

Estrutura de Dados I

Algoritmos básicos de Ordenação: **Quick-Sort**

Prof. Rodrigo Minetto

Universidade Tecnológica Federal do Paraná

Sumário

- 1 Introdução
- 2 Particionamento
- 3 Quick-Sort
- 4 Conquista (esquerda)
- 5 Quick-Sort
- 6 Conquista (direita)
- 7 Complexidade

Quick Sort

- Quicksort é o algoritmo de ordenação mais rápido que se conhece para uma ampla variedade de situações;
- É provavelmente o algoritmo mais utilizado para ordenação;
- Inventado em 1960 (Tony Hoare).
- Algoritmo de ordenação não estável.

Problema: ordenar um vetor em ordem crescente.

- **Entrada:** um array $A[0 \dots n-1]$
- **Saída:** array $A[0 \dots n-1]$
rearranjado em ordem crescente

Quick Sort - Ideia

Segue o paradigma de **divisão-e-conquista**:

Divisão: divida o array A em dois subvetores

$A[\text{esq} \dots \text{pivot} - 1]$ e $A[\text{pivot} + 1 \dots \text{dir}]$ tais que:



$$A[l \dots \text{pivot} - 1] \leq A[\text{pivot}] < A[\text{pivot} + 1 \dots r]$$

Conquista: ordene os dois subarrays

recursivamente utilizando o **Quick Sort**.

Partição

Problema: Rearranjar um array $A[\textit{esq} \dots \textit{dir}]$ e devolver um índice *pivot* (p), onde $e \leq p \leq d$, tal que:

$$A[e \dots p - 1] \leq A[p] < A[p + 1 \dots d] \quad (1)$$

Exemplo: entrada

<i>e</i>					<i>d</i>				
99	33	55	77	11	33	88	66	22	44

Saída:

<i>e</i>			<i>p</i>			<i>d</i>			
33	11	33	22	44	55	88	66	77	99

Sumário

- 1 Introdução
- 2 Particionamento**
- 3 Quick-Sort
- 4 Conquista (esquerda)
- 5 Quick-Sort
- 6 Conquista (direita)
- 7 Complexidade

Particione: funcionamento

99	33	55	77	11	33	88	66	22	44
----	----	----	----	----	----	----	----	----	----

e

d

- 1: PARTICIONE(A, e, d)
- 2: $p \leftarrow A[d];$ $\{p \text{ é o "pivô"}\}$
- 3: $i \leftarrow e - 1;$
- 4: **Para** $j \leftarrow e$ **até** $d - 1$ **faça**
- 5: **Se** $A[j] \leq p$ **então**
- 6: $i \leftarrow i + 1;$
- 7: $A[i] \leftrightarrow A[j];$
- 8: **fim**
- 9: **fim**
- 10: $A[i + 1] \leftrightarrow A[d];$
- 11: **Retorne** $i + 1;$

Particione: funcionamento

99	33	55	77	11	33	88	66	22	44
									p
									d

- 1: PARTICIONE(A, e, d)
- 2: $\triangleright p \leftarrow A[d]; \quad \{p \text{ é o "pivô"}\}$
- 3: $i \leftarrow e - 1;$
- 4: **Para** $j \leftarrow e$ até $d - 1$ **faça**
- 5: **Se** $A[j] \leq p$ **então**
- 6: $i \leftarrow i + 1;$
- 7: $A[i] \leftrightarrow A[j];$
- 8: **fim**
- 9: **fim**
- 10: $A[i + 1] \leftrightarrow A[d];$
- 11: **Retorne** $i + 1;$

Particione: funcionamento

i

p

99	33	55	77	11	33	88	66	22	44
----	----	----	----	----	----	----	----	----	----

e

d

- 1: PARTICIONE(A, e, d)
- 2: $p \leftarrow A[d]$; $\{p \text{ é o "pivô"}\}$
- 3: $\triangleright i \leftarrow e - 1$;
- 4: **Para** $j \leftarrow e$ **até** $d - 1$ **faça**
- 5: **Se** $A[j] \leq p$ **então**
- 6: $i \leftarrow i + 1$;
- 7: $A[i] \leftrightarrow A[j]$;
- 8: **fim**
- 9: **fim**
- 10: $A[i + 1] \leftrightarrow A[d]$;
- 11: **Retorne** $i + 1$;

Particione: funcionamento

<i>i</i>	<i>j</i>								<i>p</i>
99	33	55	77	11	33	88	66	22	44
<i>e</i>									<i>d</i>

- 1: PARTICIONE(A, e, d)
- 2: $p \leftarrow A[d]$; { p é o “pivô”}
- 3: $i \leftarrow e - 1$;
- 4: ▷ Para $j \leftarrow e$ até $d - 1$ faça
- 5: Se $A[j] \leq p$ então
- 6: $i \leftarrow i + 1$;
- 7: $A[i] \leftrightarrow A[j]$;
- 8: fim
- 9: fim
- 10: $A[i + 1] \leftrightarrow A[d]$;
- 11: Retorne $i + 1$;

Particione: funcionamento

i	j								p
99	33	55	77	11	33	88	66	22	44
e									d

- 1: PARTICIONE(A, e, d)
- 2: $p \leftarrow A[d];$ $\{p \text{ é o "pivô"}\}$
- 3: $i \leftarrow e - 1;$
- 4: **Para** $j \leftarrow e$ até $d - 1$ **faça**
- 5: \triangleright **Se** $A[j] \leq p$ **então**
- 6: $i \leftarrow i + 1;$
- 7: $A[i] \leftrightarrow A[j];$
- 8: **fim**
- 9: **fim**
- 10: $A[i + 1] \leftrightarrow A[d];$
- 11: **Retorne** $i + 1;$

Particione: funcionamento

<i>i</i>	<i>j</i>								<i>p</i>
99	33	55	77	11	33	88	66	22	44
<i>e</i>									<i>d</i>

- 1: PARTICIONE(A, e, d)
- 2: $p \leftarrow A[d];$ $\{p \text{ é o "pivô"}\}$
- 3: $i \leftarrow e - 1;$
- 4: ▷ Para $j \leftarrow e$ até $d - 1$ faça
- 5: Se $A[j] \leq p$ então
- 6: $i \leftarrow i + 1;$
- 7: $A[i] \leftrightarrow A[j];$
- 8: fim
- 9: fim
- 10: $A[i + 1] \leftrightarrow A[d];$
- 11: Retorne $i + 1;$

Particione: funcionamento

i	j								p
99	33	55	77	11	33	88	66	22	44
e									d

- 1: PARTICIONE(A, e, d)
- 2: $p \leftarrow A[d]$; { p é o “pivô”}
- 3: $i \leftarrow e - 1$;
- 4: **Para** $j \leftarrow e$ até $d - 1$ **faça**
- 5: \triangleright **Se** $A[j] \leq p$ **então**
- 6: $i \leftarrow i + 1$;
- 7: $A[i] \leftrightarrow A[j]$;
- 8: **fim**
- 9: **fim**
- 10: $A[i + 1] \leftrightarrow A[d]$;
- 11: **Retorne** $i + 1$;

Particione: funcionamento

<i>i</i>	<i>j</i>								<i>p</i>
99	33	55	77	11	33	88	66	22	44
<i>e</i>									<i>d</i>

- 1: PARTICIONE(A, e, d)
- 2: $p \leftarrow A[d];$ $\{p \text{ é o "pivô"}\}$
- 3: $i \leftarrow e - 1;$
- 4: **Para** $j \leftarrow e$ até $d - 1$ **faça**
- 5: **Se** $A[j] \leq p$ **então**
- 6: $\triangleright i \leftarrow i + 1;$
- 7: $A[i] \leftrightarrow A[j];$
- 8: **fim**
- 9: **fim**
- 10: $A[i + 1] \leftrightarrow A[d];$
- 11: **Retorne** $i + 1;$

Particione: funcionamento

<i>i</i>	<i>j</i>								<i>p</i>
33	99	55	77	11	33	88	66	22	44
<i>e</i>									<i>d</i>

- 1: PARTICIONE(*A*, *e*, *d*)
- 2: *p* ← *A*[*d*]; {*p* é o “pivô”}
- 3: *i* ← *e* − 1;
- 4: **Para** *j* ← *e* até *d* − 1 **faça**
- 5: **Se** *A*[*j*] ≤ *p* **então**
- 6: *i* ← *i* + 1;
- 7: *A*[*i*] ↔ *A*[*j*];
- 8: **fim**
- 9: **fim**
- 10: *A*[*i* + 1] ↔ *A*[*d*];
- 11: **Retorne** *i* + 1;

Particione: funcionamento

<i>i</i>		<i>j</i>							<i>p</i>
33	99	55	77	11	33	88	66	22	44
<i>e</i>									<i>d</i>

- 1: PARTICIONE(A, e, d)
- 2: $p \leftarrow A[d];$ { p é o “pivô”}
- 3: $i \leftarrow e - 1;$
- 4: ▷ Para $j \leftarrow e$ até $d - 1$ faça
- 5: Se $A[j] \leq p$ então
- 6: $i \leftarrow i + 1;$
- 7: $A[i] \leftrightarrow A[j];$
- 8: fim
- 9: fim
- 10: $A[i + 1] \leftrightarrow A[d];$
- 11: Retorne $i + 1;$

Particione: funcionamento

<i>i</i>		<i>j</i>							<i>p</i>
33	99	55	77	11	33	88	66	22	44
<i>e</i>									<i>d</i>

- 1: PARTICIONE(A, e, d)
- 2: $p \leftarrow A[d];$ $\{p \text{ é o "pivô"}\}$
- 3: $i \leftarrow e - 1;$
- 4: **Para** $j \leftarrow e$ até $d - 1$ **faça**
- 5: \triangleright **Se** $A[j] \leq p$ **então**
- 6: $i \leftarrow i + 1;$
- 7: $A[i] \leftrightarrow A[j];$
- 8: **fim**
- 9: **fim**
- 10: $A[i + 1] \leftrightarrow A[d];$
- 11: **Retorne** $i + 1;$

Particione: funcionamento

<i>i</i>		<i>j</i>				<i>p</i>			
33	99	55	77	11	33	88	66	22	44
<i>e</i>		<i>d</i>							

- 1: PARTICIONE(A, e, d)
- 2: $p \leftarrow A[d];$ $\{p \text{ é o "pivô"}\}$
- 3: $i \leftarrow e - 1;$
- 4: ▷ Para $j \leftarrow e$ até $d - 1$ faça
- 5: Se $A[j] \leq p$ então
- 6: $i \leftarrow i + 1;$
- 7: $A[i] \leftrightarrow A[j];$
- 8: fim
- 9: fim
- 10: $A[i + 1] \leftrightarrow A[d];$
- 11: Retorne $i + 1;$

Particione: funcionamento

<i>i</i>		<i>j</i>			<i>p</i>					<i>d</i>
33	99	55	77	11	33	88	66	22	44	
<i>e</i>										<i>d</i>

- 1: PARTICIONE(A, e, d)
- 2: $p \leftarrow A[d];$ $\{p \text{ é o "pivô"}\}$
- 3: $i \leftarrow e - 1;$
- 4: **Para** $j \leftarrow e$ até $d - 1$ **faça**
- 5: \triangleright **Se** $A[j] \leq p$ **então**
- 6: $i \leftarrow i + 1;$
- 7: $A[i] \leftrightarrow A[j];$
- 8: **fim**
- 9: **fim**
- 10: $A[i + 1] \leftrightarrow A[d];$
- 11: **Retorne** $i + 1;$

Particione: funcionamento

<i>i</i>		<i>j</i>				<i>p</i>			
33	99	55	77	11	33	88	66	22	44
<i>e</i>		<i>d</i>							

- 1: PARTICIONE(A, e, d)
- 2: $p \leftarrow A[d];$ $\{p \text{ é o "pivô"}\}$
- 3: $i \leftarrow e - 1;$
- 4: ▷ Para $j \leftarrow e$ até $d - 1$ faça
- 5: Se $A[j] \leq p$ então
- 6: $i \leftarrow i + 1;$
- 7: $A[i] \leftrightarrow A[j];$
- 8: fim
- 9: fim
- 10: $A[i + 1] \leftrightarrow A[d];$
- 11: Retorne $i + 1;$

Particione: funcionamento

<i>i</i>		<i>j</i>				<i>p</i>			
33	99	55	77	11	33	88	66	22	44
<i>e</i>		<i>d</i>							

- 1: PARTICIONE(A, e, d)
- 2: $p \leftarrow A[d];$ $\{p \text{ é o "pivô"}\}$
- 3: $i \leftarrow e - 1;$
- 4: **Para** $j \leftarrow e$ até $d - 1$ **faça**
- 5: \triangleright **Se** $A[j] \leq p$ **então**
- 6: $i \leftarrow i + 1;$
- 7: $A[i] \leftrightarrow A[j];$
- 8: **fim**
- 9: **fim**
- 10: $A[i + 1] \leftrightarrow A[d];$
- 11: **Retorne** $i + 1;$

Particione: funcionamento

<i>i</i>		<i>j</i>				<i>p</i>			
33	99	55	77	11	33	88	66	22	44
<i>e</i>									
								<i>d</i>	

- 1: PARTICIONE(A, e, d)
- 2: $p \leftarrow A[d];$ $\{p \text{ é o "pivô"}\}$
- 3: $i \leftarrow e - 1;$
- 4: **Para** $j \leftarrow e$ até $d - 1$ **faça**
- 5: **Se** $A[j] \leq p$ **então**
- 6: $\triangleright i \leftarrow i + 1;$
- 7: $A[i] \leftrightarrow A[j];$
- 8: **fim**
- 9: **fim**
- 10: $A[i + 1] \leftrightarrow A[d];$
- 11: **Retorne** $i + 1;$

Particione: funcionamento

	i			j					p
33	11	55	77	99	33	88	66	22	44
e									d

- 1: PARTICIONE(A, e, d)
- 2: $p \leftarrow A[d]$; { p é o “pivô”}
- 3: $i \leftarrow e - 1$;
- 4: **Para** $j \leftarrow e$ até $d - 1$ **faça**
- 5: **Se** $A[j] \leq p$ **então**
- 6: $i \leftarrow i + 1$;
- 7: $\triangleright A[i] \leftrightarrow A[j]$;
- 8: **fim**
- 9: **fim**
- 10: $A[i + 1] \leftrightarrow A[d]$;
- 11: **Retorne** $i + 1$;

Particione: funcionamento

<i>i</i>		<i>j</i>				<i>p</i>			
33	11	55	77	99	33	88	66	22	44
<i>e</i>									<i>d</i>

- 1: PARTICIONE(A, e, d)
- 2: $p \leftarrow A[d]; \quad \{p \text{ é o "pivô"}\}$
- 3: $i \leftarrow e - 1;$
- 4: ▷ Para $j \leftarrow e$ até $d - 1$ faça
- 5: Se $A[j] \leq p$ então
- 6: $i \leftarrow i + 1;$
- 7: $A[i] \leftrightarrow A[j];$
- 8: fim
- 9: fim
- 10: $A[i + 1] \leftrightarrow A[d];$
- 11: Retorne $i + 1;$

Particione: funcionamento

	i				j				p
33	11	55	77	99	33	88	66	22	44
e									d

- 1: PARTICIONE(A, e, d)
- 2: $p \leftarrow A[d]$; $\{p \text{ é o "pivô"}\}$
- 3: $i \leftarrow e - 1$;
- 4: **Para** $j \leftarrow e$ **até** $d - 1$ **faça**
- 5: \triangleright **Se** $A[j] \leq p$ **então**
- 6: $i \leftarrow i + 1$;
- 7: $A[i] \leftrightarrow A[j]$;
- 8: **fim**
- 9: **fim**
- 10: $A[i + 1] \leftrightarrow A[d]$;
- 11: **Retorne** $i + 1$;

Particione: funcionamento

		i			j				p
33	11	55	77	99	33	88	66	22	44
e									d

- 1: PARTICIONE(A, e, d)
- 2: $p \leftarrow A[d]$; { p é o “pivô”}
- 3: $i \leftarrow e - 1$;
- 4: **Para** $j \leftarrow e$ até $d - 1$ **faça**
- 5: **Se** $A[j] \leq p$ **então**
- 6: $\triangleright i \leftarrow i + 1$;
- 7: $A[i] \leftrightarrow A[j]$;
- 8: **fim**
- 9: **fim**
- 10: $A[i + 1] \leftrightarrow A[d]$;
- 11: **Retorne** $i + 1$;

Particione: funcionamento

		i			j				p
33	11	33	77	99	55	88	66	22	44
e									d

- 1: PARTICIONE(A, e, d)
- 2: $p \leftarrow A[d]$; { p é o “pivô”}
- 3: $i \leftarrow e - 1$;
- 4: **Para** $j \leftarrow e$ até $d - 1$ **faça**
- 5: **Se** $A[j] \leq p$ **então**
- 6: $i \leftarrow i + 1$;
- 7: $\triangleright A[i] \leftrightarrow A[j]$;
- 8: **fim**
- 9: **fim**
- 10: $A[i + 1] \leftrightarrow A[d]$;
- 11: **Retorne** $i + 1$;

Particione: funcionamento

		i				j			p
33	11	33	77	99	55	88	66	22	44
e									d

- 1: PARTICIONE(A, e, d)
- 2: $p \leftarrow A[d]$; { p é o “pivô”}
- 3: $i \leftarrow e - 1$;
- 4: ▷ Para $j \leftarrow e$ até $d - 1$ faça
- 5: Se $A[j] \leq p$ então
- 6: $i \leftarrow i + 1$;
- 7: $A[i] \leftrightarrow A[j]$;
- 8: fim
- 9: fim
- 10: $A[i + 1] \leftrightarrow A[d]$;
- 11: Retorne $i + 1$;

Particione: funcionamento

		i				j			p
33	11	33	77	99	55	88	66	22	44
e									d

- 1: PARTICIONE(A, e, d)
- 2: $p \leftarrow A[d];$ $\{p \text{ é o "pivô"}\}$
- 3: $i \leftarrow e - 1;$
- 4: **Para** $j \leftarrow e$ **até** $d - 1$ **faça**
- 5: \triangleright **Se** $A[j] \leq p$ **então**
- 6: $i \leftarrow i + 1;$
- 7: $A[i] \leftrightarrow A[j];$
- 8: **fim**
- 9: **fim**
- 10: $A[i + 1] \leftrightarrow A[d];$
- 11: **Retorne** $i + 1;$

Particione: funcionamento

		i					j		p
33	11	33	77	99	55	88	66	22	44
e									d

- 1: PARTICIONE(A, e, d)
- 2: $p \leftarrow A[d]$; { p é o “pivô”}
- 3: $i \leftarrow e - 1$;
- 4: ▷ Para $j \leftarrow e$ até $d - 1$ faça
- 5: Se $A[j] \leq p$ então
- 6: $i \leftarrow i + 1$;
- 7: $A[i] \leftrightarrow A[j]$;
- 8: fim
- 9: fim
- 10: $A[i + 1] \leftrightarrow A[d]$;
- 11: Retorne $i + 1$;

Particione: funcionamento

			<i>i</i>					<i>j</i>	<i>p</i>
33	11	33	77	99	55	88	66	22	44
								<i>e</i>	<i>d</i>

- 1: PARTICIONE(A, e, d)
- 2: $p \leftarrow A[d]$; { p é o “pivô”}
- 3: $i \leftarrow e - 1$;
- 4: **Para** $j \leftarrow e$ até $d - 1$ **faça**
- 5: \triangleright **Se** $A[j] \leq p$ **então**
- 6: $i \leftarrow i + 1$;
- 7: $A[i] \leftrightarrow A[j]$;
- 8: **fim**
- 9: **fim**
- 10: $A[i + 1] \leftrightarrow A[d]$;
- 11: **Retorne** $i + 1$;

Particione: funcionamento

i						j			p
33	11	33	77	99	55	88	66	22	44
e								d	

- 1: PARTICIONE(A, e, d)
- 2: $p \leftarrow A[d]$; { p é o “pivô”}
- 3: $i \leftarrow e - 1$;
- 4: ▷ Para $j \leftarrow e$ até $d - 1$ faça
- 5: Se $A[j] \leq p$ então
- 6: $i \leftarrow i + 1$;
- 7: $A[i] \leftrightarrow A[j]$;
- 8: fim
- 9: fim
- 10: $A[i + 1] \leftrightarrow A[d]$;
- 11: Retorne $i + 1$;

Particione: funcionamento

			i						j	p		
33	11	33	77	99	55	88	66	22	44			
e										d		

- 1: PARTICIONE(*A*, *e*, *d*)
- 2: *p* ← *A*[*d*]; {*p* é o “pivô”}
- 3: *i* ← *e* − 1;
- 4: **Para** *j* ← *e* até *d* − 1 **faça**
- 5: ▷ **Se** *A*[*j*] ≤ *p* **então**
- 6: *i* ← *i* + 1;
- 7: *A*[*i*] ↔ *A*[*j*];
- 8: **fim**
- 9: **fim**
- 10: *A*[*i* + 1] ↔ *A*[*d*];
- 11: **Retorne** *i* + 1;

Particione: funcionamento

			i						j p
33	11	33	77	99	55	88	66	22	44
e								d	

- 1: PARTICIONE(A, e, d)
- 2: $p \leftarrow A[d]$; { p é o “pivô”}
- 3: $i \leftarrow e - 1$;
- 4: **Para** $j \leftarrow e$ até $d - 1$ **faça**
- 5: **Se** $A[j] \leq p$ **então**
- 6: $\triangleright i \leftarrow i + 1$;
- 7: $A[i] \leftrightarrow A[j]$;
- 8: **fim**
- 9: **fim**
- 10: $A[i + 1] \leftrightarrow A[d]$;
- 11: **Retorne** $i + 1$;

Particione: funcionamento

			i					j	p
33	11	33	22	99	55	88	66	77	44
e									d

- 1: PARTICIONE(A, e, d)
- 2: $p \leftarrow A[d]$; { p é o “pivô”}
- 3: $i \leftarrow e - 1$;
- 4: **Para** $j \leftarrow e$ até $d - 1$ **faça**
- 5: **Se** $A[j] \leq p$ **então**
- 6: $i \leftarrow i + 1$;
- 7: $\triangleright A[i] \leftrightarrow A[j]$;
- 8: **fim**
- 9: **fim**
- 10: $A[i + 1] \leftrightarrow A[d]$;
- 11: **Retorne** $i + 1$;

Particione: funcionamento

			i					j	p
33	11	33	22	99	55	88	66	77	44
e									d

- 1: PARTICIONE(A, e, d)
- 2: $p \leftarrow A[d]$; { p é o “pivô”}
- 3: $i \leftarrow e - 1$;
- 4: ▷ Para $j \leftarrow e$ até $d - 1$ faça
- 5: Se $A[j] \leq p$ então
- 6: $i \leftarrow i + 1$;
- 7: $A[i] \leftrightarrow A[j]$;
- 8: fim
- 9: fim
- 10: $A[i + 1] \leftrightarrow A[d]$;
- 11: Retorne $i + 1$;

Particione: funcionamento

			i					j	p
33	11	33	22	99	55	88	66	77	44
e									d

- 1: PARTICIONE(A, e, d)
- 2: $p \leftarrow A[d]$; { p é o “pivô”}
- 3: $i \leftarrow e - 1$;
- 4: **Para** $j \leftarrow e$ até $d - 1$ **faça**
- 5: **Se** $A[j] \leq p$ **então**
- 6: $i \leftarrow i + 1$;
- 7: $A[i] \leftrightarrow A[j]$;
- 8: **fim**
- 9: **fim**
- 10: $\triangleright A[i + 1] \leftrightarrow A[d]$;
- 11: **Retorne** $i + 1$;

Particione: funcionamento

										$i+1$	j	p
33	11	33	22	99	55	88	66	77	44			
										e		d

- 1: PARTICIONE(A, e, d)
- 2: $p \leftarrow A[d]$; { p é o “pivô”}
- 3: $i \leftarrow e - 1$;
- 4: **Para** $j \leftarrow e$ até $d - 1$ **faça**
- 5: **Se** $A[j] \leq p$ **então**
- 6: $i \leftarrow i + 1$;
- 7: $A[i] \leftrightarrow A[j]$;
- 8: **fim**
- 9: **fim**
- 10: $\triangleright A[i + 1] \leftrightarrow A[d]$;
- 11: **Retorne** $i + 1$;

Particione: funcionamento

				$i+1$				j	
33	11	33	22	44	55	88	66	77	99
e				p					d

- 1: PARTICIONE(A, e, d)
- 2: $p \leftarrow A[d];$ $\{p \text{ é o "pivô"}\}$
- 3: $i \leftarrow e - 1;$
- 4: **Para** $j \leftarrow e$ até $d - 1$ **faça**
- 5: **Se** $A[j] \leq p$ **então**
- 6: $i \leftarrow i + 1;$
- 7: $A[i] \leftrightarrow A[j];$
- 8: **fim**
- 9: **fim**
- 10: $\triangleright A[i + 1] \leftrightarrow A[d];$
- 11: **Retorne** $i + 1;$

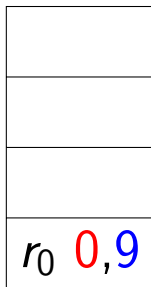
Sumário

- 1 Introdução
- 2 Particionamento
- 3 Quick-Sort**
- 4 Conquista (esquerda)
- 5 Quick-Sort
- 6 Conquista (direita)
- 7 Complexidade

Quick-Sort

r_0) Quick-Sort (A, e, d)

1. se $e < d$ então
2. $p = \text{Particione}(A, e, d);$
3. Quick-Sort ($A, e, p - 1$);
4. Quick-Sort ($A, p + 1, d$);



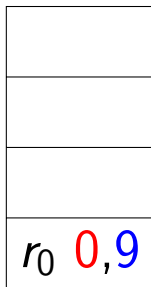
pilha
(recursão)

A =										
	e								d	
	99	33	55	77	11	33	88	66	22	44
	0	1	2	3	4	5	6	7	8	9

Quick-Sort

r_0) Quick-Sort (A, e, d)

- ▷
1. se $e < d$ então
 2. $p = \text{Particione}(A, e, d);$
 3. Quick-Sort ($A, e, p - 1$);
 4. Quick-Sort ($A, p + 1, d$);



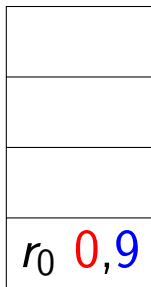
pilha
(recursão)

A =										
	e								d	
	99	33	55	77	11	33	88	66	22	44
	0	1	2	3	4	5	6	7	8	9

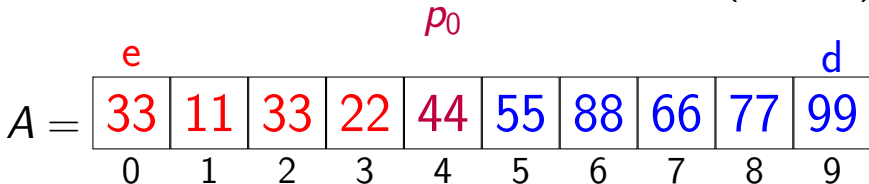
Quick-Sort

r_0) Quick-Sort (A, e, d)

- ▷
1. se $e < d$ então
 2. $p = \text{Particione}(A, e, d);$
 3. Quick-Sort ($A, e, p - 1$);
 4. Quick-Sort ($A, p + 1, d$);



pilha
(recursão)



Quick-Sort

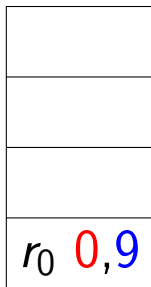
r_0) Quick-Sort ($A, \mathbf{e}, \mathbf{d}$)

1. se $\mathbf{e} < \mathbf{d}$ então

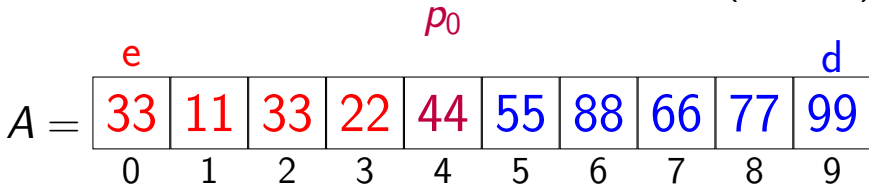
2. $\mathbf{p} = \text{Particione}(A, \mathbf{e}, \mathbf{d});$

r_1) 3. Quick-Sort ($A, \mathbf{e}, \mathbf{p} - 1$);

4. Quick-Sort ($A, \mathbf{p} + 1, \mathbf{d}$);



pilha
(recursão)



Quick-Sort

r_0) Quick-Sort (A, e, d)

1. se $e < d$ então

2. $p = \text{Particione}(A, e, d);$

r_1) 3. Quick-Sort ($A, e, p - 1$);

4. Quick-Sort ($A, p + 1, d$);

r_1 0, 3
r_0 0, 9

pilha
(recursão)

	e		d	p_0						
$A =$	33	11	33	22	44	55	88	66	77	99
	0	1	2	3	4	5	6	7	8	9

Sumário

- 1 Introdução
- 2 Particionamento
- 3 Quick-Sort
- 4 Conquista (esquerda)**
- 5 Quick-Sort
- 6 Conquista (direita)
- 7 Complexidade

Particione: conquista a esquerda

33	11	33	22	44	55	88	66	77	99
----	----	----	----	----	----	----	----	----	----

e

d

- 1: PARTICIONE(A, e, d)
- 2: $p \leftarrow A[d]$; { p é o “pivô”}
- 3: $i \leftarrow e - 1$;
- 4: **Para** $j \leftarrow e$ até $d - 1$ **faça**
- 5: **Se** $A[j] \leq p$ **então**
- 6: $i \leftarrow i + 1$;
- 7: $A[i] \leftrightarrow A[j]$;
- 8: **fim**
- 9: **fim**
- 10: $A[i + 1] \leftrightarrow A[d]$;
- 11: **Retorne** $i + 1$;

Particione: conquista a esquerda

p

33	11	33	22	44	55	88	66	77	99
----	----	----	-----------	----	----	----	----	----	----

e **d**

- 1: PARTICIONE(A, e, d)
- 2: $\triangleright p \leftarrow A[d]; \quad \{p \text{ é o "pivô"}\}$
- 3: $i \leftarrow e - 1;$
- 4: **Para** $j \leftarrow e$ até $d - 1$ **faça**
- 5: **Se** $A[j] \leq p$ **então**
- 6: $i \leftarrow i + 1;$
- 7: $A[i] \leftrightarrow A[j];$
- 8: **fim**
- 9: **fim**
- 10: $A[i + 1] \leftrightarrow A[d];$
- 11: **Retorne** $i + 1;$

Particione: conquista a esquerda

i

p

33	11	33	22	44	55	88	66	77	99
----	----	----	----	----	----	----	----	----	----

e

d

- 1: PARTICIONE(A, e, d)
- 2: $p \leftarrow A[d]$; $\{p \text{ é o "pivô"}\}$
- 3: $\triangleright i \leftarrow e - 1$;
- 4: **Para** $j \leftarrow e$ **até** $d - 1$ **faça**
- 5: **Se** $A[j] \leq p$ **então**
- 6: $i \leftarrow i + 1$;
- 7: $A[i] \leftrightarrow A[j]$;
- 8: **fim**
- 9: **fim**
- 10: $A[i + 1] \leftrightarrow A[d]$;
- 11: **Retorne** $i + 1$;

Particione: conquista a esquerda

<i>i</i>	<i>j</i>		<i>p</i>						
33	11	33	22	44	55	88	66	77	99

e *d*

- 1: PARTICIONE(*A*, *e*, *d*)
- 2: *p* $\leftarrow A[d]$; {*p* é o “pivô”}
- 3: *i* $\leftarrow e - 1$;
- 4: ▷ Para *j* $\leftarrow e$ até *d* - 1 faça
- 5: Se $A[j] \leq p$ então
- 6: *i* $\leftarrow i + 1$;
- 7: $A[i] \leftrightarrow A[j]$;
- 8: fim
- 9: fim
- 10: $A[i + 1] \leftrightarrow A[d]$;
- 11: Retorne *i* + 1;

Particione: conquista a esquerda

i	j		p						
33	11	33	22	44	55	88	66	77	99

e d

- 1: PARTICIONE(A, e, d)
- 2: $p \leftarrow A[d]$; $\{p \text{ é o "pivô"}\}$
- 3: $i \leftarrow e - 1$;
- 4: **Para** $j \leftarrow e$ até $d - 1$ **faça**
- 5: \triangleright **Se** $A[j] \leq p$ **então**
- 6: $i \leftarrow i + 1$;
- 7: $A[i] \leftrightarrow A[j]$;
- 8: **fim**
- 9: **fim**
- 10: $A[i + 1] \leftrightarrow A[d]$;
- 11: **Retorne** $i + 1$;

Particione: conquista a esquerda

i	j	p							
33	11	33	22	44	55	88	66	77	99

e d

- 1: PARTICIONE(A, e, d)
- 2: $p \leftarrow A[d]$; { p é o “pivô”}
- 3: $i \leftarrow e - 1$;
- 4: ▷ Para $j \leftarrow e$ até $d - 1$ faça
- 5: Se $A[j] \leq p$ então
- 6: $i \leftarrow i + 1$;
- 7: $A[i] \leftrightarrow A[j]$;
- 8: fim
- 9: fim
- 10: $A[i + 1] \leftrightarrow A[d]$;
- 11: Retorne $i + 1$;

Particione: conquista a esquerda

i	j	p							
33	11	33	22	44	55	88	66	77	99

e d

- 1: PARTICIONE(A, e, d)
- 2: $p \leftarrow A[d]$; { p é o “pivô”}
- 3: $i \leftarrow e - 1$;
- 4: **Para** $j \leftarrow e$ até $d - 1$ **faça**
- 5: \triangleright **Se** $A[j] \leq p$ **então**
- 6: $i \leftarrow i + 1$;
- 7: $A[i] \leftrightarrow A[j]$;
- 8: **fim**
- 9: **fim**
- 10: $A[i + 1] \leftrightarrow A[d]$;
- 11: **Retorne** $i + 1$;

Particione: conquista a esquerda

<i>i</i>	<i>j</i>		<i>p</i>						
33	11	33	22	44	55	88	66	77	99

e *d*

- 1: PARTICIONE(*A*, *e*, *d*)
- 2: *p* ← *A*[*d*]; {*p* é o “pivô”}
- 3: *i* ← *e* − 1;
- 4: **Para** *j* ← *e* até *d* − 1 **faça**
- 5: **Se** *A*[*j*] ≤ *p* **então**
- 6: ▷ *i* ← *i* + 1;
- 7: *A*[*i*] ↔ *A*[*j*];
- 8: **fim**
- 9: **fim**
- 10: *A*[*i* + 1] ↔ *A*[*d*];
- 11: **Retorne** *i* + 1;

Particione: conquista a esquerda

<i>i</i>	<i>j</i>		<i>p</i>						
11	33	33	22	44	55	88	66	77	99
<i>e</i>				<i>d</i>					

- 1: PARTICIONE(A, e, d)
- 2: $p \leftarrow A[d];$ $\{p \text{ é o "pivô"}\}$
- 3: $i \leftarrow e - 1;$
- 4: **Para** $j \leftarrow e$ até $d - 1$ **faça**
- 5: **Se** $A[j] \leq p$ **então**
- 6: $i \leftarrow i + 1;$
- 7: $\triangleright A[i] \leftrightarrow A[j];$
- 8: **fim**
- 9: **fim**
- 10: $A[i + 1] \leftrightarrow A[d];$
- 11: **Retorne** $i + 1;$

Particione: conquista a esquerda

<i>i</i>		<i>j</i>	<i>p</i>						
11	33	33	22	44	55	88	66	77	99
<i>e</i>				<i>d</i>					

- 1: PARTICIONE(A, e, d)
- 2: $p \leftarrow A[d]$; { p é o “pivô”}
- 3: $i \leftarrow e - 1$;
- 4: ▷ Para $j \leftarrow e$ até $d - 1$ faça
- 5: Se $A[j] \leq p$ então
- 6: $i \leftarrow i + 1$;
- 7: $A[i] \leftrightarrow A[j]$;
- 8: fim
- 9: fim
- 10: $A[i + 1] \leftrightarrow A[d]$;
- 11: Retorne $i + 1$;

Particione: conquista a esquerda

<i>i</i>		<i>j</i>	<i>p</i>						
11	33	33	22	44	55	88	66	77	99

e *d*

- 1: PARTICIONE(*A*, *e*, *d*)
- 2: *p* $\leftarrow A[d]$; {*p* é o “pivô”}
- 3: *i* $\leftarrow e - 1$;
- 4: **Para** *j* $\leftarrow e$ até *d* - 1 **faça**
- 5: ▷ **Se** $A[j] \leq p$ **então**
- 6: *i* $\leftarrow i + 1$;
- 7: $A[i] \leftrightarrow A[j]$;
- 8: **fim**
- 9: **fim**
- 10: $A[i + 1] \leftrightarrow A[d]$;
- 11: **Retorne** *i* + 1;

Particione: conquista a esquerda

i		j		p					
11	33	33	22	44	55	88	66	77	99
e		d							

- 1: PARTICIONE(A, e, d)
- 2: $p \leftarrow A[d];$ $\{p \text{ é o "pivô"}\}$
- 3: $i \leftarrow e - 1;$
- 4: ▷ Para $j \leftarrow e$ até $d - 1$ faça
- 5: Se $A[j] \leq p$ então
- 6: $i \leftarrow i + 1;$
- 7: $A[i] \leftrightarrow A[j];$
- 8: fim
- 9: fim
- 10: $A[i + 1] \leftrightarrow A[d];$
- 11: Retorne $i + 1;$

Particione: conquista a esquerda

	$i+1$	j	p						
11	33	33	22	44	55	88	66	77	99
e				d					

- 1: PARTICIONE(A, e, d)
- 2: $p \leftarrow A[d]$; { p é o “pivô”}
- 3: $i \leftarrow e - 1$;
- 4: **Para** $j \leftarrow e$ até $d - 1$ **faça**
- 5: **Se** $A[j] \leq p$ **então**
- 6: $i \leftarrow i + 1$;
- 7: $A[i] \leftrightarrow A[j]$;
- 8: **fim**
- 9: **fim**
- 10: $\triangleright A[i + 1] \leftrightarrow A[d]$;
- 11: **Retorne** $i + 1$;

Particione: conquista a esquerda

	$i+1$	j							
11	22	33	33	44	55	88	66	77	99

e p d

- 1: PARTICIONE(A, e, d)
- 2: $p \leftarrow A[d];$ $\{p \text{ é o "pivô"}\}$
- 3: $i \leftarrow e - 1;$
- 4: **Para** $j \leftarrow e$ **até** $d - 1$ **faça**
- 5: **Se** $A[j] \leq p$ **então**
- 6: $i \leftarrow i + 1;$
- 7: $A[i] \leftrightarrow A[j];$
- 8: **fim**
- 9: **fim**
- 10: $\triangleright A[i + 1] \leftrightarrow A[d];$
- 11: **Retorne** $i + 1;$

Particione: conquista a esquerda

	$i+1$	j							
11	22	33	33	44	55	88	66	77	99

e p d

- 1: PARTICIONE(A, e, d)
- 2: $p \leftarrow A[d];$ $\{p \text{ é o "pivô"}\}$
- 3: $i \leftarrow e - 1;$
- 4: **Para** $j \leftarrow e$ até $d - 1$ **faça**
- 5: **Se** $A[j] \leq p$ **então**
- 6: $i \leftarrow i + 1;$
- 7: $A[i] \leftrightarrow A[j];$
- 8: **fim**
- 9: **fim**
- 10: $A[i + 1] \leftrightarrow A[d];$
- 11: **Retorne** $i + 1;$

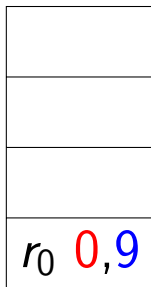
Sumário

- 1 Introdução
- 2 Particionamento
- 3 Quick-Sort
- 4 Conquista (esquerda)
- 5 Quick-Sort**
- 6 Conquista (direira)
- 7 Complexidade

Quick-Sort

r_0) Quick-Sort (A, e, d)

1. se $e < d$ então
2. $p = \text{Particione}(A, e, d);$
3. Quick-Sort ($A, e, p - 1$);
4. Quick-Sort ($A, p + 1, d$);



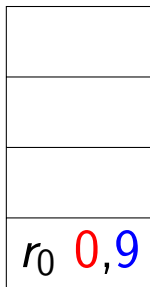
pilha
(recursão)

A =										
	e								d	
	99	33	55	77	11	33	88	66	22	44
	0	1	2	3	4	5	6	7	8	9

Quick-Sort

r_0) Quick-Sort (A, e, d)

- ▷
1. se $e < d$ então
 2. $p = \text{Particione}(A, e, d);$
 3. Quick-Sort ($A, e, p - 1$);
 4. Quick-Sort ($A, p + 1, d$);



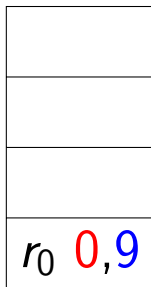
pilha
(recursão)

A =										
	e									d
	99	33	55	77	11	33	88	66	22	44
	0	1	2	3	4	5	6	7	8	9

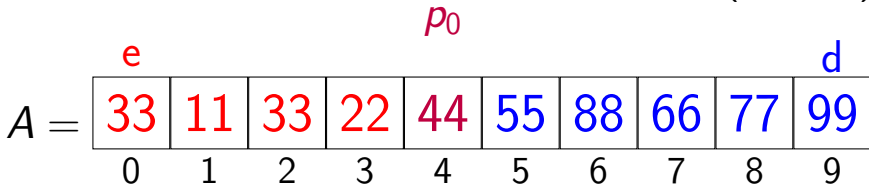
Quick-Sort

r_0) Quick-Sort (A, e, d)

- ▷
1. se $e < d$ então
 2. $p = \text{Particione}(A, e, d);$
 3. Quick-Sort ($A, e, p - 1$);
 4. Quick-Sort ($A, p + 1, d$);



pilha
(recursão)



Quick-Sort

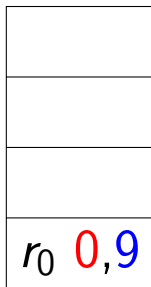
r_0) Quick-Sort ($A, \mathbf{e}, \mathbf{d}$)

1. se $\mathbf{e} < \mathbf{d}$ então

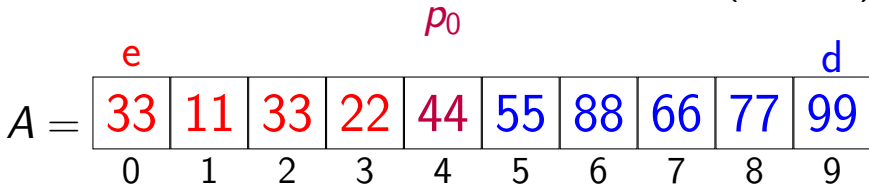
2. $\mathbf{p} = \text{Particione}(A, \mathbf{e}, \mathbf{d});$

r_1) 3. Quick-Sort ($A, \mathbf{e}, \mathbf{p} - 1$);

4. Quick-Sort ($A, \mathbf{p} + 1, \mathbf{d}$);



pilha
(recursão)



Quick-Sort

r_0) Quick-Sort (A, e, d)

1. se $e < d$ então

2. $p = \text{Particione}(A, e, d);$

r_1) 3. Quick-Sort ($A, e, p - 1$);

4. Quick-Sort ($A, p + 1, d$);

r_1 0, 3
r_0 0, 9

pilha
(recursão)

				p_0						
	e		d							
$A =$	33	11	33	22	44	55	88	66	77	99
	0	1	2	3	4	5	6	7	8	9

Quick-Sort

r_0) Quick-Sort (A, e, d)

- ▷ 1. se $e < d$ então
2. $p = \text{Particione}(A, e, d);$
- r_1) 3. Quick-Sort ($A, e, p - 1$);
4. Quick-Sort ($A, p + 1, d$);

r_1 0, 3
r_0 0, 9

pilha
(recursão)

	e		d	p_0						
$A =$	33	11	33	22	44	55	88	66	77	99
	0	1	2	3	4	5	6	7	8	9

Quick-Sort

r_0) Quick-Sort (A, e, d)

1. se $e < d$ então

▷ 2. $p = \text{Particione}(A, e, d);$

r_1) 3. Quick-Sort ($A, e, p - 1$);

4. Quick-Sort ($A, p + 1, d$);

r_1 0,3
r_0 0,9

pilha
(recursão)

	e	p_1		d	p_0					
$A =$	11	22	33	33	44	55	88	66	77	99
	0	1	2	3	4	5	6	7	8	9

Quick-Sort

$r_0)$ Quick-Sort (A, e, d)

1. se $e < d$ então

2. $p = \text{Particione}(A, e, d);$

$r_1)r_2)$ 3. Quick-Sort ($A, e, p - 1$);

4. Quick-Sort ($A, p + 1, d$);

r_1 0,3
r_0 0,9

pilha
(recursão)

	e	p_1		d	p_0					
$A =$	11	22	33	33	44	55	88	66	77	99
	0	1	2	3	4	5	6	7	8	9

Quick-Sort

$r_0)$ Quick-Sort ($A, \mathbf{e}, \mathbf{d}$)

1. se $\mathbf{e} < \mathbf{d}$ então

2. $\mathbf{p} = \text{Particione}(A, \mathbf{e}, \mathbf{d});$

$r_1)r_2)$ 3. Quick-Sort ($A, \mathbf{e}, \mathbf{p} - 1$);

4. Quick-Sort ($A, \mathbf{p} + 1, \mathbf{d}$);

r_2 0,0
r_1 0,3
r_0 0,9

pilha
(recursão)

\mathbf{d}
 \mathbf{e}

p_1

p_0

$A =$	11	22	33	33	44	55	88	66	77	99
	0	1	2	3	4	5	6	7	8	9

Quick-Sort

$r_0)$ Quick-Sort ($A, \mathbf{e}, \mathbf{d}$)
 \triangleright 1. se $\mathbf{e} < \mathbf{d}$ então
 $r_1)r_2)$ 2. $\mathbf{p} = \text{Particione}(A, \mathbf{e}, \mathbf{d});$
 3. Quick-Sort ($A, \mathbf{e}, \mathbf{p} - 1$);
 4. Quick-Sort ($A, \mathbf{p} + 1, \mathbf{d}$);

r_2	$\mathbf{0}, \mathbf{0}$
r_1	$\mathbf{0}, \mathbf{3}$
r_0	$\mathbf{0}, \mathbf{9}$

pilha
 (recursão)

	\mathbf{d}	p_1		p_0						
	\mathbf{e}									
$A =$	11	22	33	33	44	55	88	66	77	99
	0	1	2	3	4	5	6	7	8	9

Quick-Sort

r_0) Quick-Sort ($A, \mathbf{e}, \mathbf{d}$)

1. se $\mathbf{e} < \mathbf{d}$ então

2. $\mathbf{p} = \text{Particione}(A, \mathbf{e}, \mathbf{d});$

r_1) 3. Quick-Sort ($A, \mathbf{e}, \mathbf{p} - 1$);

4. Quick-Sort ($A, \mathbf{p} + 1, \mathbf{d}$);

r_1 0, 3
r_0 0, 9

pilha
(recursão)

p_1

p_0

$A =$	11	22	33	33	44	55	88	66	77	99
	0	1	2	3	4	5	6	7	8	9

Quick-Sort

r_0) Quick-Sort (A, e, d)

1. se $e < d$ então

2. $p = \text{Particione}(A, e, d);$

r_1) 3. Quick-Sort ($A, e, p - 1$);

4. Quick-Sort ($A, p + 1, d$);

r_1 0,3
r_0 0,9

pilha
(recursão)

	e	p_1		d	p_0					
$A =$	11	22	33	33	44	55	88	66	77	99
	0	1	2	3	4	5	6	7	8	9

Quick-Sort

$r_0)$ Quick-Sort (A, e, d)

1. se $e < d$ então

2. $p = \text{Particione}(A, e, d);$

$r_1)$ 3. Quick-Sort ($A, e, p - 1$);

$r_3)$ 4. Quick-Sort ($A, p + 1, d$);

r_3 2,3
r_1 0,3
r_0 0,9

pilha
(recursão)

	p_1			p_0						
		e	d							
$A =$	11	22	33	33	44	55	88	66	77	99
	0	1	2	3	4	5	6	7	8	9

Quick-Sort

$r_0)$ Quick-Sort ($A, \mathbf{e}, \mathbf{d}$)

\triangleright 1. se $\mathbf{e} < \mathbf{d}$ então

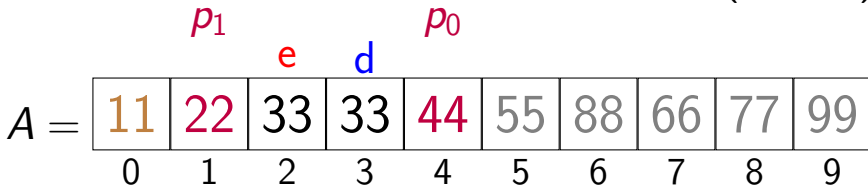
2. $\mathbf{p} = \text{Particione}(A, \mathbf{e}, \mathbf{d});$

$r_1)$ 3. Quick-Sort ($A, \mathbf{e}, \mathbf{p} - 1$);

$r_3)$ 4. Quick-Sort ($A, \mathbf{p} + 1, \mathbf{d}$);

r_3 2,3
r_1 0,3
r_0 0,9

pilha
(recursão)



Quick-Sort

r_0) Quick-Sort (A, e, d)

1. se $e < d$ então

▷ 2. $p = \text{Particione}(A, e, d);$

r_1) 3. Quick-Sort ($A, e, p - 1$);

r_3) 4. Quick-Sort ($A, p + 1, d$);

r_3 2,3
r_1 0,3
r_0 0,9

pilha
(recursão)

		p_1		p_3	p_0					
			e	d						
$A =$	11	22	33	33	44	55	88	66	77	99
	0	1	2	3	4	5	6	7	8	9

Quick-Sort

$r_0)$ Quick-Sort (A, e, d)

1. se $e < d$ então

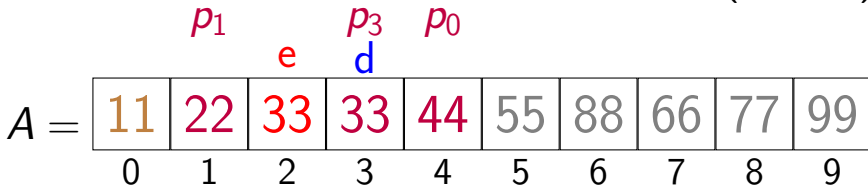
2. $p = \text{Particione}(A, e, d);$

$r_1)r_4)$ 3. Quick-Sort ($A, e, p - 1$);

$r_3)$ 4. Quick-Sort ($A, p + 1, d$);

r_3	2,3
r_1	0,3
r_0	0,9

pilha
(recursão)



Quick-Sort

$r_0)$ Quick-Sort ($A, \mathbf{e}, \mathbf{d}$)

 1. se $\mathbf{e} < \mathbf{d}$ então
 2. $\mathbf{p} = \text{Particione}(A, \mathbf{e}, \mathbf{d});$
 $r_1)r_4)$ 3. Quick-Sort ($A, \mathbf{e}, \mathbf{p} - 1$);
 $r_3)$ 4. Quick-Sort ($A, \mathbf{p} + 1, \mathbf{d}$);

r_4	2,2
r_3	2,3
r_1	0,3
r_0	0,9

pilha
 (recursão)

		p_1	\mathbf{d}	p_3	p_0					
			\mathbf{e}							
$A =$	11	22	33	33	44	55	88	66	77	99
	0	1	2	3	4	5	6	7	8	9

Quick-Sort

$r_0)$	Quick-Sort ($A, \mathbf{e}, \mathbf{d}$)
\triangleright	1. se $\mathbf{e} < \mathbf{d}$ então
	2. $\mathbf{p} = \text{Particione}(A, \mathbf{e}, \mathbf{d});$
$r_1)r_4)$	3. Quick-Sort ($A, \mathbf{e}, \mathbf{p} - 1$);
$r_3)$	4. Quick-Sort ($A, \mathbf{p} + 1, \mathbf{d}$);

r_4	2,2
r_3	2,3
r_1	0,3
r_0	0,9

pilha
 (recursão)

		p_1	\mathbf{d}	p_3	p_0					
			\mathbf{e}							
$A =$	11	22	33	33	44	55	88	66	77	99
	0	1	2	3	4	5	6	7	8	9

Quick-Sort

$r_0)$ Quick-Sort ($A, \mathbf{e}, \mathbf{d}$)

1. se $\mathbf{e} < \mathbf{d}$ então

2. $\mathbf{p} = \text{Particione}(A, \mathbf{e}, \mathbf{d});$

$r_1)$ 3. Quick-Sort ($A, \mathbf{e}, \mathbf{p} - 1$);

$r_3)$ 4. Quick-Sort ($A, \mathbf{p} + 1, \mathbf{d}$);

r_3 2,3
r_1 0,3
r_0 0,9

pilha
(recursão)

p_1 p_3 p_0

$A =$	11	22	33	33	44	55	88	66	77	99
	0	1	2	3	4	5	6	7	8	9

Quick-Sort

$r_0)$ Quick-Sort (A, e, d)

1. se $e < d$ então

2. $p = \text{Particione}(A, e, d);$

$r_1)$ 3. Quick-Sort ($A, e, p - 1$);

$r_3)$ 4. Quick-Sort ($A, p + 1, d$);

r_3 2,3
r_1 0,3
r_0 0,9

pilha
(recursão)

		p_1		p_3	p_0					
			e	d						
$A =$	11	22	33	33	44	55	88	66	77	99
	0	1	2	3	4	5	6	7	8	9

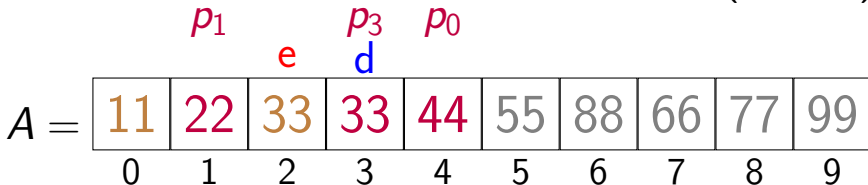
Quick-Sort

$r_0)$ Quick-Sort ($A, \mathbf{e}, \mathbf{d}$)

 1. se $\mathbf{e} < \mathbf{d}$ então
 2. $\mathbf{p} = \text{Particione}(A, \mathbf{e}, \mathbf{d});$
 $r_1)$ 3. Quick-Sort ($A, \mathbf{e}, \mathbf{p} - 1$);
 $r_3)r_5)$ 4. Quick-Sort ($A, \mathbf{p} + 1, \mathbf{d}$);

r_3	$\mathbf{2}, \mathbf{3}$
r_1	$\mathbf{0}, \mathbf{3}$
r_0	$\mathbf{0}, \mathbf{9}$

pilha
 (recursão)



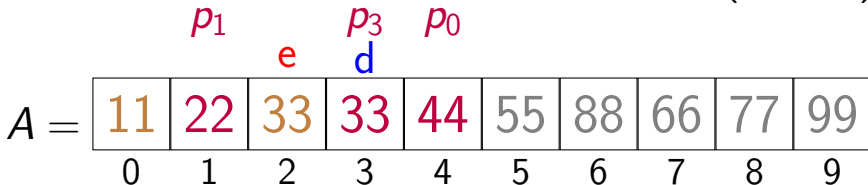
Quick-Sort

$r_0)$ Quick-Sort (A, e, d)

 1. se $e < d$ então
 2. $p = \text{Particione}(A, e, d);$
 $r_1)$ 3. Quick-Sort ($A, e, p - 1$);
 $r_3)r_5)$ 4. Quick-Sort ($A, p + 1, d$);

r_3 2,3
r_1 0,3
r_0 0,9

pilha
 (recursão)



Quick-Sort

$r_0)$ Quick-Sort ($A, \mathbf{e}, \mathbf{d}$)

 1. se $\mathbf{e} < \mathbf{d}$ então
 2. $\mathbf{p} = \text{Particione}(A, \mathbf{e}, \mathbf{d});$
 $r_1)$ 3. Quick-Sort ($A, \mathbf{e}, \mathbf{p} - 1$);
 $r_3)r_5)$ 4. Quick-Sort ($A, \mathbf{p} + 1, \mathbf{d}$);

r_5	4,3
r_3	2,3
r_1	0,3
r_0	0,9

pilha
(recursão)

	p_1		p_3	p_0						
			d	e						
$A =$	11	22	33	33	44	55	88	66	77	99
	0	1	2	3	4	5	6	7	8	9

Quick-Sort

$r_0)$	Quick-Sort ($A, \mathbf{e}, \mathbf{d}$)
\triangleright	1. se $\mathbf{e} < \mathbf{d}$ então
	2. $\mathbf{p} = \text{Particione}(A, \mathbf{e}, \mathbf{d});$
$r_1)$	3. Quick-Sort ($A, \mathbf{e}, \mathbf{p} - 1$);
$r_3)r_5)$	4. Quick-Sort ($A, \mathbf{p} + 1, \mathbf{d}$);

r_5	4,3
r_3	2,3
r_1	0,3
r_0	0,9

pilha
(recursão)

	p_1		p_3	p_0						
			d	e						
$A =$	11	22	33	33	44	55	88	66	77	99
	0	1	2	3	4	5	6	7	8	9

Quick-Sort

$r_0)$ Quick-Sort ($A, \mathbf{e}, \mathbf{d}$)

1. se $\mathbf{e} < \mathbf{d}$ então

2. $\mathbf{p} = \text{Particione}(A, \mathbf{e}, \mathbf{d});$

$r_1)$ 3. Quick-Sort ($A, \mathbf{e}, \mathbf{p} - 1$);

$r_3)$ 4. Quick-Sort ($A, \mathbf{p} + 1, \mathbf{d}$);

r_3 2,3
r_1 0,3
r_0 0,9

pilha
(recursão)

p_1 p_3 p_0

$A =$	11	22	33	33	44	55	88	66	77	99
	0	1	2	3	4	5	6	7	8	9

Quick-Sort

r_0) Quick-Sort (A, e, d)

1. se $e < d$ então

2. $p = \text{Particione}(A, e, d);$

r_1) 3. Quick-Sort ($A, e, p - 1$);

4. Quick-Sort ($A, p + 1, d$);

r_1 0,3
r_0 0,9

pilha
(recursão)

p_1

p_3

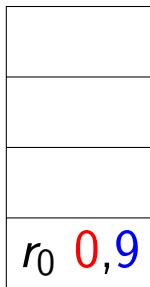
p_0

$A =$	11	22	33	33	44	55	88	66	77	99
	0	1	2	3	4	5	6	7	8	9

Quick-Sort

r_0) Quick-Sort (A, e, d)

1. se $e < d$ então
2. $p = \text{Particione}(A, e, d);$
3. Quick-Sort ($A, e, p - 1$);
4. Quick-Sort ($A, p + 1, d$);



pilha
(recursão)

p_1 p_3 p_0

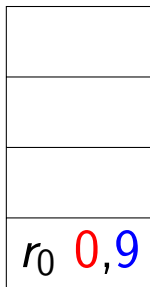
$A =$	11	22	33	33	44	55	88	66	77	99
	0	1	2	3	4	5	6	7	8	9

Quick-Sort

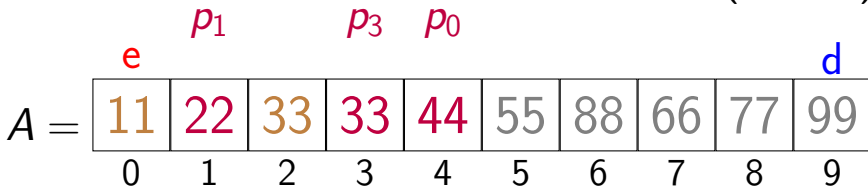
r_0) Quick-Sort (A, e, d)

1. se $e < d$ então
2. $p = \text{Particione}(A, e, d);$
3. Quick-Sort ($A, e, p - 1$);
4. Quick-Sort ($A, p + 1, d$);

r_6)



pilha
(recursão)



Quick-Sort

$r_0)$ Quick-Sort (A, e, d)

1. se $e < d$ então
2. $p = \text{Particione}(A, e, d);$
3. Quick-Sort ($A, e, p - 1$);
4. Quick-Sort ($A, p + 1, d$);

$r_6)$

r_6 5,9
r_0 0,9

pilha
(recursão)

p_1

p_3

p_0

e

d

$A =$

11	22	33	33	44	55	88	66	77	99
0	1	2	3	4	5	6	7	8	9

Sumário

- 1 Introdução
- 2 Particionamento
- 3 Quick-Sort
- 4 Conquista (esquerda)
- 5 Quick-Sort
- 6 Conquista (direita)**
- 7 Complexidade

Particione: conquista a direita

11	22	33	33	44	55	88	66	77	99
					e				d

- 1: PARTICIONE(A, e, d)
- 2: $p \leftarrow A[d]$; { p é o “pivô”}
- 3: $i \leftarrow e - 1$;
- 4: **Para** $j \leftarrow e$ até $d - 1$ **faça**
- 5: **Se** $A[j] \leq p$ **então**
- 6: $i \leftarrow i + 1$;
- 7: $A[i] \leftrightarrow A[j]$;
- 8: **fim**
- 9: **fim**
- 10: $A[i + 1] \leftrightarrow A[d]$;
- 11: **Retorne** $i + 1$;

Particione: conquista a direita

11	22	33	33	44	55	88	66	77	99
					e				p
									d

- 1: PARTICIONE(A, e, d)
- 2: $\triangleright p \leftarrow A[d]; \quad \{p \text{ é o "pivô"}\}$
- 3: $i \leftarrow e - 1;$
- 4: **Para** $j \leftarrow e$ até $d - 1$ **faça**
- 5: **Se** $A[j] \leq p$ **então**
- 6: $i \leftarrow i + 1;$
- 7: $A[i] \leftrightarrow A[j];$
- 8: **fim**
- 9: **fim**
- 10: $A[i + 1] \leftrightarrow A[d];$
- 11: **Retorne** $i + 1;$

Particione: conquista a direita

										p
11	22	33	33	44	55	88	66	77	99	
										d

- 1: PARTICIONE(A, e, d)
- 2: $p \leftarrow A[d]$; { p é o “pivô”}
- 3: $\triangleright i \leftarrow e - 1$;
- 4: **Para** $j \leftarrow e$ até $d - 1$ **faça**
- 5: **Se** $A[j] \leq p$ **então**
- 6: $i \leftarrow i + 1$;
- 7: $A[i] \leftrightarrow A[j]$;
- 8: **fim**
- 9: **fim**
- 10: $A[i + 1] \leftrightarrow A[d]$;
- 11: **Retorne** $i + 1$;

Particione: conquista a direita

				i	j				p
11	22	33	33	44	55	88	66	77	99
					e				d

- 1: PARTICIONE(A, e, d)
- 2: $p \leftarrow A[d]$; { p é o “pivô”}
- 3: $i \leftarrow e - 1$;
- 4: ▷ Para $j \leftarrow e$ até $d - 1$ faça
- 5: Se $A[j] \leq p$ então
- 6: $i \leftarrow i + 1$;
- 7: $A[i] \leftrightarrow A[j]$;
- 8: fim
- 9: fim
- 10: $A[i + 1] \leftrightarrow A[d]$;
- 11: Retorne $i + 1$;

Particione: conquista a direita

				i	j				p
11	22	33	33	44	55	88	66	77	99
					e				d

- 1: PARTICIONE(A, e, d)
- 2: $p \leftarrow A[d]$; { p é o “pivô”}
- 3: $i \leftarrow e - 1$;
- 4: **Para** $j \leftarrow e$ até $d - 1$ **faça**
- 5: \triangleright **Se** $A[j] \leq p$ **então**
- 6: $i \leftarrow i + 1$;
- 7: $A[i] \leftrightarrow A[j]$;
- 8: **fim**
- 9: **fim**
- 10: $A[i + 1] \leftrightarrow A[d]$;
- 11: **Retorne** $i + 1$;

Particione: conquista a direita

					i,j				p
11	22	33	33	44	55	88	66	77	99
					e				d

- 1: PARTICIONE(A, e, d)
- 2: $p \leftarrow A[d]$; $\{p \text{ é o "pivô"}\}$
- 3: $i \leftarrow e - 1$;
- 4: **Para** $j \leftarrow e$ até $d - 1$ **faça**
- 5: **Se** $A[j] \leq p$ **então**
- 6: $i \leftarrow i + 1$;
- 7: $A[i] \leftrightarrow A[j]$;
- 8: **fim**
- 9: **fim**
- 10: $A[i + 1] \leftrightarrow A[d]$;
- 11: **Retorne** $i + 1$;

Particione: conquista a direita

					i, j				p
11	22	33	33	44	55	88	66	77	99
					e				d

- 1: PARTICIONE(A, e, d)
- 2: $p \leftarrow A[d]$; { p é o “pivô”}
- 3: $i \leftarrow e - 1$;
- 4: **Para** $j \leftarrow e$ até $d - 1$ **faça**
- 5: **Se** $A[j] \leq p$ **então**
- 6: $i \leftarrow i + 1$;
- 7: $\triangleright A[i] \leftrightarrow A[j]$;
- 8: **fim**
- 9: **fim**
- 10: $A[i + 1] \leftrightarrow A[d]$;
- 11: **Retorne** $i + 1$;

Particione: conquista a direita

					i	j			p
11	22	33	33	44	55	88	66	77	99
					e				d

- 1: PARTICIONE(A, e, d)
- 2: $p \leftarrow A[d]$; { p é o “pivô”}
- 3: $i \leftarrow e - 1$;
- 4: ▷ Para $j \leftarrow e$ até $d - 1$ faça
- 5: Se $A[j] \leq p$ então
- 6: $i \leftarrow i + 1$;
- 7: $A[i] \leftrightarrow A[j]$;
- 8: fim
- 9: fim
- 10: $A[i + 1] \leftrightarrow A[d]$;
- 11: Retorne $i + 1$;

Particione: conquista a direita

					i	j			p
11	22	33	33	44	55	88	66	77	99
					e				d

- 1: PARTICIONE(A, e, d)
- 2: $p \leftarrow A[d]$; { p é o “pivô”}
- 3: $i \leftarrow e - 1$;
- 4: **Para** $j \leftarrow e$ até $d - 1$ **faça**
- 5: \triangleright **Se** $A[j] \leq p$ **então**
- 6: $i \leftarrow i + 1$;
- 7: $A[i] \leftrightarrow A[j]$;
- 8: **fim**
- 9: **fim**
- 10: $A[i + 1] \leftrightarrow A[d]$;
- 11: **Retorne** $i + 1$;

Particione: conquista a direita

						i,j			p
11	22	33	33	44	55	88	66	77	99
					e				d

- 1: PARTICIONE(A, e, d)
- 2: $p \leftarrow A[d]$; $\{p \text{ é o "pivô"}\}$
- 3: $i \leftarrow e - 1$;
- 4: **Para** $j \leftarrow e$ até $d - 1$ **faça**
- 5: **Se** $A[j] \leq p$ **então**
- 6: $i \leftarrow i + 1$;
- 7: $A[i] \leftrightarrow A[j]$;
- 8: **fim**
- 9: **fim**
- 10: $A[i + 1] \leftrightarrow A[d]$;
- 11: **Retorne** $i + 1$;

Particione: conquista a direita

						i, j			p
11	22	33	33	44	55	88	66	77	99
					e				d

- 1: PARTICIONE(A, e, d)
- 2: $p \leftarrow A[d];$ $\{p \text{ é o "pivô"}\}$
- 3: $i \leftarrow e - 1;$
- 4: **Para** $j \leftarrow e$ até $d - 1$ **faça**
- 5: **Se** $A[j] \leq p$ **então**
- 6: $i \leftarrow i + 1;$
- 7: $\triangleright A[i] \leftrightarrow A[j];$
- 8: **fim**
- 9: **fim**
- 10: $A[i + 1] \leftrightarrow A[d];$
- 11: **Retorne** $i + 1;$

Particione: conquista a direita

						i	j		p
11	22	33	33	44	55	88	66	77	99
					e				d

- 1: PARTICIONE(A, e, d)
- 2: $p \leftarrow A[d]$; { p é o “pivô”}
- 3: $i \leftarrow e - 1$;
- 4: ▷ Para $j \leftarrow e$ até $d - 1$ faça
- 5: Se $A[j] \leq p$ então
- 6: $i \leftarrow i + 1$;
- 7: $A[i] \leftrightarrow A[j]$;
- 8: fim
- 9: fim
- 10: $A[i + 1] \leftrightarrow A[d]$;
- 11: Retorne $i + 1$;

Particione: conquista a direita

						i	j		p
11	22	33	33	44	55	88	66	77	99
					e				d

- 1: PARTICIONE(A, e, d)
- 2: $p \leftarrow A[d]$; { p é o “pivô”}
- 3: $i \leftarrow e - 1$;
- 4: **Para** $j \leftarrow e$ até $d - 1$ **faça**
- 5: \triangleright **Se** $A[j] \leq p$ **então**
- 6: $i \leftarrow i + 1$;
- 7: $A[i] \leftrightarrow A[j]$;
- 8: **fim**
- 9: **fim**
- 10: $A[i + 1] \leftrightarrow A[d]$;
- 11: **Retorne** $i + 1$;

Particione: conquista a direita

							i,j		p
11	22	33	33	44	55	88	66	77	99
					e				d

- 1: PARTICIONE(A, e, d)
- 2: $p \leftarrow A[d]$; $\{p \text{ é o "pivô"}\}$
- 3: $i \leftarrow e - 1$;
- 4: **Para** $j \leftarrow e$ até $d - 1$ **faça**
- 5: **Se** $A[j] \leq p$ **então**
- 6: $i \leftarrow i + 1$;
- 7: $A[i] \leftrightarrow A[j]$;
- 8: **fim**
- 9: **fim**
- 10: $A[i + 1] \leftrightarrow A[d]$;
- 11: **Retorne** $i + 1$;

Particione: conquista a direita

11	22	33	33	44	55	88	66	77	99
					e		i, j		p
									d

- 1: PARTICIONE(A, e, d)
- 2: $p \leftarrow A[d]$; $\{p \text{ é o "pivô"}\}$
- 3: $i \leftarrow e - 1$;
- 4: **Para** $j \leftarrow e$ até $d - 1$ **faça**
- 5: **Se** $A[j] \leq p$ **então**
- 6: $i \leftarrow i + 1$;
- 7: $\triangleright A[i] \leftrightarrow A[j]$;
- 8: **fim**
- 9: **fim**
- 10: $A[i + 1] \leftrightarrow A[d]$;
- 11: **Retorne** $i + 1$;

Particione: conquista a direita

							i	j	p
11	22	33	33	44	55	88	66	77	99
					e				d

- 1: PARTICIONE(A, e, d)
- 2: $p \leftarrow A[d]$; { p é o “pivô”}
- 3: $i \leftarrow e - 1$;
- 4: ▷ Para $j \leftarrow e$ até $d - 1$ faça
- 5: Se $A[j] \leq p$ então
- 6: $i \leftarrow i + 1$;
- 7: $A[i] \leftrightarrow A[j]$;
- 8: fim
- 9: fim
- 10: $A[i + 1] \leftrightarrow A[d]$;
- 11: Retorne $i + 1$;

Particione: conquista a direita

							i	j	p
11	22	33	33	44	55	88	66	77	99
					e				d

- 1: PARTICIONE(A, e, d)
- 2: $p \leftarrow A[d]$; { p é o “pivô”}
- 3: $i \leftarrow e - 1$;
- 4: **Para** $j \leftarrow e$ até $d - 1$ **faça**
- 5: \triangleright **Se** $A[j] \leq p$ **então**
- 6: $i \leftarrow i + 1$;
- 7: $A[i] \leftrightarrow A[j]$;
- 8: **fim**
- 9: **fim**
- 10: $A[i + 1] \leftrightarrow A[d]$;
- 11: **Retorne** $i + 1$;

Particione: conquista a direita

11	22	33	33	44	55	88	66	77	99
					e			i,j	p
									d

- 1: PARTICIONE(A, e, d)
- 2: $p \leftarrow A[d]$; { p é o “pivô”}
- 3: $i \leftarrow e - 1$;
- 4: **Para** $j \leftarrow e$ até $d - 1$ **faça**
- 5: **Se** $A[j] \leq p$ **então**
- 6: $\triangleright i \leftarrow i + 1$;
- 7: $A[i] \leftrightarrow A[j]$;
- 8: **fim**
- 9: **fim**
- 10: $A[i + 1] \leftrightarrow A[d]$;
- 11: **Retorne** $i + 1$;

Particione: conquista a direita

								i,j	p
11	22	33	33	44	55	88	66	77	99
					e				d

- 1: PARTICIONE(A, e, d)
- 2: $p \leftarrow A[d]$; { p é o “pivô”}
- 3: $i \leftarrow e - 1$;
- 4: **Para** $j \leftarrow e$ até $d - 1$ **faça**
- 5: **Se** $A[j] \leq p$ **então**
- 6: $i \leftarrow i + 1$;
- 7: $\triangleright A[i] \leftrightarrow A[j]$;
- 8: **fim**
- 9: **fim**
- 10: $A[i + 1] \leftrightarrow A[d]$;
- 11: **Retorne** $i + 1$;

Particione: conquista a direita

								i,j	p
11	22	33	33	44	55	88	66	77	99
					e				d

- 1: PARTICIONE(A, e, d)
- 2: $p \leftarrow A[d]$; { p é o “pivô”}
- 3: $i \leftarrow e - 1$;
- 4: ▷ Para $j \leftarrow e$ até $d - 1$ faça
- 5: Se $A[j] \leq p$ então
- 6: $i \leftarrow i + 1$;
- 7: $A[i] \leftrightarrow A[j]$;
- 8: fim
- 9: fim
- 10: $A[i + 1] \leftrightarrow A[d]$;
- 11: Retorne $i + 1$;

Particione: conquista a direita

								j	i+1
11	22	33	33	44	55	88	66	77	99
					e				d

- 1: PARTICIONE(A, e, d)
- 2: $p \leftarrow A[d];$ $\{p \text{ é o "pivô"}\}$
- 3: $i \leftarrow e - 1;$
- 4: **Para** $j \leftarrow e$ até $d - 1$ **faça**
- 5: **Se** $A[j] \leq p$ **então**
- 6: $i \leftarrow i + 1;$
- 7: $A[i] \leftrightarrow A[j];$
- 8: **fim**
- 9: **fim**
- 10: $\triangleright A[i + 1] \leftrightarrow A[d];$
- 11: **Retorne** $i + 1;$

Particione: conquista a direita

								j	$i+1$
11	22	33	33	44	55	88	66	77	99
					e				d

- 1: PARTICIONE(A, e, d)
- 2: $p \leftarrow A[d]$; { p é o “pivô”}
- 3: $i \leftarrow e - 1$;
- 4: **Para** $j \leftarrow e$ até $d - 1$ **faça**
- 5: **Se** $A[j] \leq p$ **então**
- 6: $i \leftarrow i + 1$;
- 7: $A[i] \leftrightarrow A[j]$;
- 8: **fim**
- 9: **fim**
- 10: $A[i + 1] \leftrightarrow A[d]$;
- 11: ▷ **Retorne** $i + 1$;

Sumário

- 1 Introdução
- 2 Particionamento
- 3 Quick-Sort
- 4 Conquista (esquerda)
- 5 Quick-Sort
- 6 Conquista (direira)
- 7 Complexidade**

Complexidade

O algoritmo Quick-Sort tem o seu **pior** caso quando as divisões produzem sub-divisões extremamente desbalanceadas (por exemplo a escolha de um pivô como menor ou maior elemento do vetor), sendo sua equação de recorrência neste caso dada por:

$$T(\mathbf{n}) = \begin{cases} \Theta(1) & \text{se } \mathbf{n} = 1, \\ T(n-1) + T(1) + \Theta(\mathbf{n}) & \text{se } \mathbf{n} > 1. \end{cases}$$

Complexidade de tempo: $\Theta(n^2)$.

Complexidade de espaço: $\Theta(n)$.