

# Estrutura de Dados II

Prof. Me. Pietro M. de Oliveira

# Quicksort

Também conhecido como técnica de **classificação por troca de partição**.

Também utiliza o conceito “divide and conquer”

- Ordena arranjos menores, para ordenar o arranjo todo

Implementação inerentemente recursiva

- Chama a si mesmo para “dividir e conquistar”
- Código levemente complexo

Algoritmo Instável

Concebido por Sir Charles Antony Richard (1960)

**Divide and Conquer:** considere o arranjo **A**

- Divisão – Escolhe um **pivô**, que será o elemento chave para realizar a divisão do arranjo **A** em sub arranjos menores **A<sub>1</sub>** e **A<sub>2</sub>**.
- Recursão – À partir do **pivô**, realiza-se uma chamada do Quicksort para o sub arranjo **A<sub>1</sub>** e outra chamada recursiva para o sub arranjo **A<sub>2</sub>**.
- Conquista – Arranjo **A<sub>1</sub>** deverá conter os elementos menores que o **pivô**, e o arranjo **A<sub>2</sub>**, os elementos maiores. Assim, conquista-se a posição correta do **pivô**, e ordenando o arranjo **A**.

## Eficiência:

- Algoritmo de ordenação mais utilizado
- Considerado o mais rápido em diversas situações
  - No melhor caso, realiza  $n \times \log n$  operações para realizar a ordenação
- Em vetores ordenados, ou parcialmente ordenados, tem-se o pior caso:  $n^2$  comparações
  - “Escolha ruim” do pivô
- Não utiliza memória auxiliar
  - Ordenação interna

## Algoritmo

**Quicksort**(arranjo *A*, *inicio*, *fim*)

1. Se *inicio* < *fim* então
2.     *meio*  $\leftarrow$  *Partition*(*A*, *inicio*, *fim*)
3.     Quicksort(*A*, *inicio*, *meio*-1)
4.     Quicksort(*A*, *meio*+1, *fim*)

## Subalgoritmo

**Partition**(arranjo *A*, *inicio*, *fim*)

1. ***pivo*  $\leftarrow A[fim]$**
2. ***i*  $\leftarrow inicio - 1$**
3. Para ***j*  $\leftarrow inicio$**  até ***fim-1*** faça
4.     Se ***A[j]*  $\leq pivo$**  então
5.         ***i*  $\leftarrow i + 1$**
6.         troca ***A[i]*  $\leftrightarrow A[j]$**
7. troca ***A[i+1]*  $\leftrightarrow A[fim]$**
8. retorne ***i+1***

**Quicksort**(arranjo *A*, *inicio*, *fim*)

1. Se *inicio* < *fim* então
2.     *meio*  $\leftarrow$  *Partition*(*A*, *inicio*, *fim*)
3.     Quicksort(*A*, *inicio*, *meio*-1)
4.     Quicksort(*A*, *meio*+1, *fim*)

0	1	2	3	4	5	6	7
26	32	46	19	15	67	81	22



## Chamada Original

**Quicksort**(arranjo **A**, *início*, *fim*)

1. Se *início* < *fim* então
2.     *meio*  $\leftarrow$  **Partition**(**A**, *início*, *fim*)
3.     **Quicksort**(**A**, *início*, *meio*-1)
4.     **Quicksort**(**A**, *meio*+1, *fim*)

0	1	2	3	4	5	6	7
26	32	46	19	15	67	81	22

## (Chamada Original)

## Chamada do Partition

**Quicksort**(arranjo **A**, **inicio**, **fim**)

1. Se **inicio** < **fim** então
2.     **meio**  $\leftarrow$  **Partition**(**A**, **inicio**, **fim**)
3.     **Quicksort**(**A**, **inicio**, **meio**-1)
4.     **Quicksort**(**A**, **meio**+1, **fim**)

0	1	2	3	4	5	6	7
26	32	46	19	15	67	81	22

## Partition (Quicksort) - Exemplo

**Partition**(arranjo **A**, *inicio*, *fim*)

1. **pivo**  $\leftarrow A[\textit{fim}]$
2. **i**  $\leftarrow \textit{inicio} - 1$
3. Para **j**  $\leftarrow \textit{inicio}$  até **fim-1** faça
4.     Se **A[j]**  $\leq$  **pivo** então
5.         **i**  $\leftarrow i + 1$
6.         troca **A[i]**  $\leftrightarrow$  **A[j]**
7. troca **A[i+1]**  $\leftrightarrow$  **A[fim]**
8. retorne **i+1**

0	1	2	3	4	5	6	7
26	32	46	19	15	67	81	22

## Partition (Quicksort) - Exemplo

**Partition**(arranjo **A**, *inicio*, *fim*)

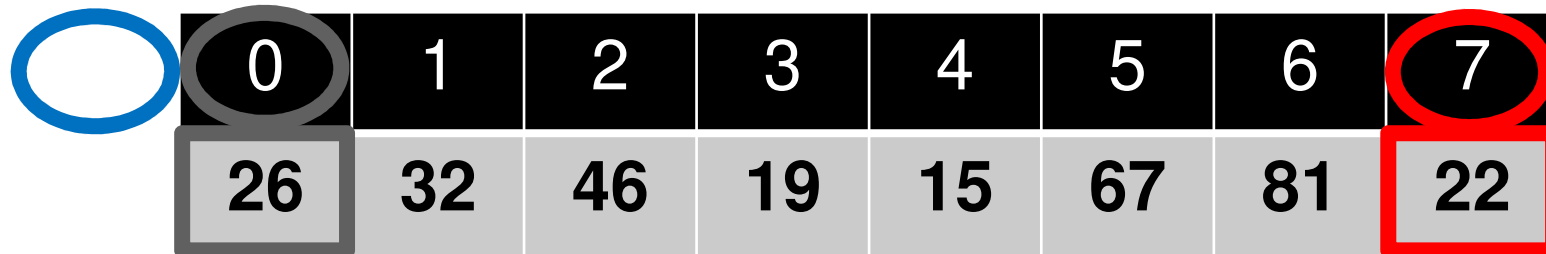
1. *pivo*  $\leftarrow A[\textit{fim}]$
2. *i*  $\leftarrow \textit{inicio} - 1$
3. Para *j*  $\leftarrow \textit{inicio}$  até *fim-1* faça
4.     Se  $A[j] \leq \textit{pivo}$  então
5.         *i*  $\leftarrow i + 1$
6.         troca  $A[i] \leftrightarrow A[j]$
7. troca  $A[i+1] \leftrightarrow A[\textit{fim}]$
8. retorne *i+1*

	0	1	2	3	4	5	6	7
	26	32	46	19	15	67	81	22

## Partition (Quicksort) - Exemplo

**Partition**(arranjo *A*, *inicio*, *fim*)

1. ***pivo***  $\leftarrow A[\textit{fim}]$
2. ***i***  $\leftarrow \textit{inicio} - 1$
3. Para ***j***  $\leftarrow \textit{inicio}$  até ***fim-1*** faça
4.     Se ***A[j]***  $\leq$  ***pivo*** então
5.         ***i***  $\leftarrow i + 1$
6.         troca ***A[i]***  $\leftrightarrow$  ***A[j]***
7. troca ***A[i+1]***  $\leftrightarrow$  ***A[fim]***
8. retorne ***i+1***

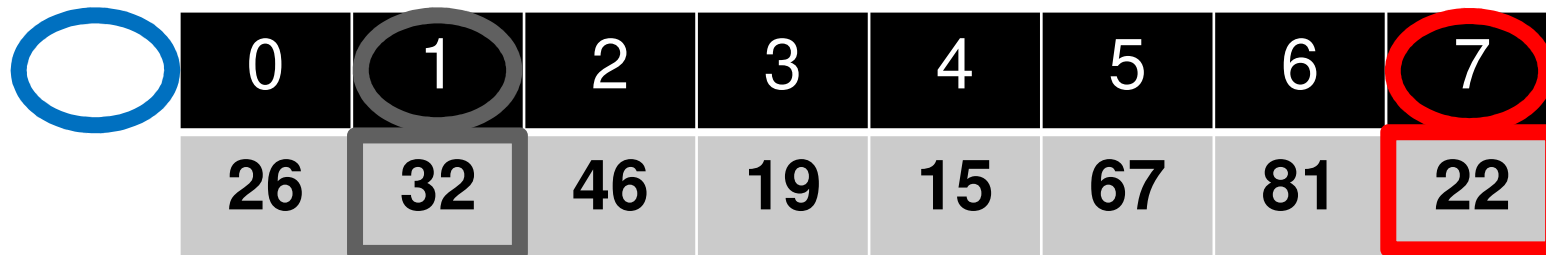


	0	1	2	3	4	5	6	7
	26	32	46	19	15	67	81	22

## Partition (Quicksort) - Exemplo

**Partition**(arranjo  $A$ ,  $inicio$ ,  $fim$ )

1.  $pivo \leftarrow A[fim]$
2.  $i \leftarrow inicio - 1$
3. Para  $j \leftarrow inicio$  até  $fim-1$  faça
4.     Se  $A[j] \leq pivo$  então
5.          $i \leftarrow i + 1$
6.         troca  $A[i] \leftrightarrow A[j]$
7. troca  $A[i+1] \leftrightarrow A[fim]$
8. retorne  $i+1$

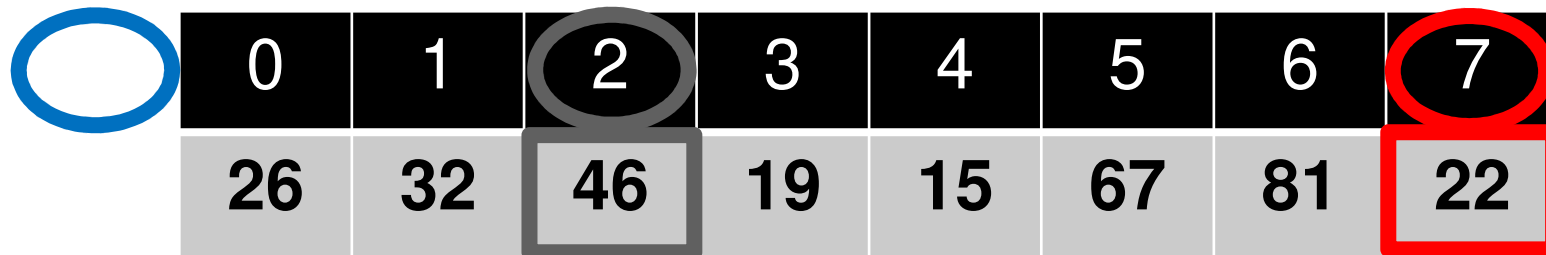


0	1	2	3	4	5	6	7
26	32	46	19	15	67	81	22

## Partition (Quicksort) - Exemplo

**Partition**(arranjo  $A$ ,  $inicio$ ,  $fim$ )

1.  $pivo \leftarrow A[fim]$
2.  $i \leftarrow inicio - 1$
3. Para  $j \leftarrow inicio$  até  $fim-1$  faça
4.     Se  $A[j] \leq pivo$  então
5.          $i \leftarrow i + 1$
6.         troca  $A[i] \leftrightarrow A[j]$
7. troca  $A[i+1] \leftrightarrow A[fim]$
8. retorne  $i+1$

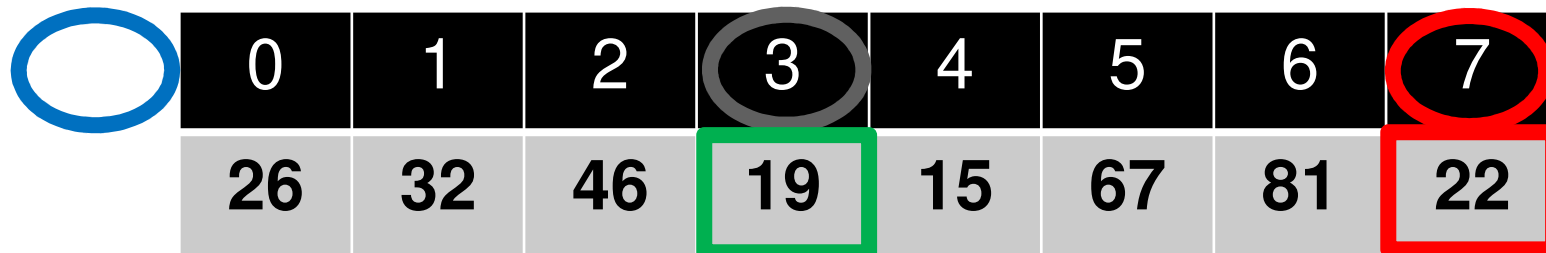


0	1	2	3	4	5	6	7
26	32	46	19	15	67	81	22

## Partition (Quicksort) - Exemplo

**Partition**(arranjo  $A$ ,  $inicio$ ,  $fim$ )

1.  $pivo \leftarrow A[fim]$
2.  $i \leftarrow inicio - 1$
3. Para  $j \leftarrow inicio$  até  $fim-1$  faça
4.     Se  $A[j] \leq pivo$  então
5.          $i \leftarrow i + 1$
6.         troca  $A[i] \leftrightarrow A[j]$
7. troca  $A[i+1] \leftrightarrow A[fim]$
8. retorne  $i+1$



0	1	2	3	4	5	6	7
26	32	46	19	15	67	81	22



## Partition (Quicksort) - Exemplo

**Partition**(arranjo  $A$ ,  $inicio$ ,  $fim$ )

1.  $pivo \leftarrow A[fim]$
2.  $i \leftarrow inicio - 1$
3. Para  $j \leftarrow inicio$  até  $fim-1$  faça
4.     Se  $A[j] \leq pivo$  então
5.          $i \leftarrow i + 1$
6.         troca  $A[i] \leftrightarrow A[j]$
7. troca  $A[i+1] \leftrightarrow A[fim]$
8. retorne  $i+1$

0	1	2	3	4	5	6	7
26	32	46	19	15	67	81	22

## Partition (Quicksort) - Exemplo

**Partition**(arranjo  $A$ ,  $inicio$ ,  $fim$ )

1.  $pivo \leftarrow A[fim]$
2.  $i \leftarrow inicio - 1$
3. Para  $j \leftarrow inicio$  até  $fim-1$  faça
4.     Se  $A[j] \leq pivo$  então
5.          $i \leftarrow i + 1$
6.         troca  $A[i] \leftrightarrow A[j]$
7. troca  $A[i+1] \leftrightarrow A[fim]$
8. retorne  $i+1$

0	1	2	3	4	5	6	7
19	32	46	26	15	67	81	22

## Partition (Quicksort) - Exemplo

**Partition**(arranjo  $A$ ,  $inicio$ ,  $fim$ )

1.  $pivo \leftarrow A[fim]$
2.  $i \leftarrow inicio - 1$
3. Para  $j \leftarrow inicio$  até  $fim-1$  faça
4.     Se  $A[j] \leq pivo$  então
5.          $i \leftarrow i + 1$
6.         troca  $A[i] \leftrightarrow A[j]$
7. troca  $A[i+1] \leftrightarrow A[fim]$
8. retorne  $i+1$

0	1	2	3	4	5	6	7
19	32	46	26	15	67	81	22

## Partition (Quicksort) - Exemplo

**Partition**(arranjo  $A$ ,  $inicio$ ,  $fim$ )

1.  $pivo \leftarrow A[fim]$
2.  $i \leftarrow inicio - 1$
3. Para  $j \leftarrow inicio$  até  $fim-1$  faça
4.     Se  $A[j] \leq pivo$  então
5.          $i \leftarrow i + 1$
6.         troca  $A[i] \leftrightarrow A[j]$
7. troca  $A[i+1] \leftrightarrow A[fim]$
8. retorne  $i+1$

0	1	2	3	4	5	6	7
19	32	46	26	15	67	81	22

## Partition (Quicksort) - Exemplo

**Partition**(arranjo  $A$ ,  $inicio$ ,  $fim$ )

1.  $pivo \leftarrow A[fim]$
2.  $i \leftarrow inicio - 1$
3. Para  $j \leftarrow inicio$  até  $fim-1$  faça
4.     Se  $A[j] \leq pivo$  então
5.          $i \leftarrow i + 1$
6.         troca  $A[i] \leftrightarrow A[j]$
7. troca  $A[i+1] \leftrightarrow A[fim]$
8. retorne  $i+1$

0	1	2	3	4	5	6	7
19	15	46	26	32	67	81	22

## Partition (Quicksort) - Exemplo

**Partition**(arranjo *A*, *inicio*, *fim*)

1. ***pivo***  $\leftarrow A[\textit{fim}]$
2. ***i***  $\leftarrow \textit{inicio} - 1$
3. Para ***j***  $\leftarrow \textit{inicio}$  até ***fim-1*** faça
4.     Se ***A[j]***  $\leq$  ***pivo*** então
5.         ***i***  $\leftarrow i + 1$
6.         troca ***A[i]***  $\leftrightarrow$  ***A[j]***
7. troca ***A[i+1]***  $\leftrightarrow$  ***A[fim]***
8. retorne ***i+1***

0	1	2	3	4	5	6	7
19	15	46	26	32	67	81	22

## Partition (Quicksort) - Exemplo

**Partition**(arranjo  $A$ ,  $inicio$ ,  $fim$ )

1.  $pivo \leftarrow A[fim]$
2.  $i \leftarrow inicio - 1$
3. Para  $j \leftarrow inicio$  até  $fim-1$  faça
4.     Se  $A[j] \leq pivo$  então
5.          $i \leftarrow i + 1$
6.         troca  $A[i] \leftrightarrow A[j]$
7. troca  $A[i+1] \leftrightarrow A[fim]$
8. retorne  $i+1$

0	1	2	3	4	5	6	7
19	15	46	26	32	67	81	22

## Partition (Quicksort) - Exemplo

**Partition**(arranjo  $A$ ,  $inicio$ ,  $fim$ )

1.  $pivo \leftarrow A[fim]$
2.  $i \leftarrow inicio - 1$
3. Para  $j \leftarrow inicio$  até  $fim-1$  faça
4.     Se  $A[j] \leq pivo$  então
5.          $i \leftarrow i + 1$
6.         troca  $A[i] \leftrightarrow A[j]$
7. troca  $A[i+1] \leftrightarrow A[fim]$
8. retorne  $i+1$

0	1	2	3	4	5	6	7
19	15	46	26	32	67	81	22



## Partition (Quicksort) - Exemplo

**Partition**(arranjo  $A$ ,  $inicio$ ,  $fim$ )

1.  $pivo \leftarrow A[fim]$
2.  $i \leftarrow inicio - 1$
3. Para  $j \leftarrow inicio$  até  $fim-1$  faça
4.     Se  $A[j] \leq pivo$  então
5.          $i \leftarrow i + 1$
6.         troca  $A[i] \leftrightarrow A[j]$
7. troca  $A[i+1] \leftrightarrow A[fim]$
8. retorne  $i+1$

0	1	2	3	4	5	6	7
19	15	22	26	32	67	81	46

## (Chamada Original)

## Retorno do Partition

**Quicksort**(arranjo ***A***, ***inicio***, ***fim***)

1. Se ***inicio*** < ***fim*** então
2.     ***meio***  $\leftarrow$  ***Partition(A, inicio, fim)***
3.     **Quicksort**(***A***, ***inicio***, ***meio***-1)
4.     **Quicksort**(***A***, ***meio***+1, ***fim***)

0	1	2	3	4	5	6	7
15	19	22	26	32	67	81	46

## (Chamada Original)

## Chamada Recursiva 1

**Quicksort**(arranjo ***A***, ***inicio***, ***fim***)

1. Se ***inicio*** < ***fim*** então
2.     ***meio***  $\leftarrow$  ***Partition***(***A***, ***inicio***, ***fim***)
3.     **Quicksort**(***A***, ***inicio***, ***meio*-1**)
4.     **Quicksort**(***A***, ***meio*+1**, ***fim***)

0	1	2	3	4	5	6	7
19	15	22	26	32	67	81	46

## (Chamada Recursiva 1)

### Chamada do Partition

**Quicksort**(arranjo **A**, *inicio*, *fim*)

1. Se *inicio* < *fim* então
2.     *meio*  $\leftarrow$  **Partition**(**A**, *inicio*, *fim*)
3.     **Quicksort**(**A**, inicio, meio-1)
4.     **Quicksort**(**A**, meio+1, fim)

0	1
15	19

**Partition**(arranjo **A**, *inicio*, *fim*)

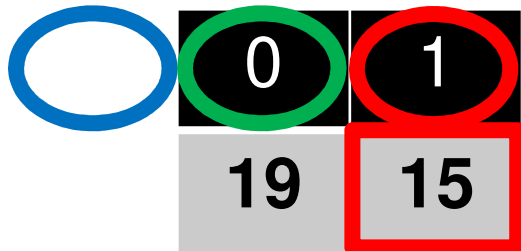
1. **pivo**  $\leftarrow A[\textit{fim}]$
2. **i**  $\leftarrow \textit{inicio} - 1$
3. Para **j**  $\leftarrow \textit{inicio}$  até **fim-1** faça
4.     Se **A[j]**  $\leq$  **pivo** então
5.         **i**  $\leftarrow i + 1$
6.         troca **A[i]**  $\leftrightarrow$  **A[j]**
7. troca **A[i+1]**  $\leftrightarrow$  **A[fim]**
8. retorne **i+1**

0	1
19	15

## Partition (Quicksort) - Exemplo

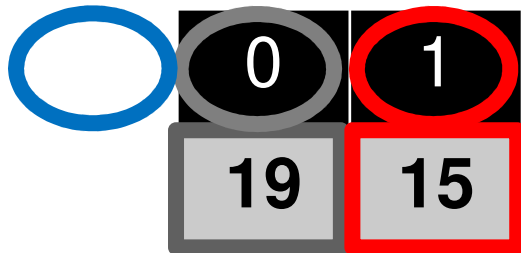
**Partition**(arranjo **A**, *inicio*, *fim*)

1. **pivo**  $\leftarrow A[\text{fim}]$
2. **i**  $\leftarrow \text{inicio} - 1$
3. Para **j**  $\leftarrow \text{inicio}$  até **fim-1** faça
4.     Se **A[j]**  $\leq$  **pivo** então
5.         **i**  $\leftarrow i + 1$
6.         troca **A[i]**  $\leftrightarrow$  **A[j]**
7. troca **A[i+1]**  $\leftrightarrow$  **A[fim]**
8. retorne **i+1**



**Partition**(arranjo  $A$ ,  $inicio$ ,  $fim$ )

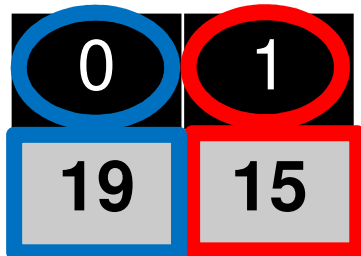
1.  $pivo \leftarrow A[fim]$
2.  $i \leftarrow inicio - 1$
3. Para  $j \leftarrow inicio$  até  $fim-1$  faça
4.     Se  $A[j] \leq pivo$  então
5.          $i \leftarrow i + 1$
6.         troca  $A[i] \leftrightarrow A[j]$
7. troca  $A[i+1] \leftrightarrow A[fim]$
8. retorne  $i+1$



## Partition (Quicksort) - Exemplo

**Partition**(arranjo  $A$ ,  $inicio$ ,  $fim$ )

1.  $pivo \leftarrow A[fim]$
2.  $i \leftarrow inicio - 1$
3. Para  $j \leftarrow inicio$  até  $fim-1$  faça
4.     Se  $A[j] \leq pivo$  então
5.          $i \leftarrow i + 1$
6.         troca  $A[i] \leftrightarrow A[j]$
7. troca  $A[i+1] \leftrightarrow A[fim]$
8. retorne  $i+1$

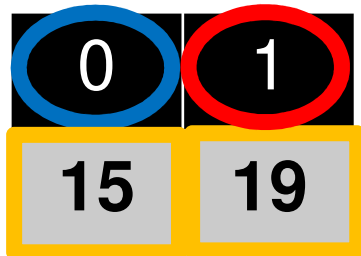




## Partition (Quicksort) - Exemplo

**Partition**(arranjo *A*, *inicio*, *fim*)

1. ***pivo*  $\leftarrow A[fim]$**
2. ***i*  $\leftarrow inicio - 1$**
3. Para ***j*  $\leftarrow inicio$**  até ***fim-1*** faça
4.     Se ***A[j]*  $\leq pivo$**  então
5.         ***i*  $\leftarrow i + 1$**
6.         troca ***A[i]*  $\leftrightarrow A[j]$**
7. troca ***A[i+1]*  $\leftrightarrow A[fim]$**
8. retorne ***i+1***



**Partition**(arranjo *A*, *inicio*, *fim*)

1. ***pivo***  $\leftarrow A[\textit{fim}]$
2. ***i***  $\leftarrow \textit{inicio} - 1$
3. Para ***j***  $\leftarrow \textit{inicio}$  até ***fim-1*** faça
4.     Se  **$A[j] \leq \textit{pivo}$**  então
5.          **$i \leftarrow i + 1$**
6.         troca  **$A[i] \leftrightarrow A[j]$**
7. troca  **$A[i+1] \leftrightarrow A[\textit{fim}]$**
8. retorne ***i+1***

0	1
15	19

## (Chamada Recursiva 1)

### Retorno do Partition

**Quicksort**(arranjo ***A***, ***inicio***, ***fim***)

1. Se ***inicio*** < ***fim*** então
2.     ***meio***  $\leftarrow$  ***Partition(A, inicio, fim)***
3.     **Quicksort**(***A***, ***inicio***, ***meio***-1)
4.     **Quicksort**(***A***, ***meio***+1, ***fim***)

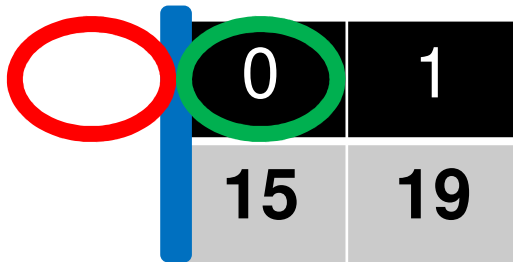
0	1
15	19

## (Chamada Recursiva 1)

### Chamada Recursiva 1.1

**Quicksort**(arranjo *A*, *inicio*, *fim*)

1. Se *inicio* < *fim* então
2.     *meio*  $\leftarrow$  *Partition*(*A*, *inicio*, *fim*)
3.     **Quicksort**(*A*, *inicio*, *meio*-1)
4.     **Quicksort**(*A*, *meio*+1, *fim*)

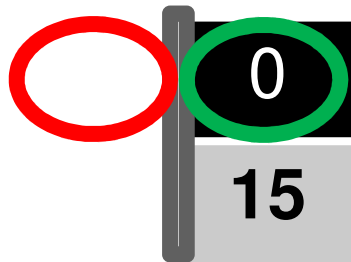


## (Chamada Recursiva 1.1)

### Retornando do Quicksort

**Quicksort**(arranjo **A**, **início**, **fim**)

1. Se **início** < **fim** então
2.     **meio**  $\leftarrow$  **Partition**(**A**, **início**, **fim**)
3.     **Quicksort**(**A**, **início**, **meio**-1)
4.     **Quicksort**(**A**, **meio**+1, **fim**)

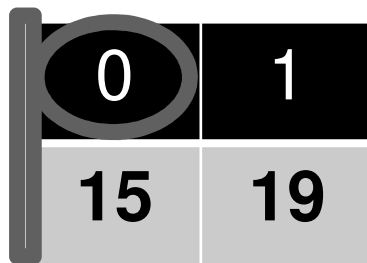


## (Chamada Recursiva 1)

### Retorno do Quicksort (Chamada 1.1)

**Quicksort**(arranjo ***A***, ***inicio***, ***fim***)

1. Se ***inicio*** < ***fim*** então
2.     ***meio***  $\leftarrow$  ***Partition(A, inicio, fim)***
3.     Quicksort(***A***, ***inicio***, ***meio***-1)
4.     Quicksort(***A***, ***meio***+1, ***fim***)



## (Chamada Recursiva 1)

### Chamada Recursiva 1.2

**Quicksort**(arranjo *A*, *inicio*, *fim*)

1. Se *inicio* < *fim* então
2.     *meio*  $\leftarrow$  *Partition*(*A*, *inicio*, *fim*)
3.     Quicksort(*A*, *inicio*, *meio*-1)
4.     Quicksort(*A*, *meio*+1, *fim*)

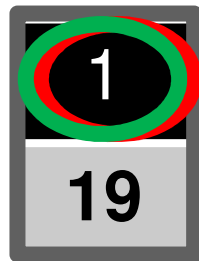
0	1
15	19

## (Chamada Recursiva 1.2)

### Retornando do Quicksort

**Quicksort**(arranjo **A**, **inicio**, **fim**)

1. Se **inicio** < **fim** então
2.     **meio**  $\leftarrow$  **Partition**(**A**, **inicio**, **fim**)
3.     **Quicksort**(**A**, **inicio**, **meio**-1)
4.     **Quicksort**(**A**, **meio**+1, **fim**)





## (Chamada Recursiva 1)

## Retorno do Quicksort (Chamada 1.2)

**Quicksort**(arranjo ***A***, ***inicio***, ***fim***)

1. Se ***inicio*** < ***fim*** então
2.     ***meio***  $\leftarrow$  ***Partition(A, inicio, fim)***
3.     Quicksort(***A***, ***inicio***, ***meio***-1)
4.     Quicksort(***A***, ***meio***+1, ***fim***)

0	1
15	19

## (Chamada Recursiva 1)

### Retornando do Quicksort

**Quicksort**(arranjo **A**, **início**, **fim**)

1. Se **início** < **fim** então
2.     **meio**  $\leftarrow$  **Partition**(**A**, **início**, **fim**)
3.     **Quicksort**(**A**, **início**, **meio**-1)
4.     **Quicksort**(**A**, **meio**+1, **fim**)

0	1
15	19

(Chamada Original)

Retorno do Quicksort (Chamada 1)

**Quicksort**(arranjo ***A***, ***inicio***, ***fim***)

1. Se ***inicio*** < ***fim*** então
2.     ***meio***  $\leftarrow$  ***Partition(A, inicio, fim)***
3.     Quicksort(***A***, ***inicio***, ***meio***-1)
4.     Quicksort(***A***, ***meio***+1, ***fim***)

0	1	2	3	4	5	6	7
15	19	22	26	32	67	81	46

## (Chamada Original)

## Chamada Recursiva 2

**Quicksort**(arranjo ***A***, ***inicio***, ***fim***)

1. Se ***inicio*** < ***fim*** então
2.     ***meio***  $\leftarrow$  ***Partition(A, inicio, fim)***
3.     **Quicksort**(***A***, ***inicio***, ***meio*-1**)
4.     **Quicksort**(***A***, ***meio*+1**, ***fim***)

0	1	2	3	4	5	6	7
15	19	22	26	32	67	81	46

## (Chamada Recursiva 2)

### Chamada do Partition

**Quicksort**(arranjo **A**, **inicio**, **fim**)

1. Se **inicio** < **fim** então
2.     **meio**  $\leftarrow$  **Partition**(**A**, **inicio**, **fim**)
3.     **Quicksort**(**A**, **inicio**, **meio**-1)
4.     **Quicksort**(**A**, **meio**+1, **fim**)

3	4	5	6	7
26	32	67	81	46

## Partition (Quicksort) - Exemplo

**Partition**(arranjo **A**, *inicio*, *fim*)

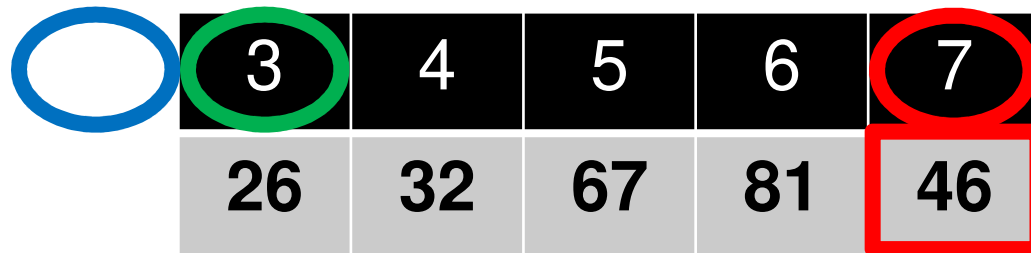
1. **pivo**  $\leftarrow A[\textit{fim}]$
2. **i**  $\leftarrow \textit{inicio} - 1$
3. Para **j**  $\leftarrow \textit{inicio}$  até **fim-1** faça
4.     Se **A[j]**  $\leq$  **pivo** então
5.         **i**  $\leftarrow i + 1$
6.         troca **A[i]**  $\leftrightarrow$  **A[j]**
7. troca **A[i+1]**  $\leftrightarrow$  **A[fim]**
8. retorne **i+1**

3	4	5	6	7
26	32	67	81	46

## Partition (Quicksort) - Exemplo

**Partition**(arranjo **A**, *inicio*, *fim*)

1. **pivo**  $\leftarrow A[\textit{fim}]$
2. **i**  $\leftarrow \textit{inicio} - 1$
3. Para **j**  $\leftarrow \textit{inicio}$  até **fim-1** faça
4.     Se **A[j]**  $\leq$  **pivo** então
5.         **i**  $\leftarrow i + 1$
6.         troca **A[i]**  $\leftrightarrow$  **A[j]**
7. troca **A[i+1]**  $\leftrightarrow$  **A[fim]**
8. retorne **i+1**

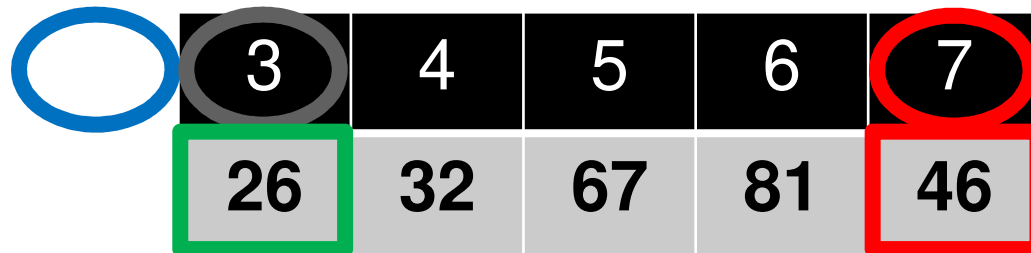


	3	4	5	6	7
	26	32	67	81	46

## Partition (Quicksort) - Exemplo

**Partition**(arranjo  $A$ ,  $inicio$ ,  $fim$ )

1.  $pivo \leftarrow A[fim]$
2.  $i \leftarrow inicio - 1$
3. Para  $j \leftarrow inicio$  até  $fim-1$  faça
4.     Se  $A[j] \leq pivo$  então
5.          $i \leftarrow i + 1$
6.         troca  $A[i] \leftrightarrow A[j]$
7. troca  $A[i+1] \leftrightarrow A[fim]$
8. retorne  $i+1$





## Partition (Quicksort) - Exemplo

**Partition**(arranjo  $A$ ,  $inicio$ ,  $fim$ )

1.  $pivo \leftarrow A[fim]$
2.  $i \leftarrow inicio - 1$
3. Para  $j \leftarrow inicio$  até  $fim-1$  faça
4.     Se  $A[j] \leq pivo$  então
5.          $i \leftarrow i + 1$
6.         troca  $A[i] \leftrightarrow A[j]$
7. troca  $A[i+1] \leftrightarrow A[fim]$
8. retorne  $i+1$

3	4	5	6	7
26	32	67	81	46

## Partition (Quicksort) - Exemplo

**Partition**(arranjo  $A$ ,  $inicio$ ,  $fim$ )

1.  $pivo \leftarrow A[fim]$
2.  $i \leftarrow inicio - 1$
3. Para  $j \leftarrow inicio$  até  $fim-1$  faça
4.     Se  $A[j] \leq pivo$  então
5.          $i \leftarrow i + 1$
6.         troca  $A[i] \leftrightarrow A[j]$
7. troca  $A[i+1] \leftrightarrow A[fim]$
8. retorne  $i+1$

3	4	5	6	7
26	32	67	81	46

## Partition (Quicksort) - Exemplo

**Partition**(arranjo  $A$ ,  $inicio$ ,  $fim$ )

1.  $pivo \leftarrow A[fim]$
2.  $i \leftarrow inicio - 1$
3. Para  $j \leftarrow inicio$  até  $fim-1$  faça
4.     Se  $A[j] \leq pivo$  então
5.          $i \leftarrow i + 1$
6.         troca  $A[i] \leftrightarrow A[j]$
7. troca  $A[i+1] \leftrightarrow A[fim]$
8. retorne  $i+1$

3	4	5	6	7
26	32	67	81	46

## Partition (Quicksort) - Exemplo

**Partition**(arranjo  $A$ ,  $inicio$ ,  $fim$ )

1.  $pivo \leftarrow A[fim]$
2.  $i \leftarrow inicio - 1$
3. Para  $j \leftarrow inicio$  até  $fim-1$  faça
4.     Se  $A[j] \leq pivo$  então
5.          $i \leftarrow i + 1$
6.         troca  $A[i] \leftrightarrow A[j]$
7. troca  $A[i+1] \leftrightarrow A[fim]$
8. retorne  $i+1$

3	4	5	6	7
26	32	67	81	46

## Partition (Quicksort) - Exemplo

**Partition**(arranjo  $A$ ,  $inicio$ ,  $fim$ )

1.  $pivo \leftarrow A[fim]$
2.  $i \leftarrow inicio - 1$
3. Para  $j \leftarrow inicio$  até  $fim-1$  faça
4.     Se  $A[j] \leq pivo$  então
5.          $i \leftarrow i + 1$
6.         troca  $A[i] \leftrightarrow A[j]$
7. troca  $A[i+1] \leftrightarrow A[fim]$
8. retorne  $i+1$

3	4	5	6	7
26	32	67	81	46

## Partition (Quicksort) - Exemplo

**Partition**(arranjo  $A$ ,  $inicio$ ,  $fim$ )

1.  $pivo \leftarrow A[fim]$
2.  $i \leftarrow inicio - 1$
3. Para  $j \leftarrow inicio$  até  $fim-1$  faça
4.     Se  $A[j] \leq pivo$  então
5.          $i \leftarrow i + 1$
6.         troca  $A[i] \leftrightarrow A[j]$
7. troca  $A[i+1] \leftrightarrow A[fim]$
8. retorne  $i+1$

3	4	5	6	7
26	32	67	81	46

## Partition (Quicksort) - Exemplo

**Partition**(arranjo  $A$ ,  $inicio$ ,  $fim$ )

1.  $pivo \leftarrow A[fim]$
2.  $i \leftarrow inicio - 1$
3. Para  $j \leftarrow inicio$  até  $fim-1$  faça
4.     Se  $A[j] \leq pivo$  então
5.          $i \leftarrow i + 1$
6.         troca  $A[i] \leftrightarrow A[j]$
7. troca  $A[i+1] \leftrightarrow A[fim]$
8. retorne  $i+1$

3	4	5	6	7
26	32	67	81	46

## Partition (Quicksort) - Exemplo

**Partition**(arranjo  $A$ ,  $inicio$ ,  $fim$ )

1.  $pivo \leftarrow A[fim]$
2.  $i \leftarrow inicio - 1$
3. Para  $j \leftarrow inicio$  até  $fim-1$  faça
4.     Se  $A[j] \leq pivo$  então
5.          $i \leftarrow i + 1$
6.         troca  $A[i] \leftrightarrow A[j]$
7. troca  $A[i+1] \leftrightarrow A[fim]$
8. retorne  $i+1$

3	4	5	6	7
26	32	67	81	46



## Partition (Quicksort) - Exemplo

**Partition**(arranjo  $A$ ,  $inicio$ ,  $fim$ )

1.  $pivo \leftarrow A[fim]$
2.  $i \leftarrow inicio - 1$
3. Para  $j \leftarrow inicio$  até  $fim-1$  faça
4.     Se  $A[j] \leq pivo$  então
5.          $i \leftarrow i + 1$
6.         troca  $A[i] \leftrightarrow A[j]$
7. troca  $A[i+1] \leftrightarrow A[fim]$
8. retorne  $i+1$

3	4	5	6	7
26	32	46	81	67

## Partition (Quicksort) - Exemplo

**Partition**(arranjo  $A$ ,  $inicio$ ,  $fim$ )

1.  $pivo \leftarrow A[fim]$
2.  $i \leftarrow inicio - 1$
3. Para  $j \leftarrow inicio$  até  $fim-1$  faça
4.     Se  $A[j] \leq pivo$  então
5.          $i \leftarrow i + 1$
6.         troca  $A[i] \leftrightarrow A[j]$
7. troca  $A[i+1] \leftrightarrow A[fim]$
8. retorne  $i+1$

3	4	5	6	7
26	32	46	81	67

## (Chamada Recursiva 2)

### Retorno do Partition

**Quicksort**(arranjo ***A***, ***inicio***, ***fim***)

1. Se ***inicio*** < ***fim*** então
2.     ***meio***  $\leftarrow$  ***Partition(A, inicio, fim)***
3.     **Quicksort**(***A***, ***inicio***, ***meio***-1)
4.     **Quicksort**(***A***, ***meio***+1, ***fim***)

3	4	5	6	7
26	32	46	81	67

## (Chamada Recursiva 2)

### Chamada Recursiva 2.1

**Quicksort**(arranjo *A*, *inicio*, *fim*)

1. Se *inicio* < *fim* então
2.     *meio*  $\leftarrow$  *Partition*(*A*, *inicio*, *fim*)
3.     **Quicksort**(*A*, *inicio*, *meio*-1)
4.     **Quicksort**(*A*, *meio*+1, *fim*)

3	4	5	6	7
26	32	46	81	67

## (Chamada Recursiva 2.1)

### Chamada do Partition

**Quicksort**(arranjo **A**, **inicio**, **fim**)

1. Se **inicio** < **fim** então
2.     **meio**  $\leftarrow$  **Partition**(**A**, **inicio**, **fim**)
3.     **Quicksort**(**A**, **inicio**, **meio**-1)
4.     **Quicksort**(**A**, **meio**+1, **fim**)

3	4
26	32

## Partition (Quicksort) - Exemplo

**Partition**(arranjo **A**, *inicio*, *fim*)

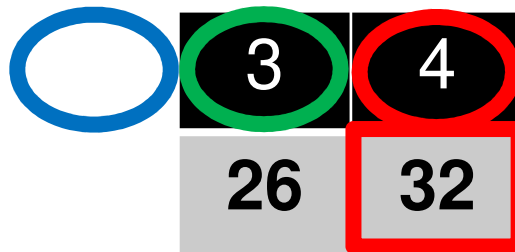
1. **pivo**  $\leftarrow A[\textit{fim}]$
2. **i**  $\leftarrow \textit{inicio} - 1$
3. Para **j**  $\leftarrow \textit{inicio}$  até **fim-1** faça
4.     Se **A[j]**  $\leq$  **pivo** então
5.         **i**  $\leftarrow i + 1$
6.         troca **A[i]**  $\leftrightarrow$  **A[j]**
7. troca **A[i+1]**  $\leftrightarrow$  **A[fim]**
8. retorne **i+1**

3	4
26	32

## Partition (Quicksort) - Exemplo

**Partition**(arranjo **A**, *inicio*, *fim*)

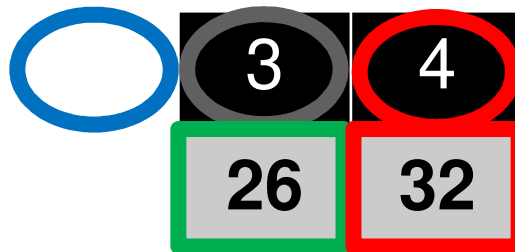
1. **pivo**  $\leftarrow A[\text{fim}]$
2. **i**  $\leftarrow \text{inicio} - 1$
3. Para **j**  $\leftarrow \text{inicio}$  até **fim-1** faça
4.     Se **A[j]**  $\leq$  **pivo** então
5.         **i**  $\leftarrow i + 1$
6.         troca **A[i]**  $\leftrightarrow$  **A[j]**
7. troca **A[i+1]**  $\leftrightarrow$  **A[fim]**
8. retorne **i+1**



## Partition (Quicksort) - Exemplo

**Partition**(arranjo  $A$ ,  $inicio$ ,  $fim$ )

1.  $pivo \leftarrow A[fim]$
2.  $i \leftarrow inicio - 1$
3. Para  $j \leftarrow inicio$  até  $fim-1$  faça
4.     Se  $A[j] \leq pivo$  então
5.          $i \leftarrow i + 1$
6.         troca  $A[i] \leftrightarrow A[j]$
7. troca  $A[i+1] \leftrightarrow A[fim]$
8. retorne  $i+1$

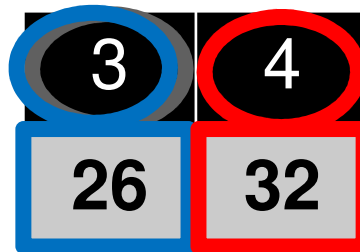




## Partition (Quicksort) - Exemplo

**Partition**(arranjo  $A$ ,  $inicio$ ,  $fim$ )

1.  $pivo \leftarrow A[fim]$
2.  $i \leftarrow inicio - 1$
3. Para  $j \leftarrow inicio$  até  $fim-1$  faça
4.     Se  $A[j] \leq pivo$  então
5.          $i \leftarrow i + 1$
6.         troca  $A[i] \leftrightarrow A[j]$
7. troca  $A[i+1] \leftrightarrow A[fim]$
8. retorne  $i+1$



## Partition (Quicksort) - Exemplo

**Partition**(arranjo  $A$ ,  $inicio$ ,  $fim$ )

1.  $pivo \leftarrow A[fim]$
2.  $i \leftarrow inicio - 1$
3. Para  $j \leftarrow inicio$  até  $fim-1$  faça
4.     Se  $A[j] \leq pivo$  então
5.          $i \leftarrow i + 1$
6.         troca  $A[i] \leftrightarrow A[j]$
7. troca  $A[i+1] \leftrightarrow A[fim]$
8. retorne  $i+1$



## Partition (Quicksort) - Exemplo

**Partition**(arranjo  $A$ ,  $inicio$ ,  $fim$ )

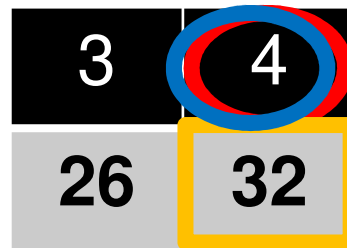
1.  $pivo \leftarrow A[fim]$
2.  $i \leftarrow inicio - 1$
3. Para  $j \leftarrow inicio$  até  $fim-1$  faça
4.     Se  $A[j] \leq pivo$  então
5.          $i \leftarrow i + 1$
6.         troca  $A[i] \leftrightarrow A[j]$
7. troca  $A[i+1] \leftrightarrow A[fim]$
8. retorne  $i+1$

3	4
26	32

## Partition (Quicksort) - Exemplo

**Partition**(arranjo  $A$ ,  $inicio$ ,  $fim$ )

1.  $pivo \leftarrow A[fim]$
2.  $i \leftarrow inicio - 1$
3. Para  $j \leftarrow inicio$  até  $fim-1$  faça
4.     Se  $A[j] \leq pivo$  então
5.          $i \leftarrow i + 1$
6.         troca  $A[i] \leftrightarrow A[j]$
7. troca  $A[i+1] \leftrightarrow A[fim]$
8. retorne  $i+1$



## Partition (Quicksort) - Exemplo

**Partition**(arranjo  $A$ ,  $inicio$ ,  $fim$ )

1.  $pivo \leftarrow A[fim]$
2.  $i \leftarrow inicio - 1$
3. Para  $j \leftarrow inicio$  até  $fim-1$  faça
4.     Se  $A[j] \leq pivo$  então
5.          $i \leftarrow i + 1$
6.         troca  $A[i] \leftrightarrow A[j]$
7. troca  $A[i+1] \leftrightarrow A[fim]$
8. retorne  $i+1$

3	4
26	32

## (Chamada Recursiva 2.1)

### Retorno do Partition

**Quicksort**(arranjo ***A***, ***inicio***, ***fim***)

1. Se ***inicio*** < ***fim*** então
2.     ***meio***  $\leftarrow$  ***Partition(A, inicio, fim)***
3.     Quicksort(***A***, ***inicio***, ***meio***-1)
4.     Quicksort(***A***, ***meio***+1, ***fim***)

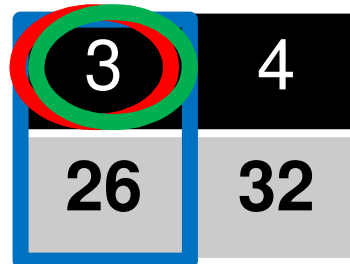
3	4
26	32

## (Chamada Recursiva 2.1)

### Chamada Recursiva 2.1.1

**Quicksort**(arranjo ***A***, ***inicio***, ***fim***)

1. Se ***inicio*** < ***fim*** então
2.     ***meio***  $\leftarrow$  ***Partition***(***A***, ***inicio***, ***fim***)
3.     **Quicksort**(***A***, ***inicio***, ***meio*-1**)
4.     **Quicksort**(***A***, ***meio*+1**, ***fim***)

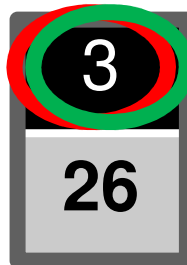


## (Chamada Recursiva 2.1.1)

### Retornando do Quicksort

**Quicksort**(arranjo **A**, *inicio*, *fim*)

1. Se *inicio* < *fim* então
2.     *meio*  $\leftarrow$  **Partition**(**A**, *inicio*, *fim*)
3.     Quicksort(**A**, *inicio*, *meio*-1)
4.     Quicksort(**A**, *meio*+1, *fim*)





## (Chamada Recursiva 2.1)

### Retorno do Quicksort (Chamada 2.1.1)

**Quicksort**(arranjo ***A***, ***inicio***, ***fim***)

1. Se ***inicio*** < ***fim*** então
2.     ***meio***  $\leftarrow$  ***Partition(A, inicio, fim)***
3.     Quicksort(***A***, ***inicio***, ***meio***-1)
4.     Quicksort(***A***, ***meio***+1, ***fim***)

3	4
26	32

## (Chamada Recursiva 2.1)

### Chamada recursiva 2.1.2

**Quicksort**(arranjo *A*, *inicio*, *fim*)

1. Se *inicio* < *fim* então
2.     *meio*  $\leftarrow$  *Partition*(*A*, *inicio*, *fim*)
3.     Quicksort(*A*, *inicio*, *meio*-1)
4.     Quicksort(*A*, *meio*+1, *fim*)

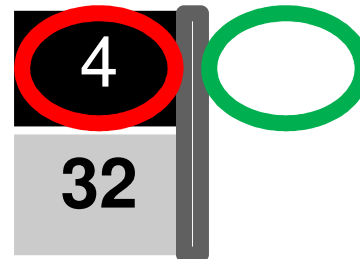


## (Chamada Recursiva 2.1.2)

### Retornando do Quicksort

**Quicksort**(arranjo **A**, **inicio**, **fim**)

1. Se **inicio** < **fim** então
2.     **meio**  $\leftarrow$  **Partition**(**A**, **inicio**, **fim**)
3.     **Quicksort**(**A**, **inicio**, **meio**-1)
4.     **Quicksort**(**A**, **meio**+1, **fim**)



## (Chamada Recursiva 2.1)

### Retorno do Quicksort (Chamada 2.1.2)

**Quicksort**(arranjo ***A***, ***inicio***, ***fim***)

1. Se ***inicio*** < ***fim*** então
2.     ***meio***  $\leftarrow$  ***Partition(A, inicio, fim)***
3.     Quicksort(***A***, ***inicio***, ***meio***-1)
4.     Quicksort(***A***, ***meio***+1, ***fim***)

3	4
26	32

## (Chamada Recursiva 2.1)

### Retornando do Quicksort

**Quicksort**(arranjo **A**, **inicio**, **fim**)

1. Se **inicio** < **fim** então
2.     **meio**  $\leftarrow$  **Partition**(**A**, **inicio**, **fim**)
3.     Quicksort(**A**, **inicio**, **meio**-1)
4.     Quicksort(**A**, **meio**+1, **fim**)

3	4
26	32

## (Chamada Recursiva 2)

### Retorno do Quicksort (Chamada 2.1)

**Quicksort**(arranjo ***A***, ***inicio***, ***fim***)

1. Se ***inicio*** < ***fim*** então
2.     ***meio***  $\leftarrow$  ***Partition(A, inicio, fim)***
3.     Quicksort(***A***, ***inicio***, ***meio***-1)
4.     Quicksort(***A***, ***meio***+1, ***fim***)

3	4	5	6	7
26	32	46	81	67

## (Chamada Recursiva 2)

### Chamada Recursiva 2.2

**Quicksort**(arranjo ***A***, ***inicio***, ***fim***)

1. Se ***inicio*** < ***fim*** então
2.     ***meio***  $\leftarrow$  ***Partition***(***A***, ***inicio***, ***fim***)
3.     **Quicksort**(***A***, ***inicio***, ***meio***-1)
4.     **Quicksort**(***A***, ***meio***+1, ***fim***)

3	4	5	6	7
26	32	46	81	67

## (Chamada Recursiva 2.2)

### Chamada do Partition

**Quicksort**(arranjo **A**, **inicio**, **fim**)

1. Se **inicio** < **fim** então
2.     **meio**  $\leftarrow$  **Partition**(**A**, **inicio**, **fim**)
3.     **Quicksort**(**A**, **inicio**, **meio**-1)
4.     **Quicksort**(**A**, **meio**+1, **fim**)

6	7
81	67



## Partition (Quicksort) - Exemplo

**Partition**(arranjo **A**, *inicio*, *fim*)

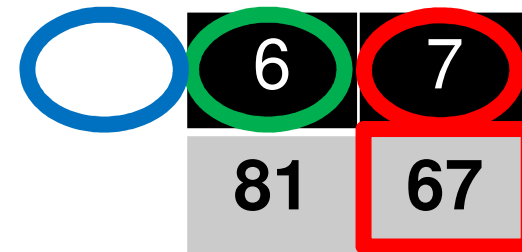
1. **pivo**  $\leftarrow A[\textit{fim}]$
2. **i**  $\leftarrow \textit{inicio} - 1$
3. Para **j**  $\leftarrow \textit{inicio}$  até **fim-1** faça
4.     Se **A[j]**  $\leq$  **pivo** então
5.         **i**  $\leftarrow i + 1$
6.         troca **A[i]**  $\leftrightarrow$  **A[j]**
7. troca **A[i+1]**  $\leftrightarrow$  **A[fim]**
8. retorne **i+1**

6	7
81	67

## Partition (Quicksort) - Exemplo

**Partition**(arranjo **A**, *inicio*, *fim*)

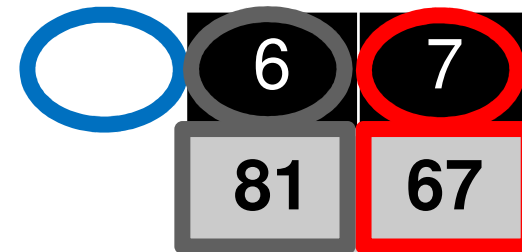
1. *pivo*  $\leftarrow A[fim]$
2. *i*  $\leftarrow inicio - 1$
3. Para *j*  $\leftarrow inicio$  até *fim-1* faça
4.     Se  $A[j] \leq pivo$  então
5.         *i*  $\leftarrow i + 1$
6.         troca  $A[i] \leftrightarrow A[j]$
7. troca  $A[i+1] \leftrightarrow A[fim]$
8. retorne *i+1*



## Partition (Quicksort) - Exemplo

**Partition**(arranjo  $A$ ,  $inicio$ ,  $fim$ )

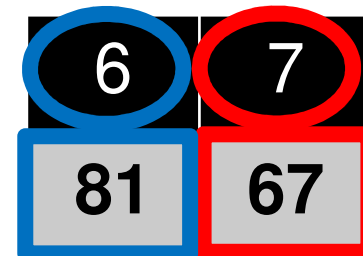
1.  $pivo \leftarrow A[fim]$
2.  $i \leftarrow inicio - 1$
3. Para  $j \leftarrow inicio$  até  $fim-1$  faça
4.     Se  $A[j] \leq pivo$  então
5.          $i \leftarrow i + 1$
6.         troca  $A[i] \leftrightarrow A[j]$
7. troca  $A[i+1] \leftrightarrow A[fim]$
8. retorne  $i+1$



## Partition (Quicksort) - Exemplo

**Partition**(arranjo  $A$ ,  $inicio$ ,  $fim$ )

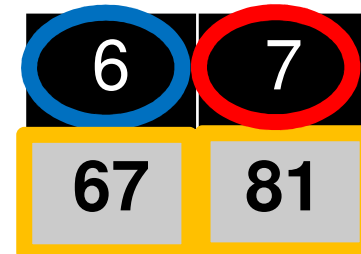
1.  $pivo \leftarrow A[fim]$
2.  $i \leftarrow inicio - 1$
3. Para  $j \leftarrow inicio$  até  $fim-1$  faça
4.     Se  $A[j] \leq pivo$  então
5.          $i \leftarrow i + 1$
6.         troca  $A[i] \leftrightarrow A[j]$
7. troca  $A[i+1] \leftrightarrow A[fim]$
8. retorne  $i+1$



## Partition (Quicksort) - Exemplo

**Partition**(arranjo  $A$ ,  $inicio$ ,  $fim$ )

1.  $pivo \leftarrow A[fim]$
2.  $i \leftarrow inicio - 1$
3. Para  $j \leftarrow inicio$  até  $fim-1$  faça
4.     Se  $A[j] \leq pivo$  então
5.          $i \leftarrow i + 1$
6.         troca  $A[i] \leftrightarrow A[j]$
7. troca  $A[i+1] \leftrightarrow A[fim]$
8. retorne  $i+1$



## Partition (Quicksort) - Exemplo

**Partition**(arranjo *A*, *inicio*, *fim*)

1. ***pivo***  $\leftarrow A[\textit{fim}]$
2. ***i***  $\leftarrow \textit{inicio} - 1$
3. Para ***j***  $\leftarrow \textit{inicio}$  até ***fim-1*** faça
4.     Se ***A[j]***  $\leq \textit{pivo}$  então
5.         ***i***  $\leftarrow i + 1$
6.         troca ***A[i]***  $\leftrightarrow$  ***A[j]***
7. troca ***A[i+1]***  $\leftrightarrow$  ***A[fim]***
8. retorne ***i+1***

6	7
67	81

## (Chamada Recursiva 2.2)

### Retorno do Partition

**Quicksort**(arranjo ***A***, ***inicio***, ***fim***)

1. Se ***inicio*** < ***fim*** então
2.     ***meio***  $\leftarrow$  ***Partition(A, inicio, fim)***
3.     **Quicksort**(***A***, ***inicio***, ***meio***-1)
4.     **Quicksort**(***A***, ***meio***+1, ***fim***)

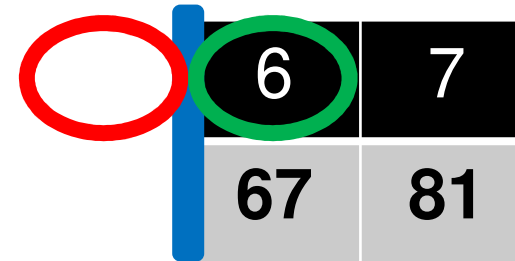
6	7
67	81

## (Chamada Recursiva 2.2)

### Chamada Recursiva 2.2.1

**Quicksort**(arranjo ***A***, ***inicio***, ***fim***)

1. Se ***inicio*** < ***fim*** então
2.     ***meio***  $\leftarrow$  ***Partition***(***A***, ***inicio***, ***fim***)
3.     **Quicksort**(***A***, ***inicio***, ***meio*-1**)
4.     **Quicksort**(***A***, ***meio*+1**, ***fim***)



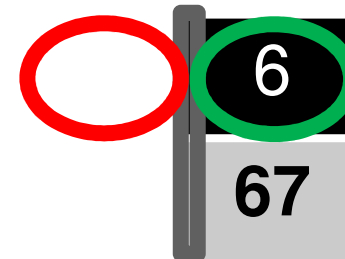


## (Chamada Recursiva 2.2.1)

### Retornando do Quicksort

**Quicksort**(arranjo **A**, **início**, **fim**)

1. Se **início** < **fim** então
2.     **meio**  $\leftarrow$  **Partition**(**A**, **início**, **fim**)
3.     **Quicksort**(**A**, **início**, **meio**-1)
4.     **Quicksort**(**A**, **meio**+1, **fim**)

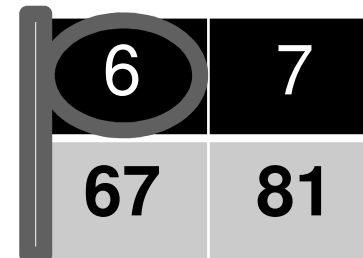


## (Chamada Recursiva 2.2)

### Retorno do Quicksort (Chamada 2.2.1)

**Quicksort**(arranjo ***A***, ***inicio***, ***fim***)

1. Se ***inicio*** < ***fim*** então
2.     ***meio***  $\leftarrow$  ***Partition(A, inicio, fim)***
3.     Quicksort(***A***, ***inicio***, ***meio***-1)
4.     Quicksort(***A***, ***meio***+1, ***fim***)



A diagram showing a 2x2 array of numbers. The top row contains 6 and 7. The bottom row contains 67 and 81. The number 6 in the top-left cell is circled with a thick grey line, indicating it is the pivot element. To the left of the array is a vertical grey bar.

6	7
67	81

## (Chamada Recursiva 2.2)

### Chamada Recursiva 2.2.2

**Quicksort**(arranjo ***A***, ***inicio***, ***fim***)

1. Se ***inicio*** < ***fim*** então
2.     ***meio***  $\leftarrow$  ***Partition***(***A***, ***inicio***, ***fim***)
3.     **Quicksort**(***A***, ***inicio***, ***meio***-1)
4.     **Quicksort**(***A***, ***meio***+1, ***fim***)

6	7
67	81

## (Chamada Recursiva 2.2.2)

### Retornando do Quicksort.

**Quicksort**(arranjo **A**, **inicio**, **fim**)

1. Se **inicio** < **fim** então
2.     **meio**  $\leftarrow$  **Partition**(**A**, **inicio**, **fim**)
3.     **Quicksort**(**A**, **inicio**, **meio**-1)
4.     **Quicksort**(**A**, **meio**+1, **fim**)



## (Chamada Recursiva 2.2)

### Retorno do Quicksort (Chamada 2.2.2)

**Quicksort**(arranjo ***A***, ***inicio***, ***fim***)

1. Se ***inicio*** < ***fim*** então
2.     ***meio***  $\leftarrow$  ***Partition(A, inicio, fim)***
3.     Quicksort(***A***, ***inicio***, ***meio***-1)
4.     Quicksort(***A***, ***meio***+1, ***fim***)

6	7
67	81

## (Chamada Recursiva 2.2)

### Retornando do Quicksort

**Quicksort**(arranjo **A**, **início**, **fim**)

1. Se **início** < **fim** então
2.     **meio**  $\leftarrow$  **Partition**(**A**, **início**, **fim**)
3.     **Quicksort**(**A**, **início**, **meio**-1)
4.     **Quicksort**(**A**, **meio**+1, **fim**)

6	7
81	67

## (Chamada Recursiva 2)

### Retorno do Quicksort (Chamada 2.2)

**Quicksort**(arranjo ***A***, ***inicio***, ***fim***)

1. Se ***inicio*** < ***fim*** então
2.     ***meio***  $\leftarrow$  ***Partition(A, inicio, fim)***
3.     Quicksort(***A***, ***inicio***, ***meio***-1)
4.     Quicksort(***A***, ***meio***+1, ***fim***)

3	4	5	6	7
26	32	46	81	67

## (Chamada Recursiva 2)

### Retornando do Quicksort

**Quicksort**(arranjo **A**, **início**, **fim**)

1. Se **início** < **fim** então
2.     **meio**  $\leftarrow$  **Partition**(**A**, **início**, **fim**)
3.     **Quicksort**(**A**, **início**, **meio**-1)
4.     **Quicksort**(**A**, **meio**+1, **fim**)

3	4	5	6	7
26	32	46	81	67



(Chamada Original)

Retorno do Quicksort (Chamada 2)

**Quicksort**(arranjo ***A***, ***inicio***, ***fim***)

1. Se ***inicio*** < ***fim*** então
2.     ***meio***  $\leftarrow$  ***Partition(A, inicio, fim)***
3.     Quicksort(***A***, ***inicio***, ***meio***-1)
4.     Quicksort(***A***, ***meio***+1, ***fim***)

0	1	2	3	4	5	6	7
15	19	22	26	32	67	81	46

(Chamada Original)

Retornando do Quicksort

**Quicksort**(arranjo **A**, **início**, **fim**)

1. Se **início** < **fim** então
2.     **meio**  $\leftarrow$  **Partition**(**A**, **início**, **fim**)
3.     **Quicksort**(**A**, **início**, **meio**-1)
4.     **Quicksort**(**A**, **meio**+1, **fim**)

0	1	2	3	4	5	6	7
15	19	22	26	32	46	67	81

## VETOR ORDENADO!

**Quicksort**(arranjo *A*, *inicio*, *fim*)

1. Se *inicio* < *fim* então
2.     *meio*  $\leftarrow$  *Partition*(*A*, *inicio*, *fim*)
3.     Quicksort(*A*, *inicio*, *meio*-1)
4.     Quicksort(*A*, *meio*+1, *fim*)

0	1	2	3	4	5	6	7
15	19	22	26	32	46	67	81

# Estrutura de Dados II

Prof. Me. Pietro M. de Oliveira