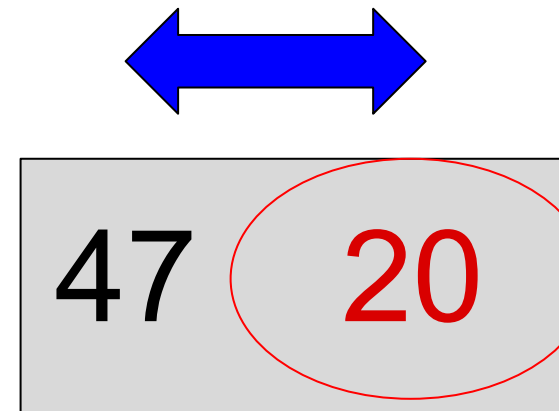
101    115    30    63

Comparação



Bolha

101    115    30    63

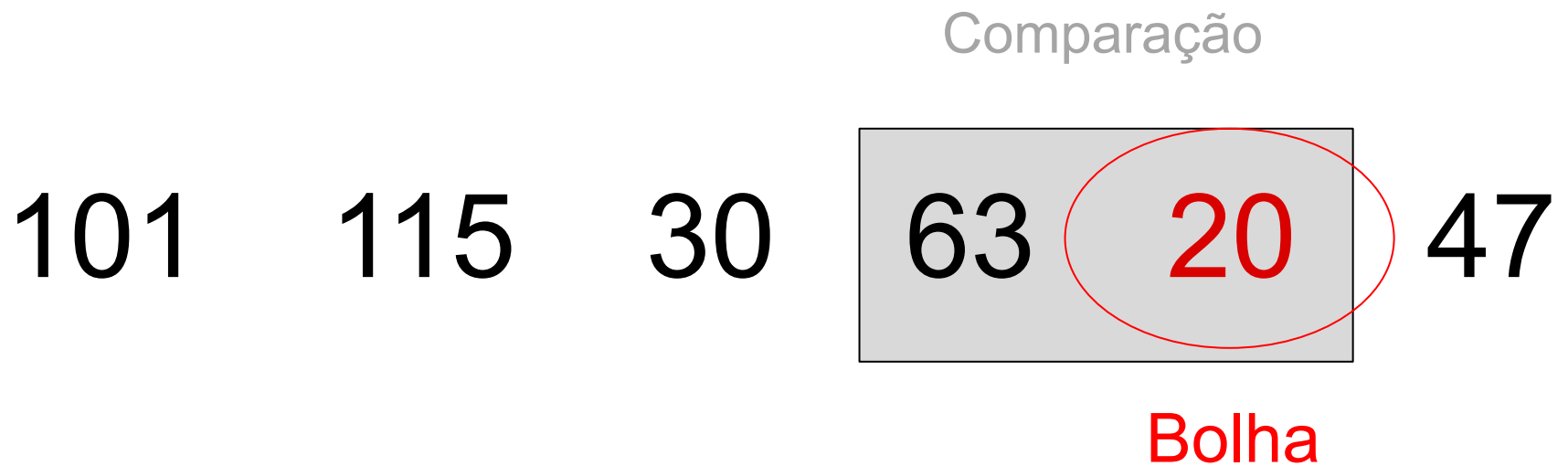


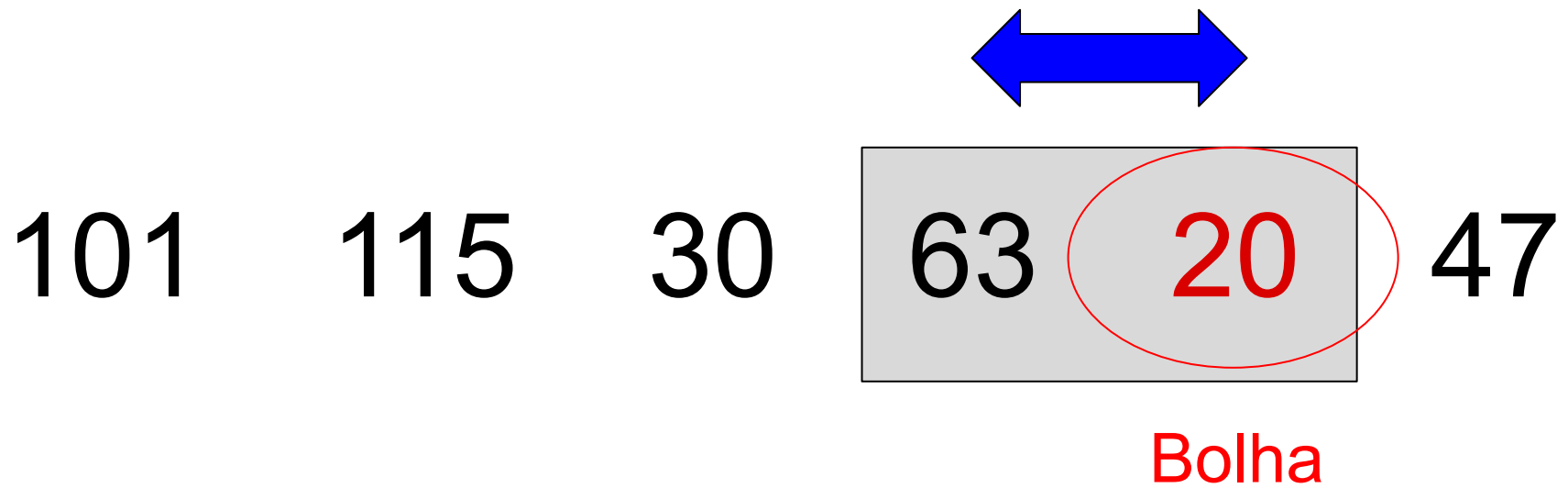
Bolha

101    115    30    63    20    47

Bolha

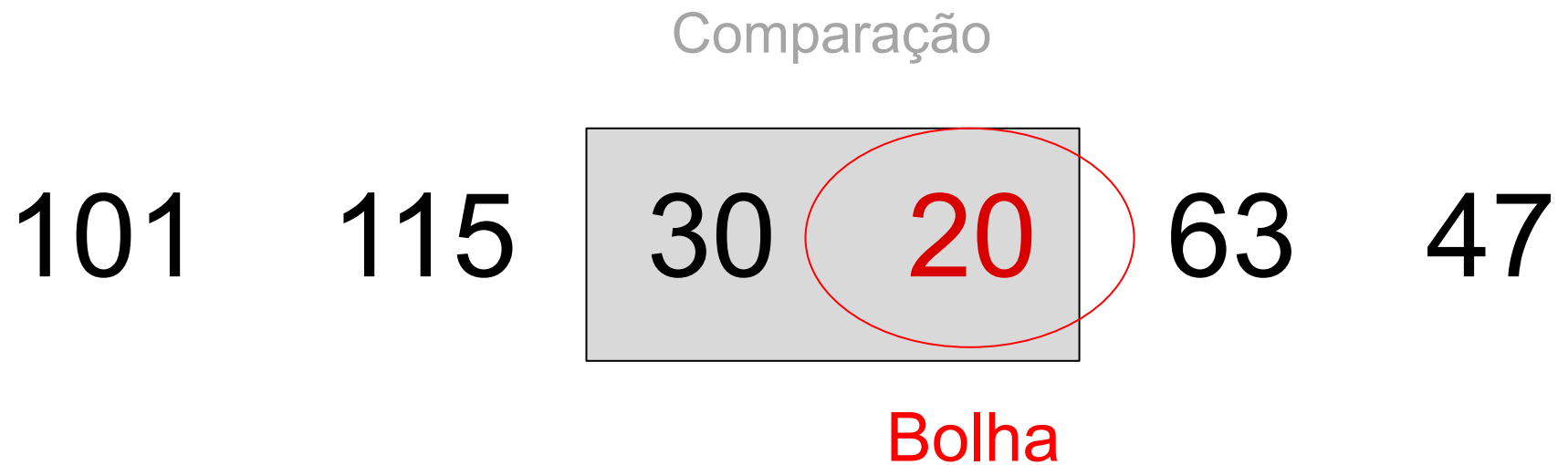


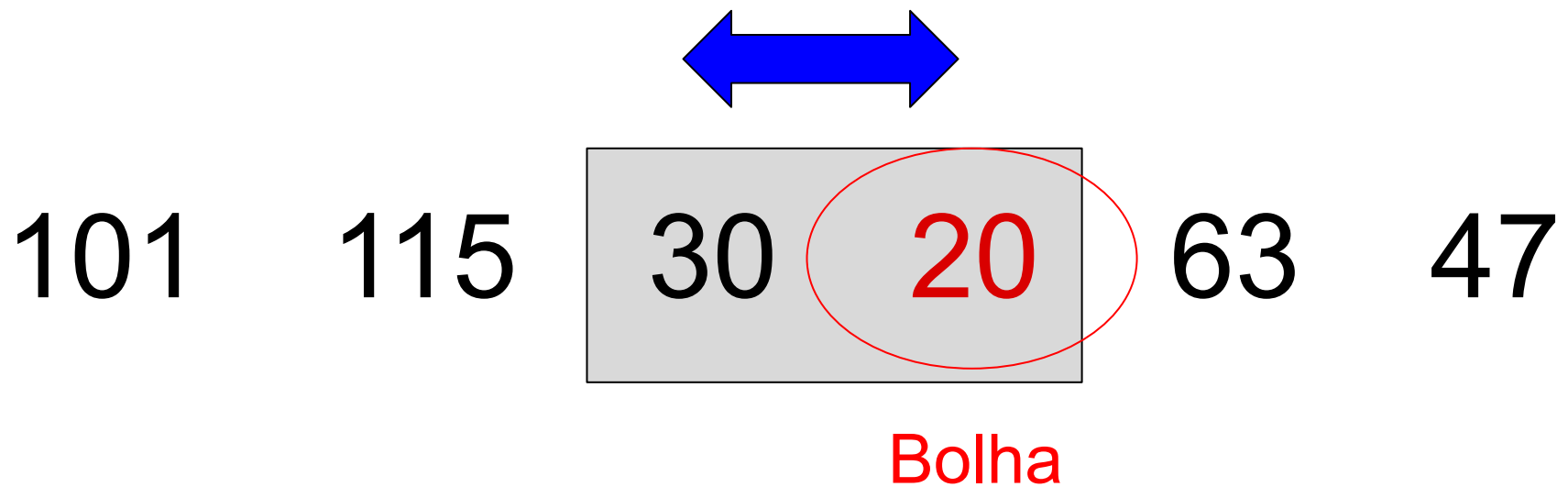




101    115    30    20    63    47

Bolha





101    115    20    30    63    47

Bolha

Comparação

101

115

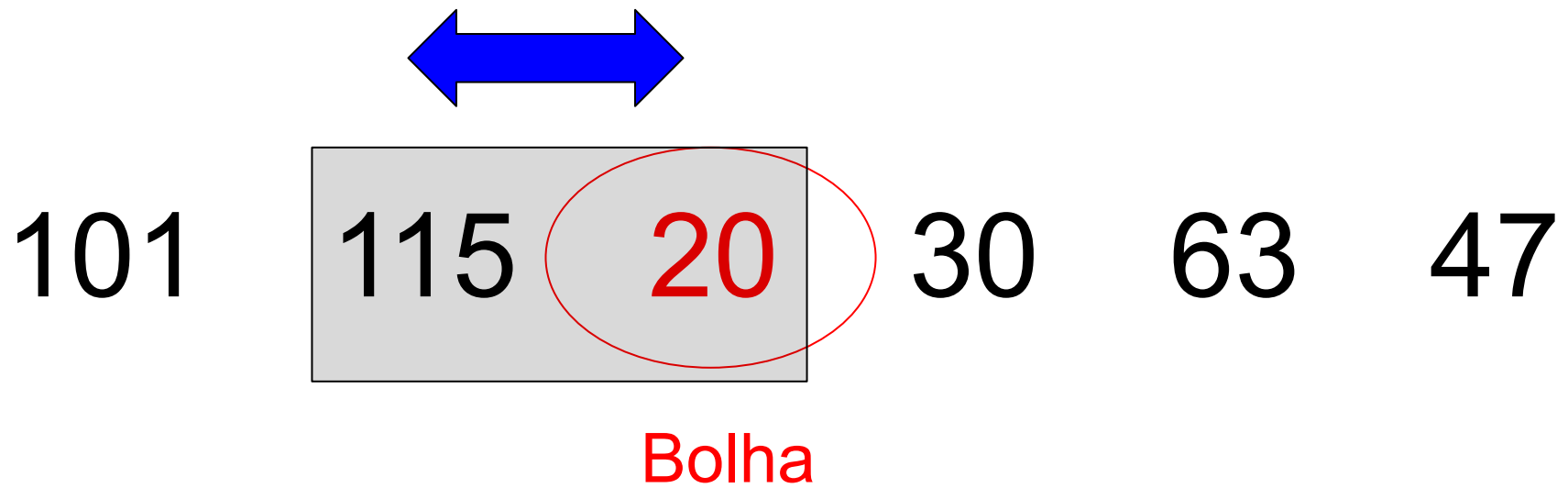
20

30

63

47

Bolha





101 20 115 30 63 47

Bolha

Comparação



Bolha



20 101 115 30 63 47

Bolha

20 101 115 30 63 47

Ordenado

20 101 115 30 63 47

Bolha

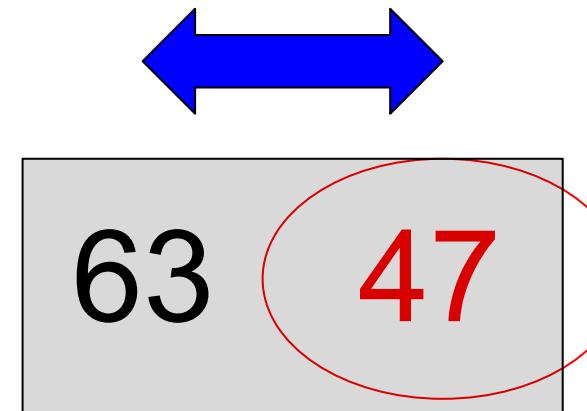
20 101 115 30

Comparação



Bolha

20 101 115 30

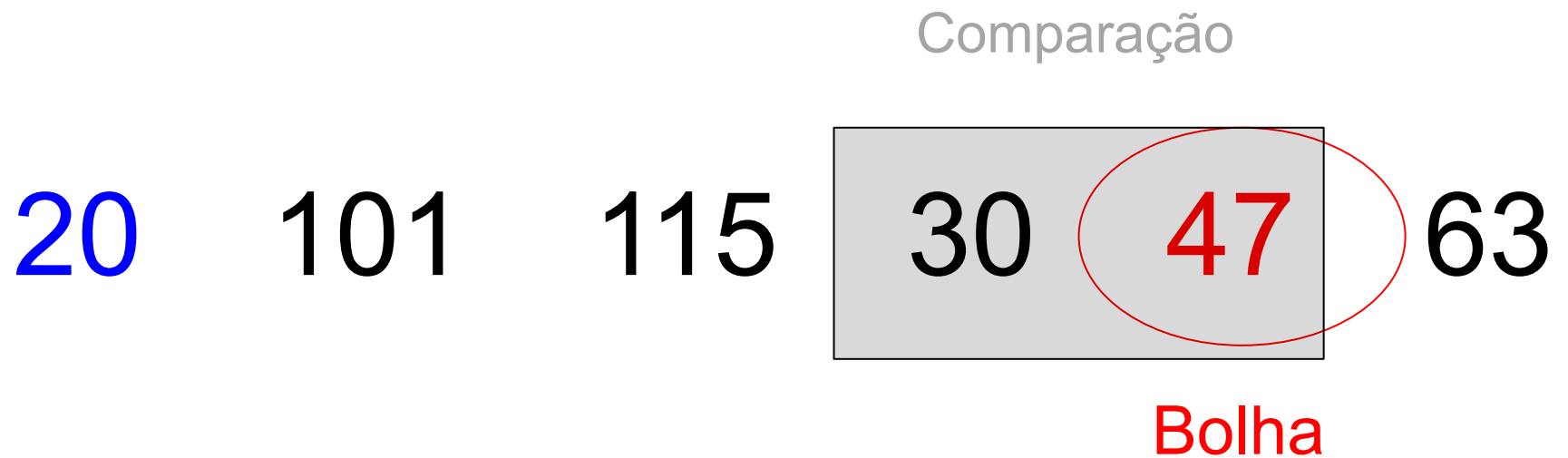


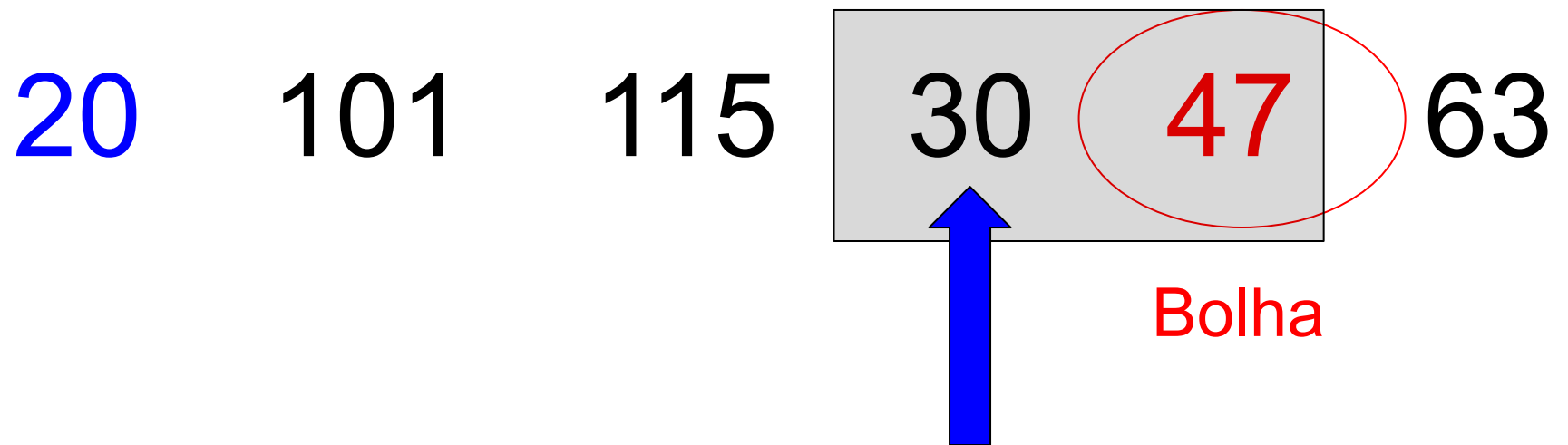
Bolha



20 101 115 30 47 63

Bolha



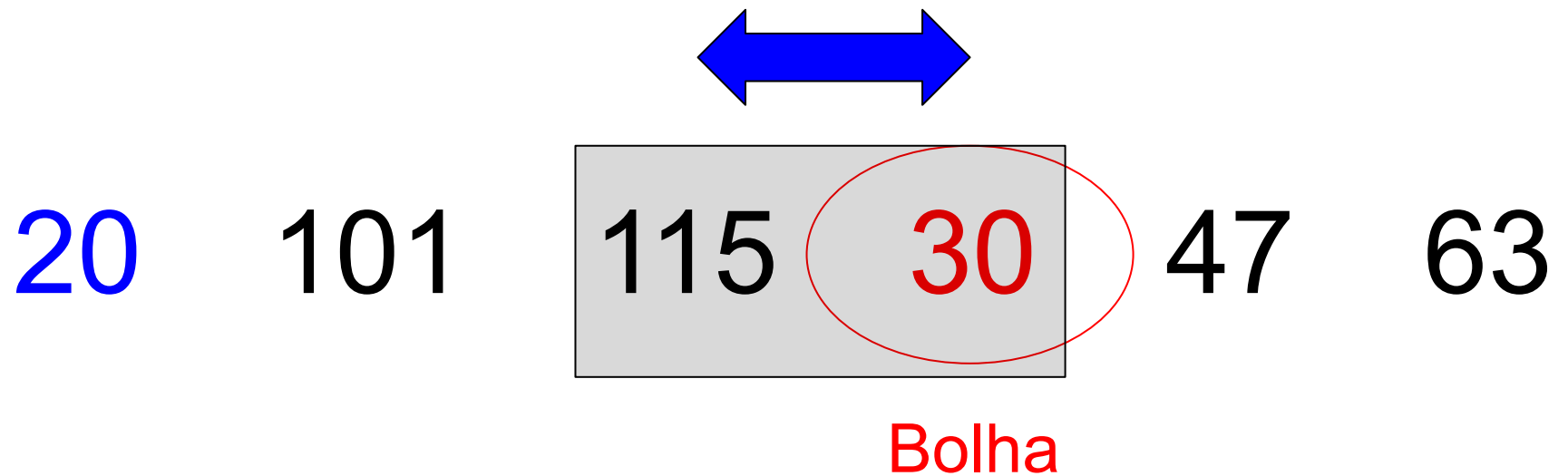


Menor (Será o número da bolha)

20 101 115 30 47 63

Bolha



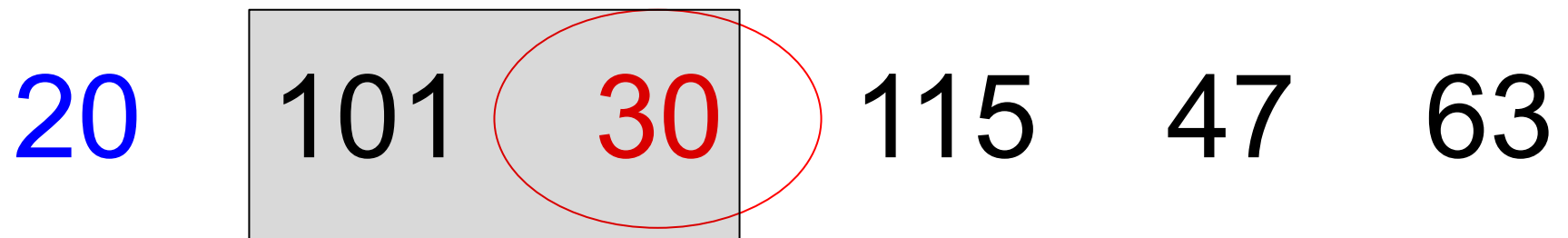


20 101 30 115 47 63



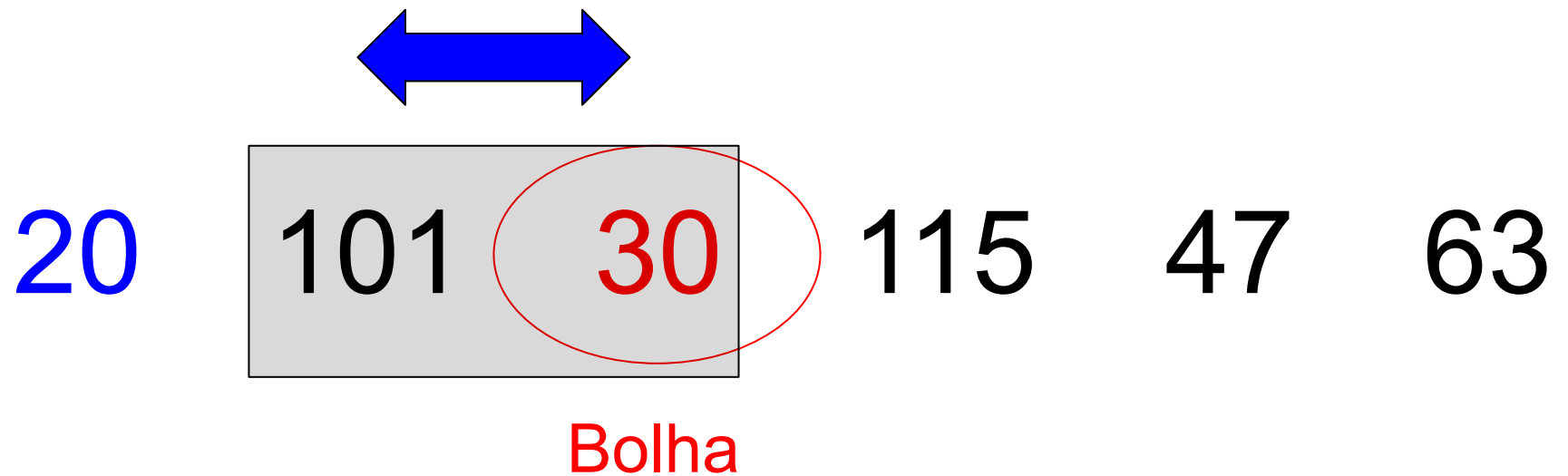
Bolha

Comparação

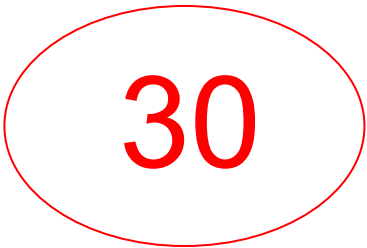


Bolha





20 30 101 115 47 63



Bolha

20 30 101 115 47 63

Ordenado

20 30 101 115 47 63

Bolha

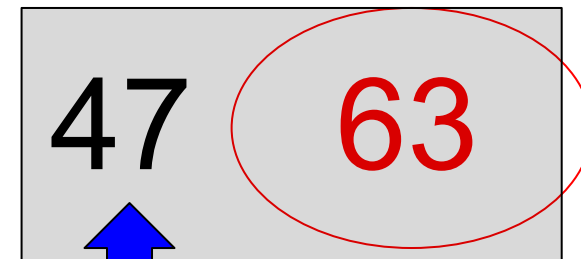
20 30 101 115

Comparação



Bolha

20      30      101      115

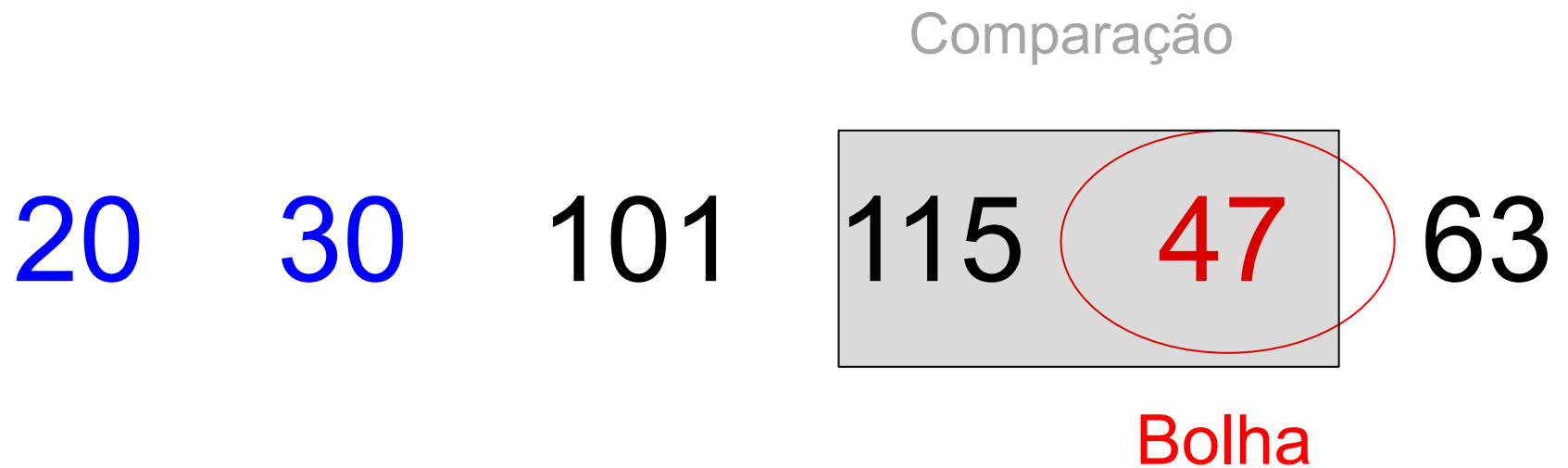


Bolha

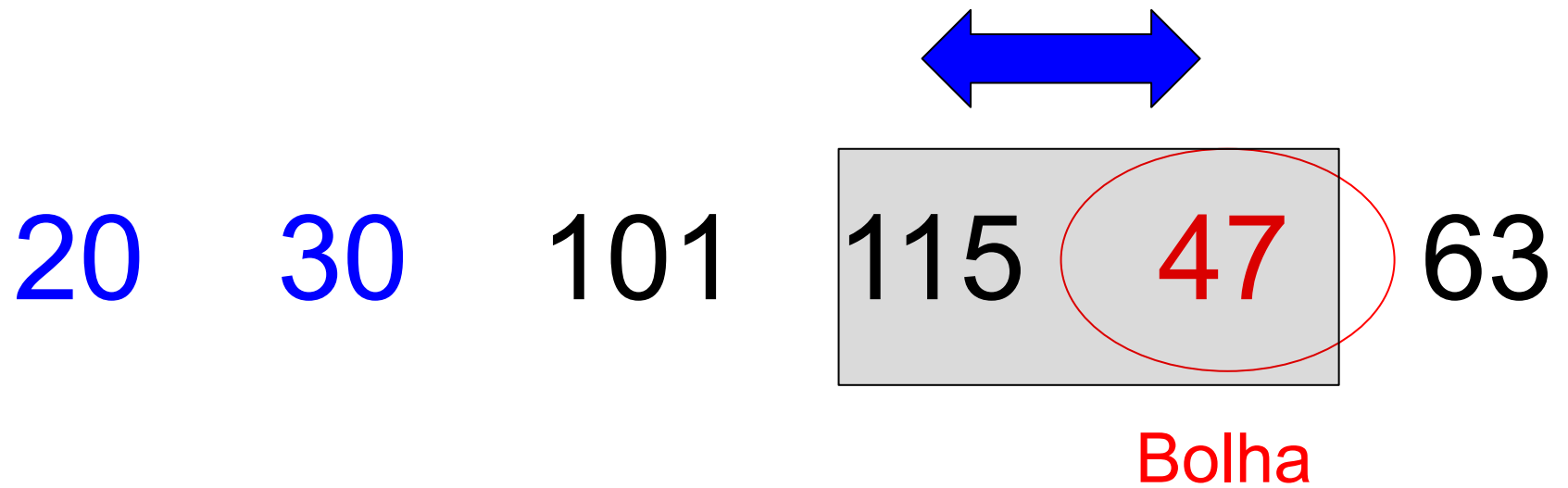
Menor (Será o número da bolha)

20 30 101 115 47 63

Bolha

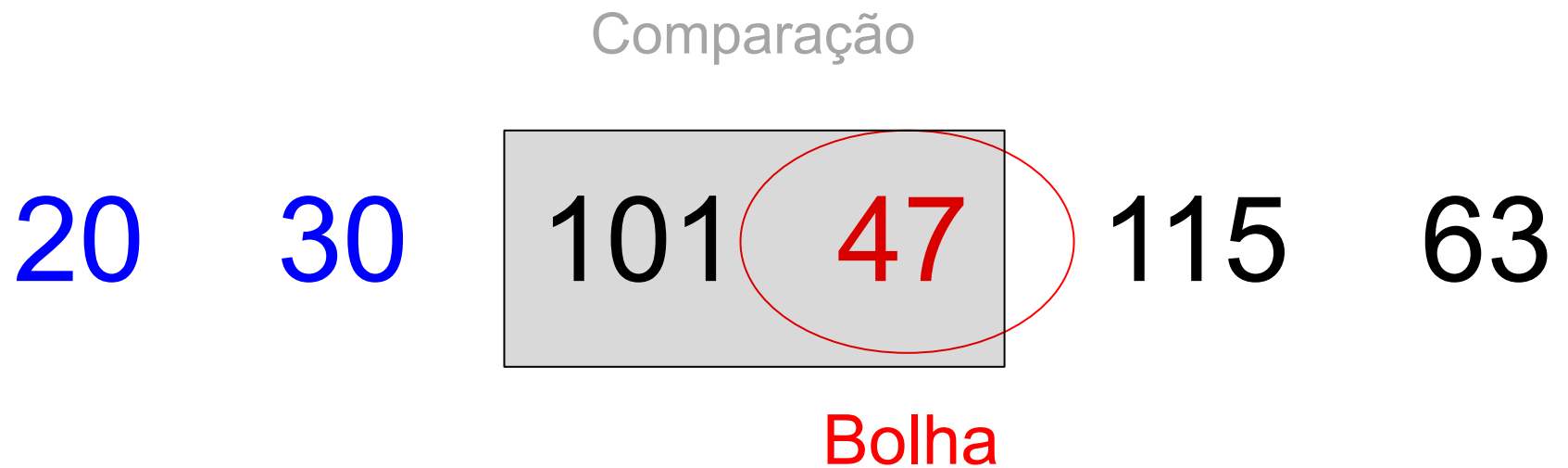


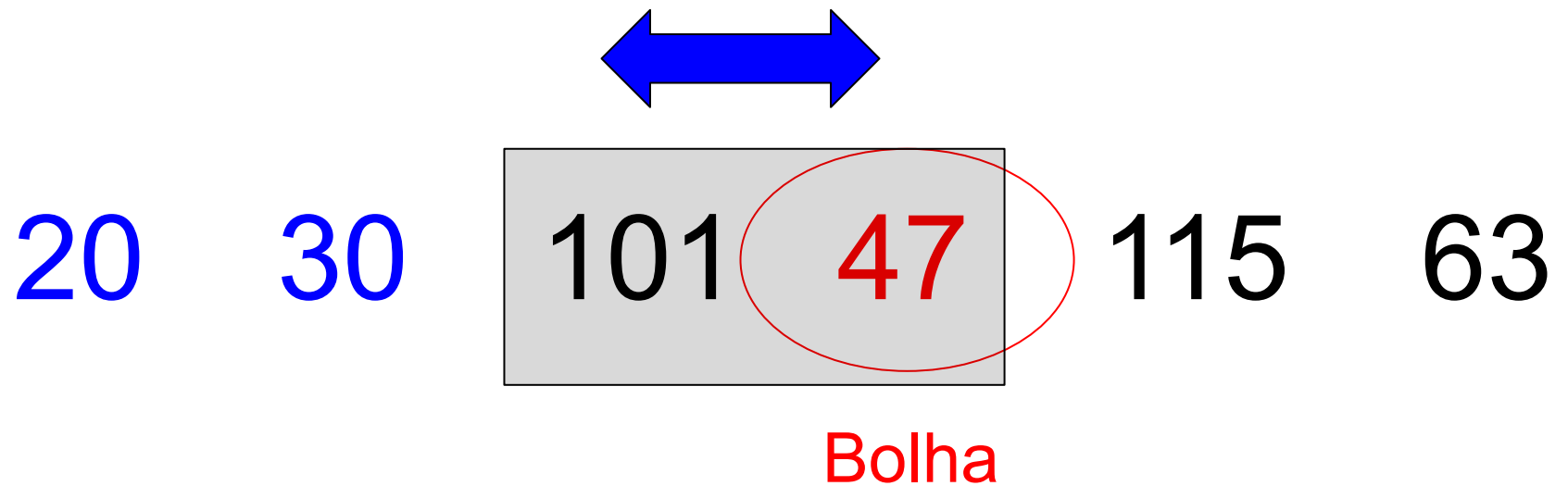




20 30 101 47 115 63

Bolha





20 30 47 101 115 63



Bolha

20    30    47    101    115    63

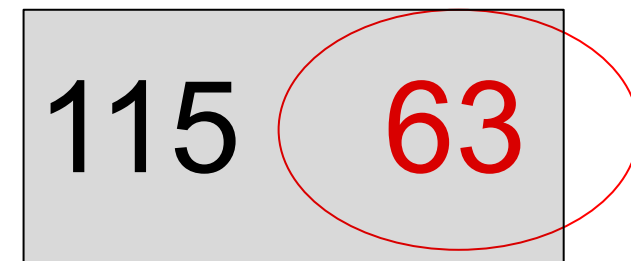
Ordenado

20 30 47 101 115 63

Bolha

20 30 47 101

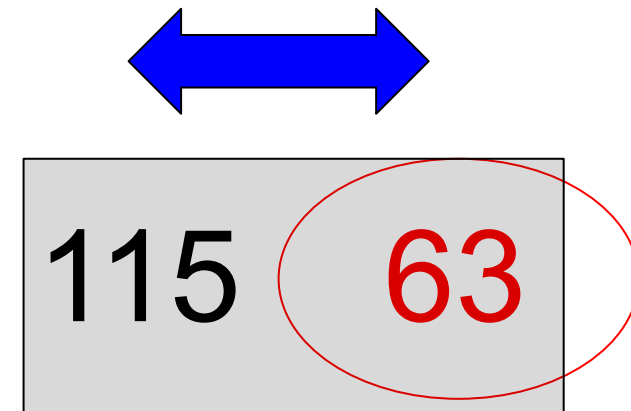
Comparação



Bolha



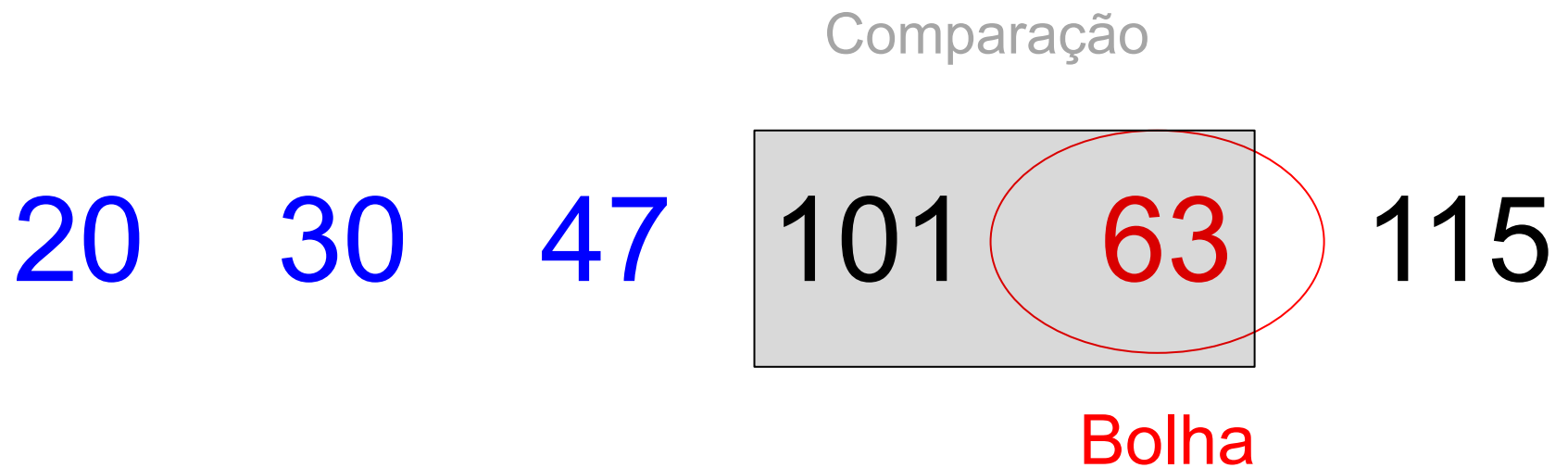
20 30 47 101

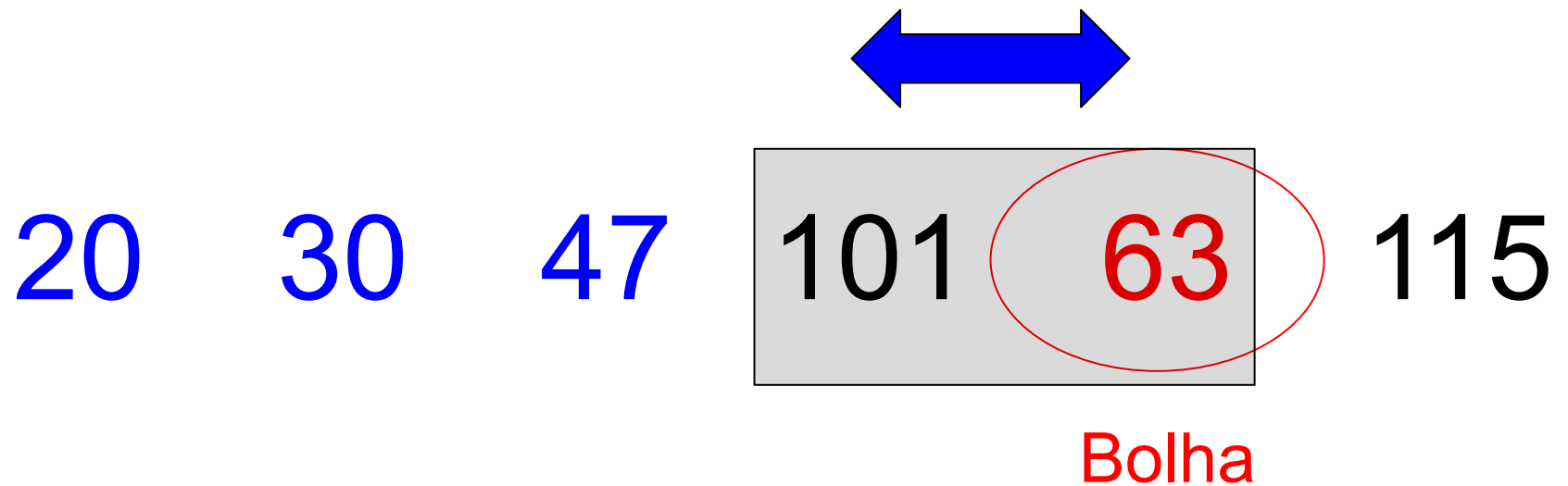


Bolha

20 30 47 101 63 115

Bolha





20 30 47 63 101 115

A horizontal sequence of numbers: 20, 30, 47, 63, 101, 115. The numbers 20, 30, and 47 are blue. The number 63 is red and enclosed in a red oval. The numbers 101 and 115 are black.

Bolha

20 30 47 63 101 115

Ordenado

20    30    47    63    101    115

Ordenado

O algoritmo terminou? Por que?

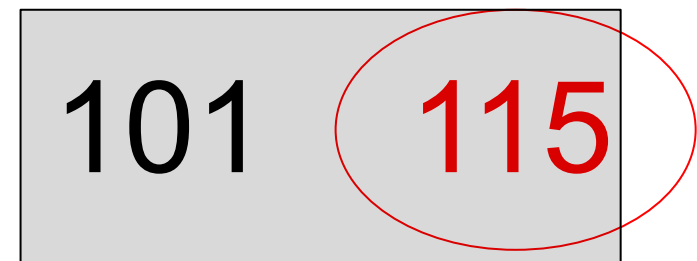
20 30 47 63 101 115

Bolha



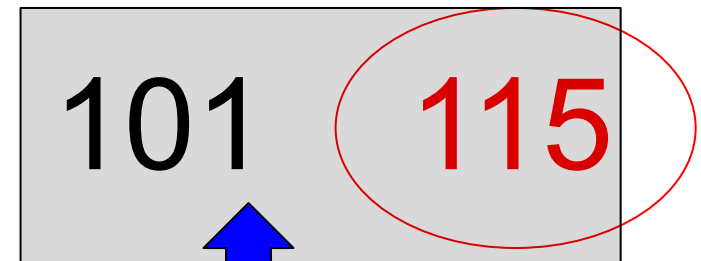
20 30 47 63

Comparação



Bolha

20 30 47 63



Bolha

Menor (Será o número da bolha)

20 30 47 63 101 115

Bolha

20 30 47 63 101 115

Ordenado


O algoritmo terminou? Por que?

20 30 47 63 101 115

Ordenado

# Conclusão

- O problema dos algoritmos de seleção e da bolha é porque eles realizam várias comparações redundantes
- Além disso, a bolha faz um número quadrático de movimentações
- Algoritmo estável

- Funcionamento básico
- **Algoritmo em C#** 
- Análise dos número de movimentações e comparações

# Algoritmo em C#

Ver código em: [fonte/unidade04/Bolha.java](#)



# Agenda

- Funcionamento básico
- Algoritmo em C#
- **Análise dos número de movimentações e comparações** 

# Análise do Número de Comparações

- Método de ordenação por seleção em que os registros são comparados, dois a dois e o menor é movimentado para o início do *array*

$$C(n) = \frac{n(n-1)}{2}, \text{ para os três casos}$$

# Análise do Número de Movimentações

- Pior caso: o *array* está ordenado de forma decrescente

$$M_{Max}(n) = 3 * \sum_{i=1}^{n-1} (n - i) = 3 * \frac{n(n - 1)}{2}$$

- Caso médio: depende do número de inversões em todas as permutações do *array*

$$M_{Med}(n) = 3 * \frac{n(n - 1)}{4}$$

## Exercício

- Mostre todas as comparações e movimentações do algoritmo anterior para o *array* abaixo:

12	4	8	2	14	17	6	18	10	16	15	5	13	9	1	11	7	3
----	---	---	---	----	----	---	----	----	----	----	---	----	---	---	----	---	---