Estrutura de Dados I

Merge-Sort

Prof. Rodrigo Minetto
Universidade Tecnológica Federal do Paraná
Material compilado de: Cormen.

Sumário

Merge-Sort

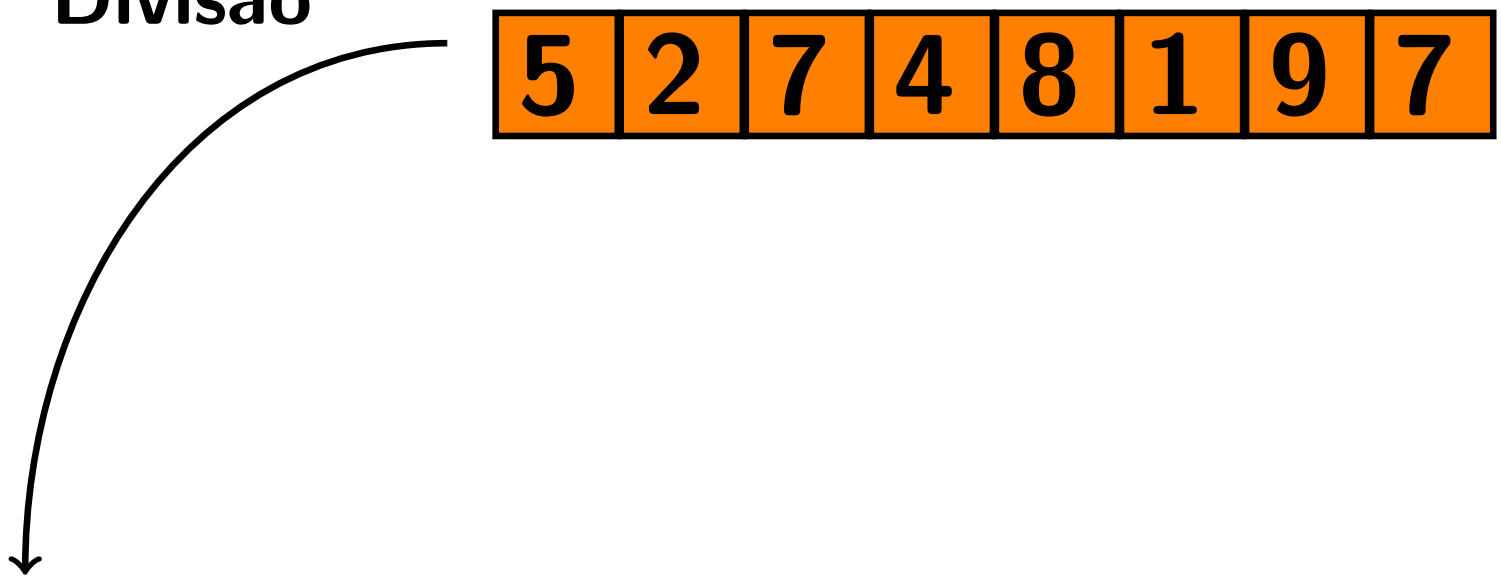
O algoritmo Merge-Sort foi inventado em 1945 por John Von Neumann. O algoritmo segue o paradigma de divisão e conquista (*divide and conquer*):

- Dividir: descubra o ponto médio do sub-arranjo (tempo constante).
- Conquistar: resolva recursivamente dois sub-problemas de tamanho n/2.
- Combinar: combine os dois sub-arranjos em um único conjunto ordenado (tempo de **n**).

Sumário

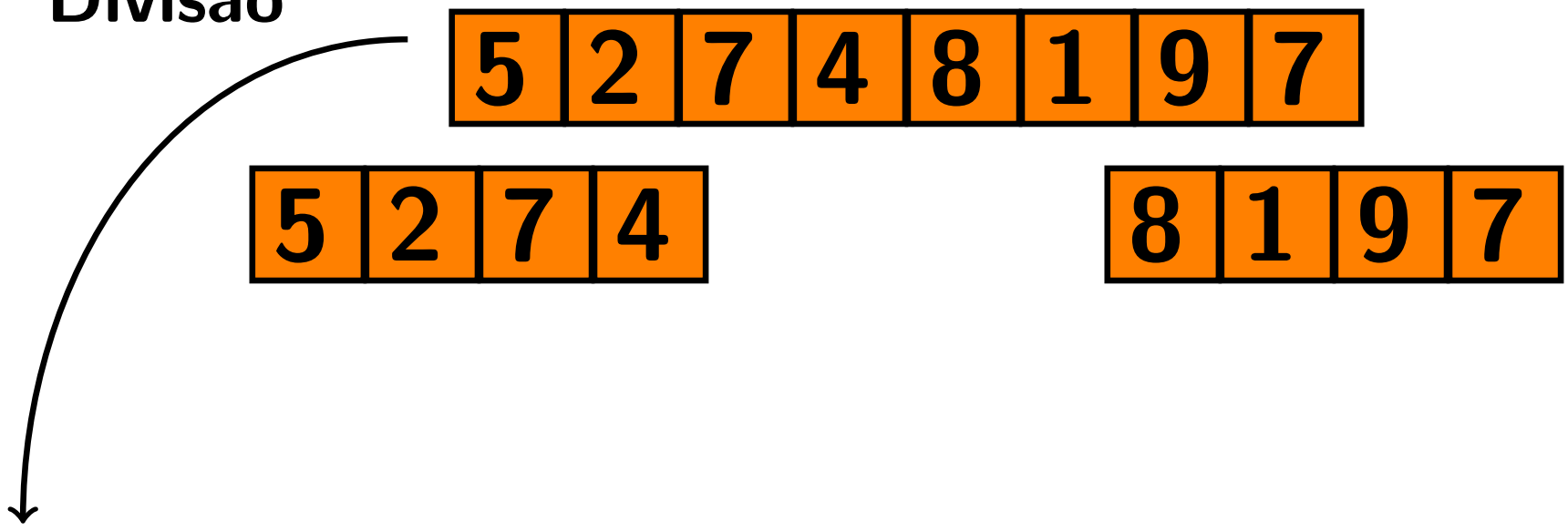
Execução

Divisão



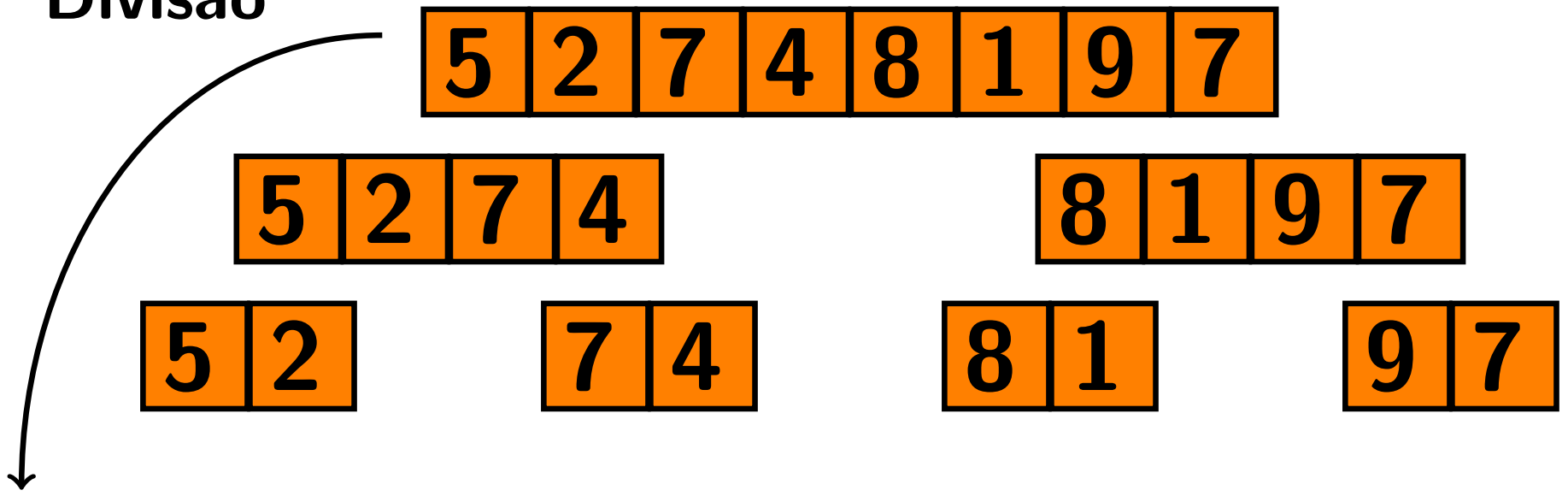
Execução

Divisão



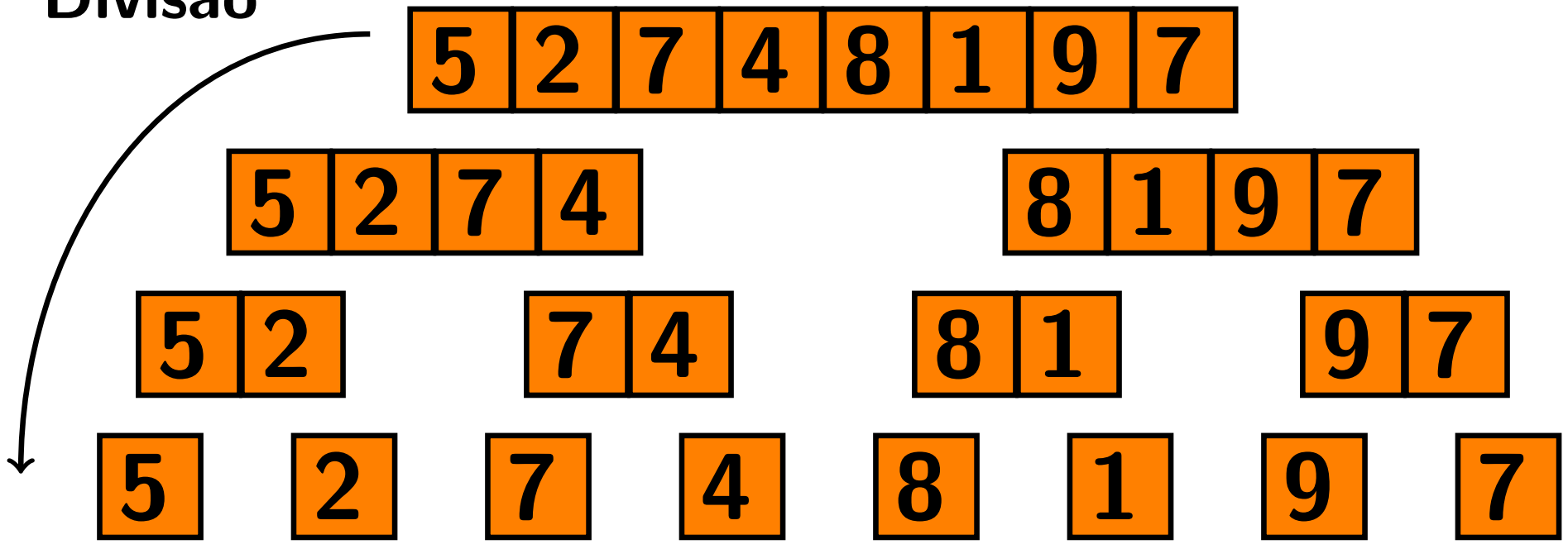
Execução

Divisão

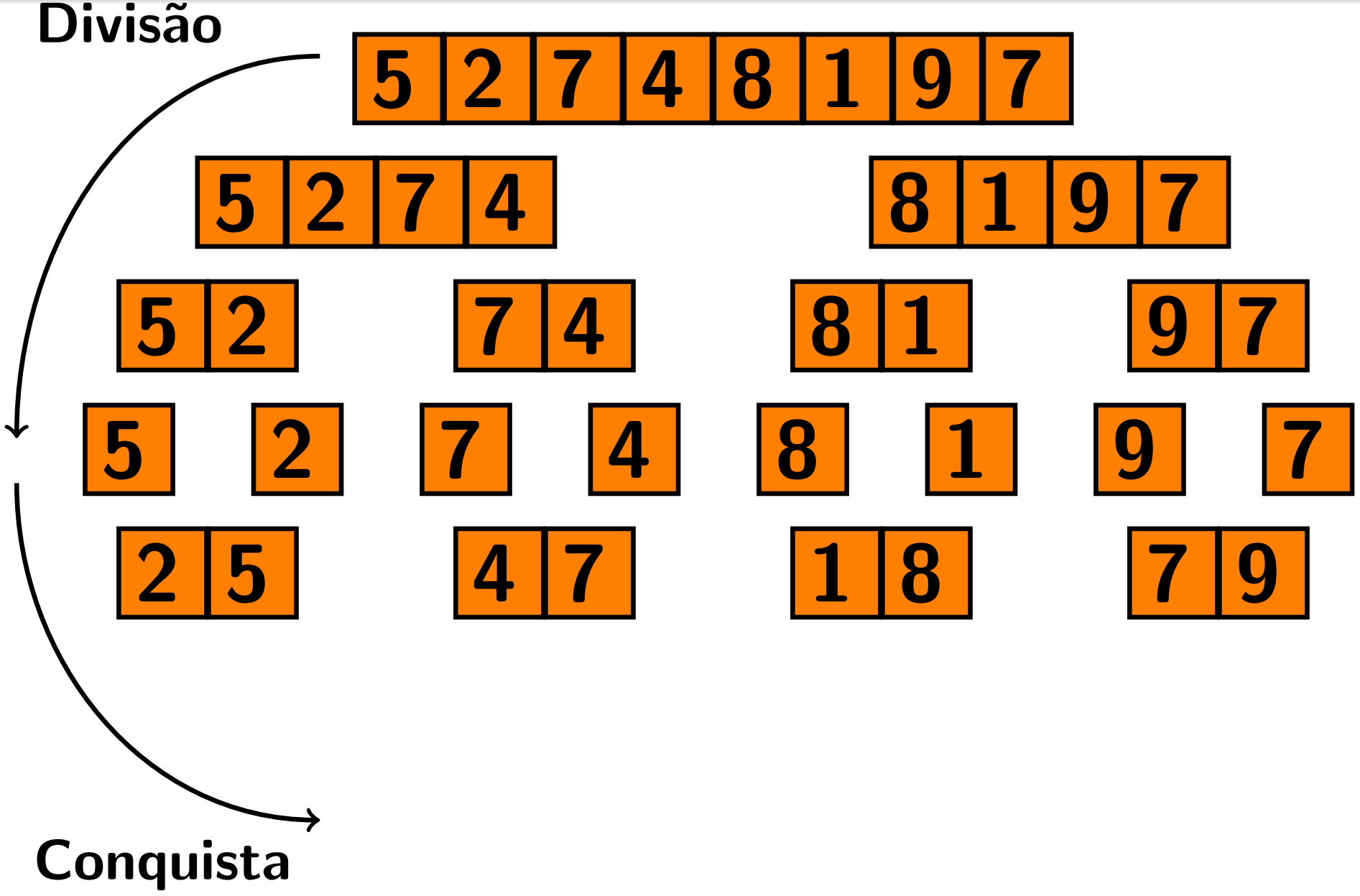


Execução

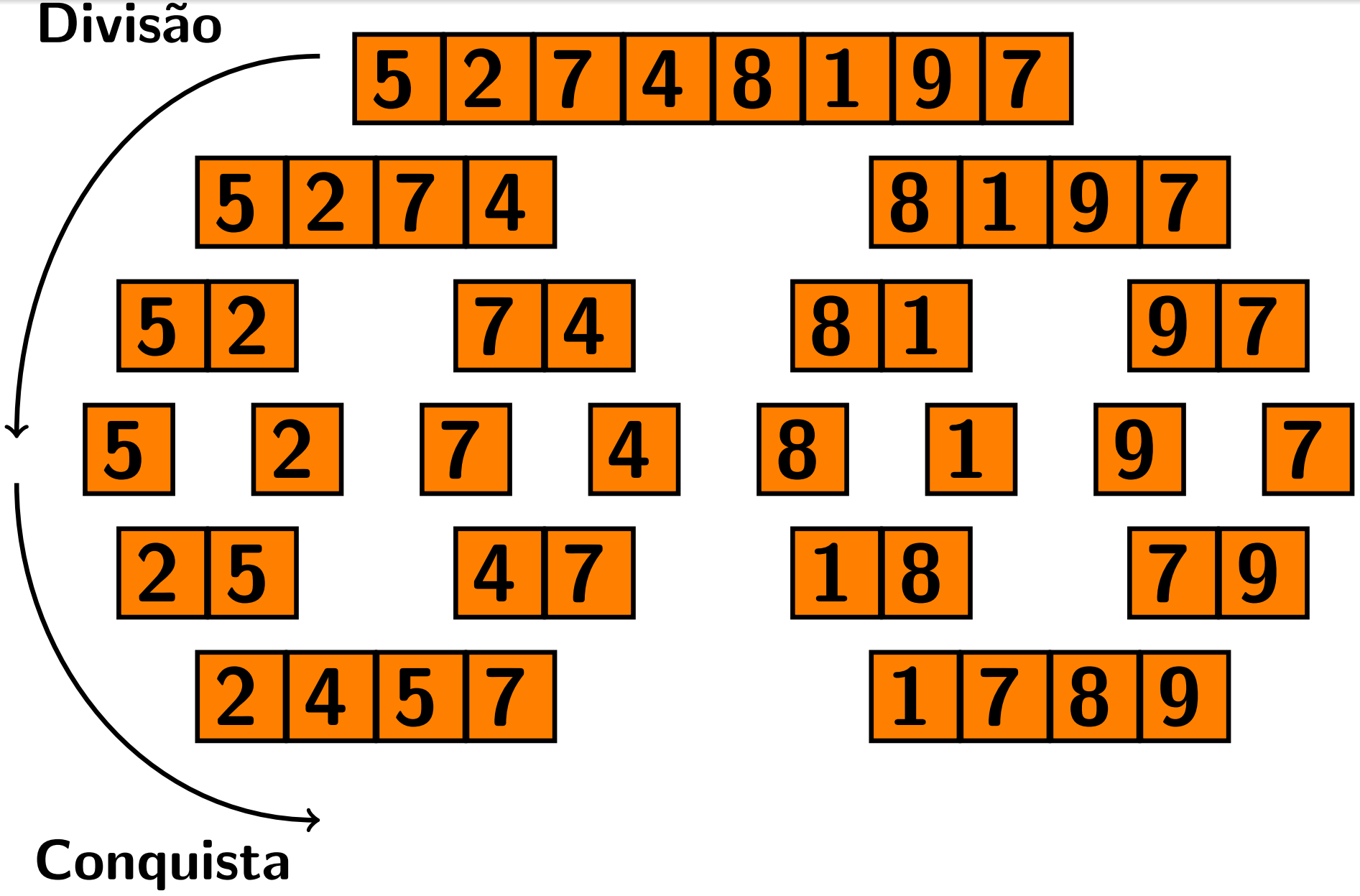
Divisão



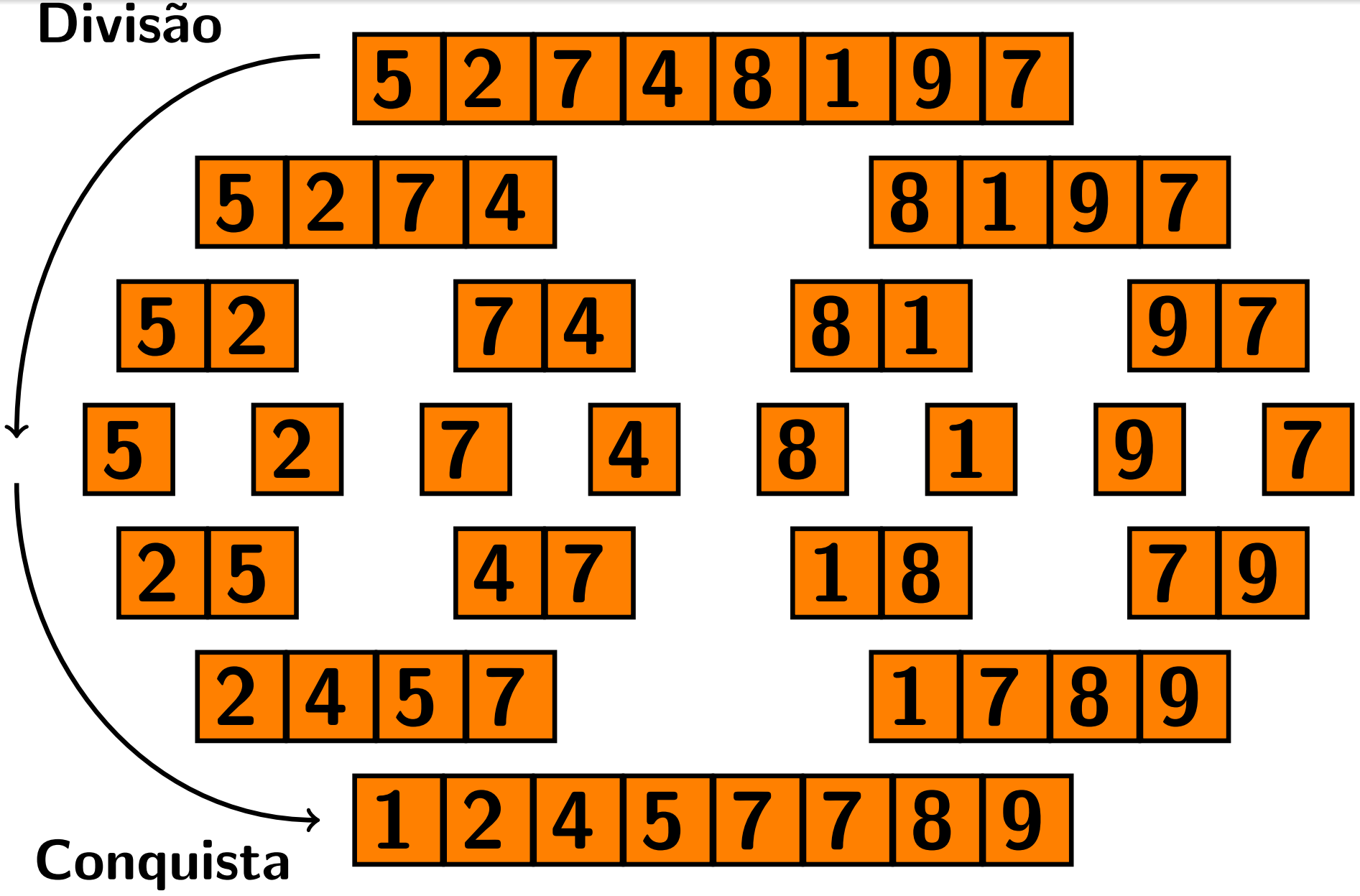
Execução


Divisão
5 2 7 4 8 1 9 7
5 2 7 4
8 1 9 7
5 2
7 4
8 1
9 7
5
2
7
4
8
1
9
7
2 5
4 7
1 8
7 9
Conquista

Execução


Divisão
5 2 7 4 8 1 9 7
5 2 7 4
8 1 9 7
5 2
7 4
8 1
9 7
5
2
7
4
8
1
9
7
2 5
4 7
1 8
7 9
2 4 5 7
1 7 8 9
Conquista

Execução

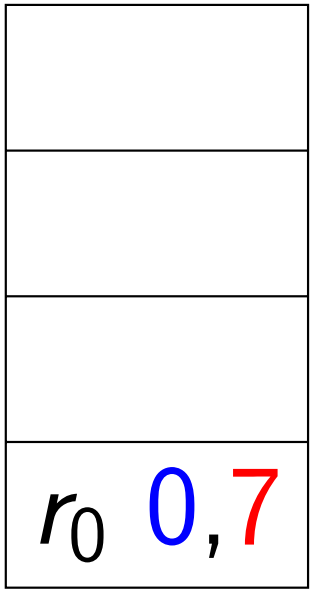

Divisão
5 2 7 4 8 1 9 7
5 2 7 4
8 1 9 7
5 2
7 4
8 1
9 7
5
2
7
4
8
1
9
7
2 5
4 7
1 8
7 9
2 4 5 7
1 7 8 9
1 2 4 5 7 7 8 9
Conquista

Sumário

Merge-Sort

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

1. se $\mathbf{e} < \mathbf{d}$ então
2. $\quad \mathbf{m} \leftarrow \lfloor (\mathbf{e} + \mathbf{d})/2 \rfloor$;
3. $\quad$ Merge-Sort $(A, \mathbf{e}, \mathbf{m}, O)$;
4. $\quad$ Merge-Sort $(A, \mathbf{m} + 1, \mathbf{d}, O)$;
5. $\quad$ Intercala $(A, \mathbf{e}, \mathbf{m}, \mathbf{d}, O)$;



pilha (recursão)

m_0

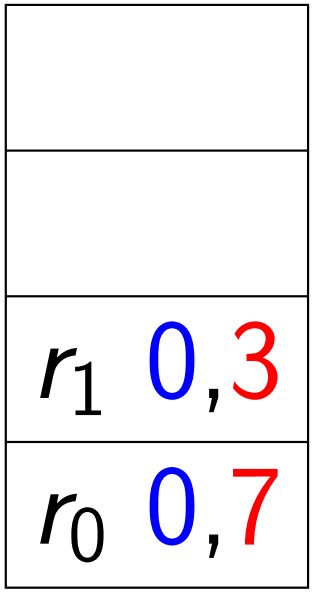
$A =$

e_0							d_0
5	2	7	4	8	1	9	7
0	1	2	3	4	5	6	7

Merge-Sort

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

1. se $\mathbf{e} < \mathbf{d}$ então
2. $\quad \mathbf{m} \leftarrow \lfloor (\mathbf{e} + \mathbf{d})/2 \rfloor$;
r_1) 3. $\quad$ Merge-Sort $(A, \mathbf{e}, \mathbf{m}, O)$;
4. $\quad$ Merge-Sort $(A, \mathbf{m} + 1, \mathbf{d}, O)$;
5. $\quad$ Intercala $(A, \mathbf{e}, \mathbf{m}, \mathbf{d}, O)$;



pilha (recursão)

m_1 m_0

e_1 d_1

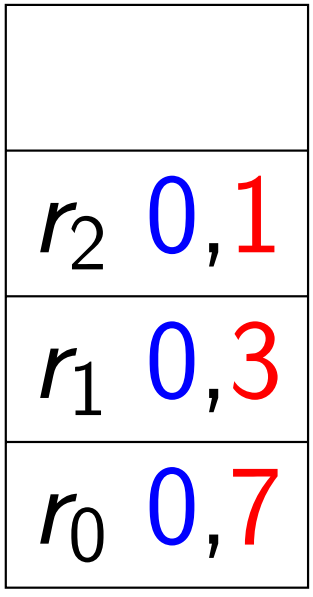
$A =$

5	2	7	4	8	1	9	7
0	1	2	3	4	5	6	7

Merge-Sort

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

1. se $\mathbf{e} < \mathbf{d}$ então
2. $\quad \mathbf{m} \leftarrow \lfloor (\mathbf{e} + \mathbf{d})/2 \rfloor$;
3. $\quad$ Merge-Sort $(A, \mathbf{e}, \mathbf{m}, O)$; r_1)r_2)
4. $\quad$ Merge-Sort $(A, \mathbf{m}+1, \mathbf{d}, O)$;
5. $\quad$ Intercala $(A, \mathbf{e}, \mathbf{m}, \mathbf{d}, O)$;



pilha (recursão)

m_2 m_1 m_0

e_2 d_2

$A =$

5	2	7	4	8	1	9	7
0	1	2	3	4	5	6	7

Merge-Sort

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

1. se $\mathbf{e} < \mathbf{d}$ então
2. $\quad \mathbf{m} \leftarrow \lfloor (\mathbf{e} + \mathbf{d})/2 \rfloor$;
3. $\quad$ Merge-Sort $(A, \mathbf{e}, \mathbf{m}, O)$; r_1)
4. $\quad$ Merge-Sort $(A, \mathbf{m}+1, \mathbf{d}, O)$; r_2)
5. $\quad$ Intercala $(A, \mathbf{e}, \mathbf{m}, \mathbf{d}, O)$; r_3)

r_3 0,0
r_2 0,1
r_1 0,3
r_0 0,7

pilha (recursão)

m_3
m_2 m_1 m_0
d_3
e_3

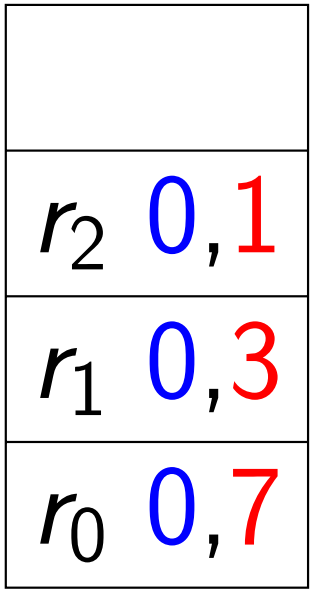
$A =$

5	2	7	4	8	1	9	7
0	1	2	3	4	5	6	7

Merge-Sort

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

1. se $\mathbf{e} < \mathbf{d}$ então
2. $\quad \mathbf{m} \leftarrow \lfloor (\mathbf{e} + \mathbf{d})/2 \rfloor$;
3. $\quad$ Merge-Sort $(A, \mathbf{e}, \mathbf{m}, O)$; r_1)r_2)
4. $\quad$ Merge-Sort $(A, \mathbf{m}+1, \mathbf{d}, O)$;
5. $\quad$ Intercala $(A, \mathbf{e}, \mathbf{m}, \mathbf{d}, O)$;



pilha (recursão)

m_2 m_1 m_0

e_2 d_2

$A =$

5	2	7	4	8	1	9	7
0	1	2	3	4	5	6	7

Merge-Sort

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

1. se $\mathbf{e} < \mathbf{d}$ então
2. $\quad \mathbf{m} \leftarrow \lfloor (\mathbf{e} + \mathbf{d})/2 \rfloor$;
3. $\quad$ Merge-Sort $(A, \mathbf{e}, \mathbf{m}, O)$; r_1)r_2)
4. $\quad$ Merge-Sort $(A, \mathbf{m}+1, \mathbf{d}, O)$; r_4)
5. $\quad$ Intercala $(A, \mathbf{e}, \mathbf{m}, \mathbf{d}, O)$;

r_4 1,1
r_2 0,1
r_1 0,3
r_0 0,7

pilha (recursão)

m_4
m_2 m_1 m_0
e_4
d_4

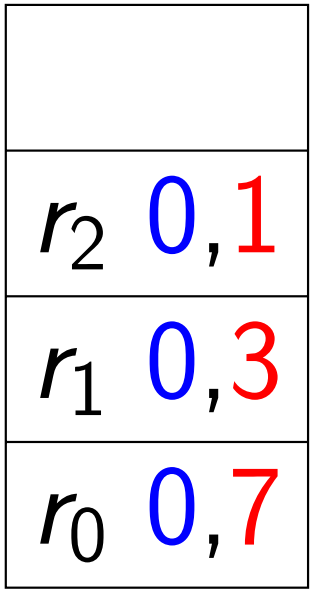
$A =$

5	2	7	4	8	1	9	7
0	1	2	3	4	5	6	7

Merge-Sort

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

1. se $\mathbf{e} < \mathbf{d}$ então
2. $\quad \mathbf{m} \leftarrow \lfloor (\mathbf{e} + \mathbf{d})/2 \rfloor$;
3. $\quad$ Merge-Sort $(A, \mathbf{e}, \mathbf{m}, O)$; r_1)r_2)
4. $\quad$ Merge-Sort $(A, \mathbf{m}+1, \mathbf{d}, O)$;
5. $\quad$ Intercala $(A, \mathbf{e}, \mathbf{m}, \mathbf{d}, O)$;



pilha (recursão)

m_2 m_1 m_0

e_2 d_2

$A =$

5	2	7	4	8	1	9	7
0	1	2	3	4	5	6	7

Merge-Sort

r_0)	Merge-Sort $(A, \mathbf{e}, \mathbf{d}, O)$	
	1. se $\mathbf{e} < \mathbf{d}$ então	
	2. $\quad \mathbf{m} \leftarrow \lfloor(\mathbf{e} + \mathbf{d})/2\rfloor$;	
r_1)r_2)	3. $\quad$ Merge-Sort $(A, \mathbf{e}, \mathbf{m}, O)$;	
	4. $\quad$ Merge-Sort $(A, \mathbf{m} + 1, \mathbf{d}, O)$;	
$\triangleright$	5. $\quad$ Intercala $(A, \mathbf{e}, \mathbf{m}, \mathbf{d}, O)$;	

r_2 0,1
r_1 0,3
r_0 0,7

pilha (recursão)

m_2 m_1 m_0

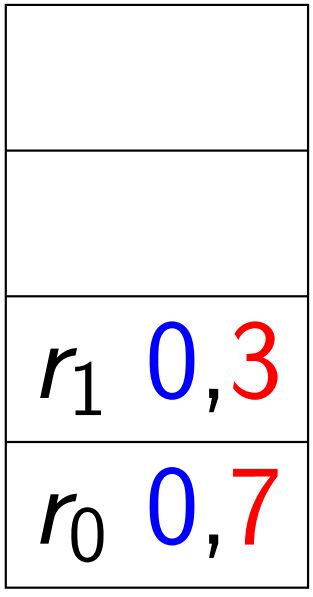
$A =$

e_2		d_2					
2	5	7	4	8	1	9	7
0	1	2	3	4	5	6	7

Merge-Sort

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

1. se $\mathbf{e} < \mathbf{d}$ então
2. $\quad \mathbf{m} \leftarrow \lfloor (\mathbf{e} + \mathbf{d})/2 \rfloor$;
r_1) 3. $\quad$ Merge-Sort $(A, \mathbf{e}, \mathbf{m}, O)$;
4. $\quad$ Merge-Sort $(A, \mathbf{m} + 1, \mathbf{d}, O)$;
5. $\quad$ Intercala $(A, \mathbf{e}, \mathbf{m}, \mathbf{d}, O)$;



pilha (recursão)

m_1 m_0

e_1 d_1

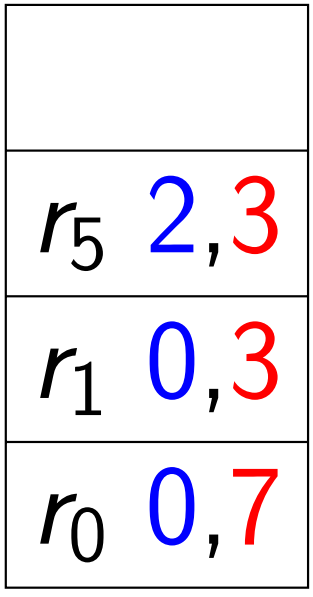
$A =$

2	5	7	4	8	1	9	7
0	1	2	3	4	5	6	7

Merge-Sort

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

1. se $\mathbf{e} < \mathbf{d}$ então
2. $\quad \mathbf{m} \leftarrow \lfloor (\mathbf{e} + \mathbf{d})/2 \rfloor$;
3. $\quad$ Merge-Sort $(A, \mathbf{e}, \mathbf{m}, O)$; ($r_1$)
4. $\quad$ Merge-Sort $(A, \mathbf{m}+1, \mathbf{d}, O)$; ($r_5$)
5. $\quad$ Intercala $(A, \mathbf{e}, \mathbf{m}, \mathbf{d}, O)$;



m_1 m_5 m_0

e_5 d_5

$A =$

2	5	7	4	8	1	9	7
0	1	2	3	4	5	6	7

Merge-Sort

r_0)	Merge-Sort $(A, \mathbf{e}, \mathbf{d}, O)$
	1. se $\mathbf{e} < \mathbf{d}$ então
	2. $\quad \mathbf{m} \leftarrow \lfloor(\mathbf{e} + \mathbf{d})/2\rfloor$;
r_1)r_6)	3. $\quad$ Merge-Sort $(A, \mathbf{e}, \mathbf{m}, O)$;
r_5)	4. $\quad$ Merge-Sort $(A, \mathbf{m} + 1, \mathbf{d}, O)$;
	5. $\quad$ Intercala $(A, \mathbf{e}, \mathbf{m}, \mathbf{d}, O)$;

r_6 2,2
r_5 2,3
r_1 0,3
r_0 0,7

pilha (recursão)

m_6
m_1 m_5 m_0
d_6
e_6

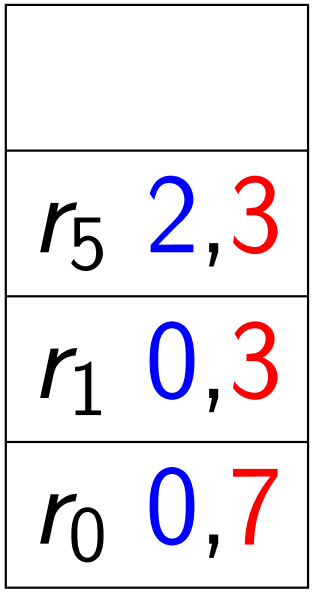
$A =$

2	5	7	4	8	1	9	7
0	1	2	3	4	5	6	7

Merge-Sort

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

1. se $\mathbf{e} < \mathbf{d}$ então
2. $\quad \mathbf{m} \leftarrow \lfloor (\mathbf{e} + \mathbf{d})/2 \rfloor$;
3. $\quad$ Merge-Sort $(A, \mathbf{e}, \mathbf{m}, O)$; ($r_1$)
4. $\quad$ Merge-Sort $(A, \mathbf{m}+1, \mathbf{d}, O)$; ($r_5$)
5. $\quad$ Intercala $(A, \mathbf{e}, \mathbf{m}, \mathbf{d}, O)$;



m_1 m_5 m_0

e_5 d_5

$A =$

2	5	7	4	8	1	9	7
0	1	2	3	4	5	6	7

Merge-Sort

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

1. se $\mathbf{e} < \mathbf{d}$ então
2. $\quad \mathbf{m} \leftarrow \lfloor (\mathbf{e} + \mathbf{d})/2 \rfloor$;
3. $\quad$ Merge-Sort $(A, \mathbf{e}, \mathbf{m}, O)$; r_1)
4. $\quad$ Merge-Sort $(A, \mathbf{m}+1, \mathbf{d}, O)$; r_5)r_7)
5. $\quad$ Intercala $(A, \mathbf{e}, \mathbf{m}, \mathbf{d}, O)$;

r_7	3,3
r_5	2,3
r_1	0,3
r_0	0,7

pilha (recursão)

m_7
m_1 m_5 m_0
e_7
d_7

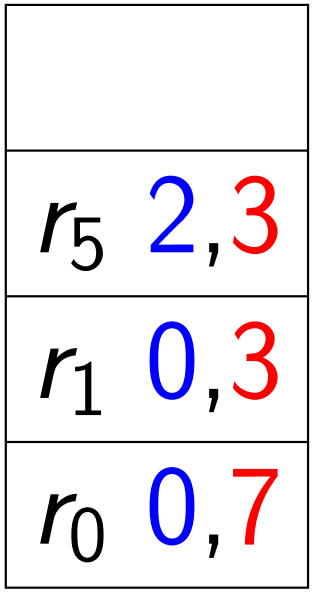
$A =$

0	1	2	3	4	5	6	7
2	5	7	4	8	1	9	7

Merge-Sort

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

1. se $\mathbf{e} < \mathbf{d}$ então
2. $\quad \mathbf{m} \leftarrow \lfloor (\mathbf{e} + \mathbf{d})/2 \rfloor$;
3. $\quad$ Merge-Sort $(A, \mathbf{e}, \mathbf{m}, O)$; ($r_1$)
4. $\quad$ Merge-Sort $(A, \mathbf{m}+1, \mathbf{d}, O)$; ($r_5$)
5. $\quad$ Intercala $(A, \mathbf{e}, \mathbf{m}, \mathbf{d}, O)$;



m_1 m_5 m_0

e_5 d_5

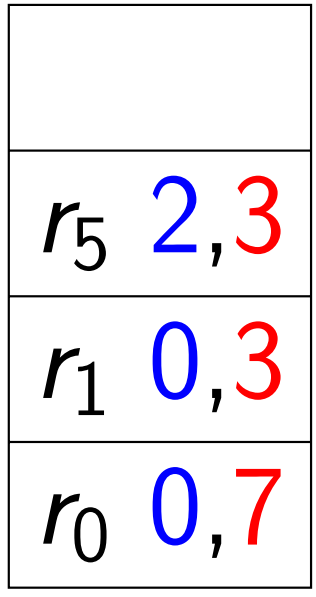
$A =$

2	5	7	4	8	1	9	7
0	1	2	3	4	5	6	7

Merge-Sort

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

1. se $\mathbf{e} < \mathbf{d}$ então
2. $\quad \mathbf{m} \leftarrow \lfloor (\mathbf{e} + \mathbf{d})/2 \rfloor$;
3. $\quad$ Merge-Sort $(A, \mathbf{e}, \mathbf{m}, O)$; r_1)
4. $\quad$ Merge-Sort $(A, \mathbf{m}+1, \mathbf{d}, O)$; r_5)
5. $\quad$ Intercala $(A, \mathbf{e}, \mathbf{m}, \mathbf{d}, O)$; ▷



pilha (recursão)

m_1 m_5 m_0

e_5 d_5

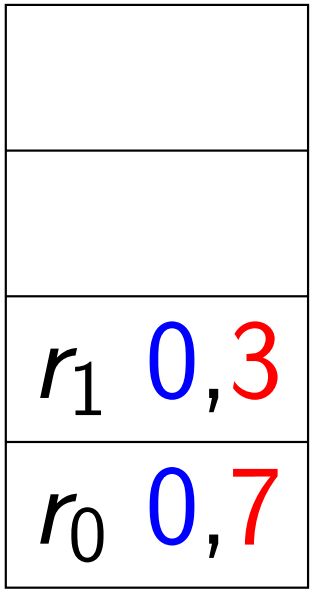
$A =$

2	5	4	7	8	1	9	7
0	1	2	3	4	5	6	7

Merge-Sort

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

1. se $\mathbf{e} < \mathbf{d}$ então
2. $\quad \mathbf{m} \leftarrow \lfloor (\mathbf{e} + \mathbf{d})/2 \rfloor$;
r_1) 3. $\quad$ Merge-Sort $(A, \mathbf{e}, \mathbf{m}, O)$;
4. $\quad$ Merge-Sort $(A, \mathbf{m} + 1, \mathbf{d}, O)$;
5. $\quad$ Intercala $(A, \mathbf{e}, \mathbf{m}, \mathbf{d}, O)$;



pilha (recursão)

m_1 m_0

e_1 d_1

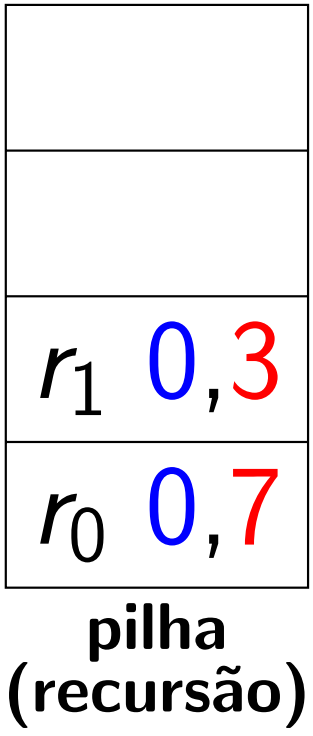
$A =$

2	5	4	7	8	1	9	7
0	1	2	3	4	5	6	7

Merge-Sort

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

1. se $\mathbf{e} < \mathbf{d}$ então
2. $\quad \mathbf{m} \leftarrow \lfloor (\mathbf{e} + \mathbf{d})/2 \rfloor$;
r_1) 3. $\quad$ Merge-Sort $(A, \mathbf{e}, \mathbf{m}, O)$;
4. $\quad$ Merge-Sort $(A, \mathbf{m} + 1, \mathbf{d}, O)$;
$\triangleright$ 5. $\quad$ Intercala $(A, \mathbf{e}, \mathbf{m}, \mathbf{d}, O)$;



pilha (recursão)

m_1 m_0

e_1 d_1

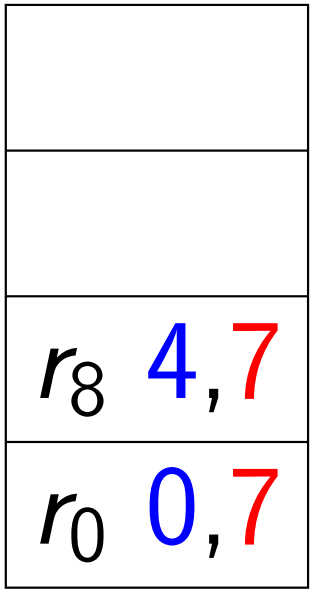
$A =$

2	4	5	7	8	1	9	7
0	1	2	3	4	5	6	7

Merge-Sort

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

1. se $\mathbf{e} < \mathbf{d}$ então
2. $\quad \mathbf{m} \leftarrow \lfloor (\mathbf{e} + \mathbf{d})/2 \rfloor$;
3. $\quad$ Merge-Sort $(A, \mathbf{e}, \mathbf{m}, O)$;
4. $\quad$ Merge-Sort $(A, \mathbf{m} + 1, \mathbf{d}, O)$; r_8)
5. $\quad$ Intercala $(A, \mathbf{e}, \mathbf{m}, \mathbf{d}, O)$;



pilha (recursão)

m_0 m_8

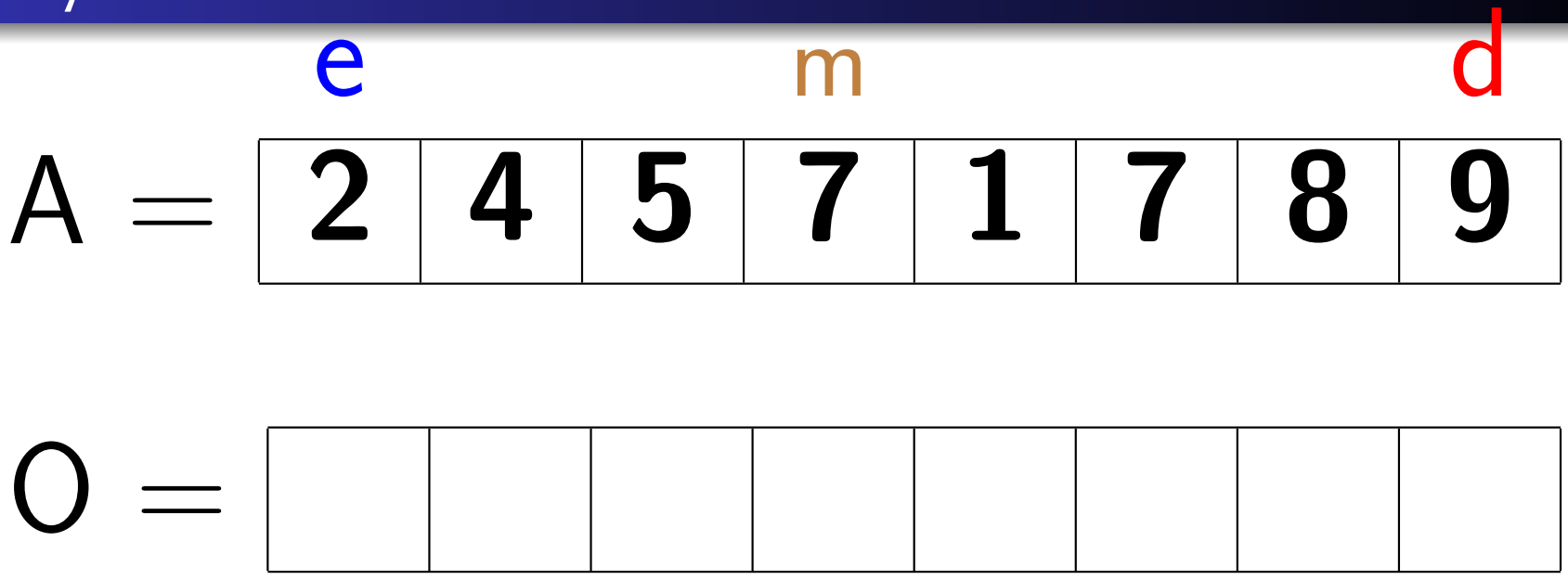
e_8 d_8

$A =$

2	4	5	7	8	1	9	7
0	1	2	3	4	5	6	7

Sumário

Função intercala: funcionamento



INTERCALA (A, e, m, d, O)

Função intercala: funcionamento

INTERCALA (A, e, m, d, O)

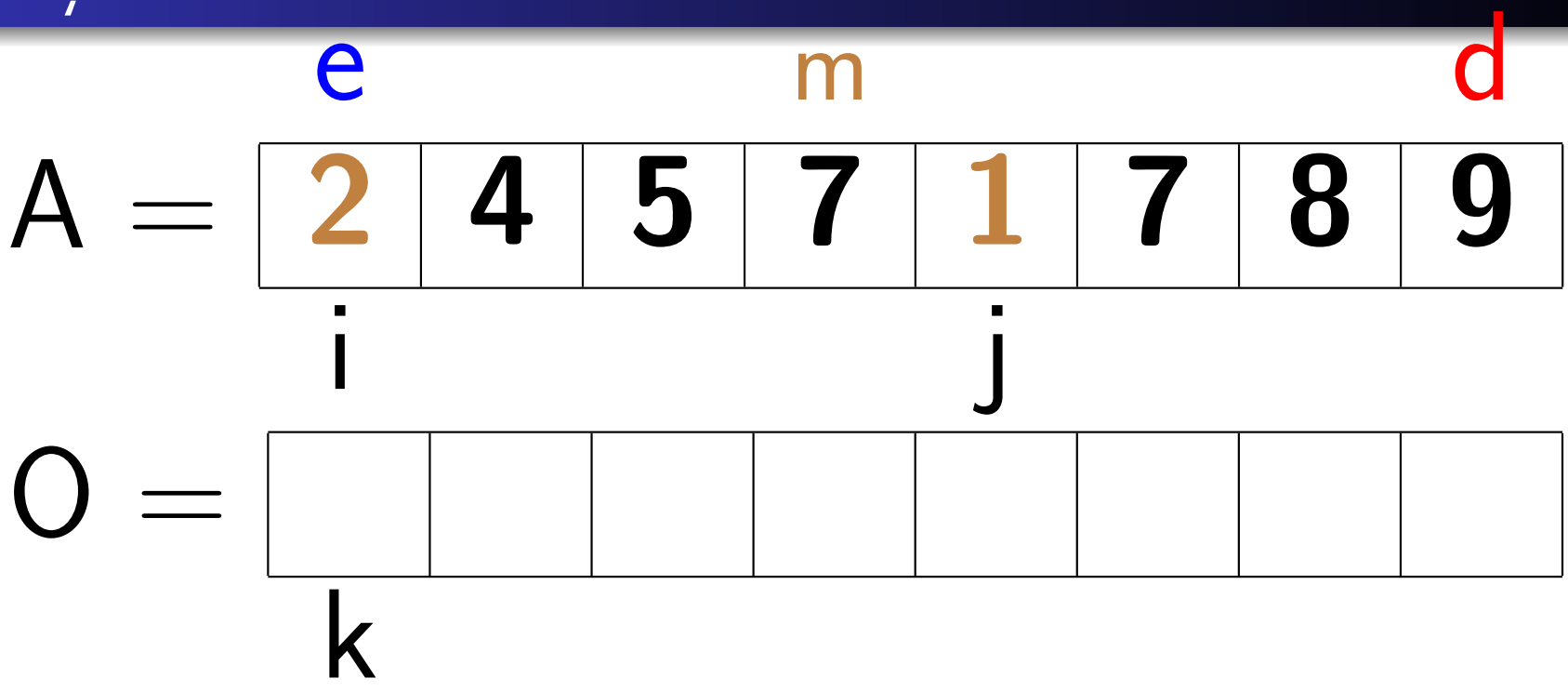
$i \leftarrow e$

$j \leftarrow m + 1$

$k \leftarrow e$

...

Função intercala: funcionamento



INTERCALA (A, e, m, d, O)

Enquanto $(i \leq m)$ e $(j \leq d)$ **faça**

$\triangleright$ **Se** $A[i] \leq A[j]$ **então**

$O[k{+}{+}] \leftarrow A[i{+}{+}]$;

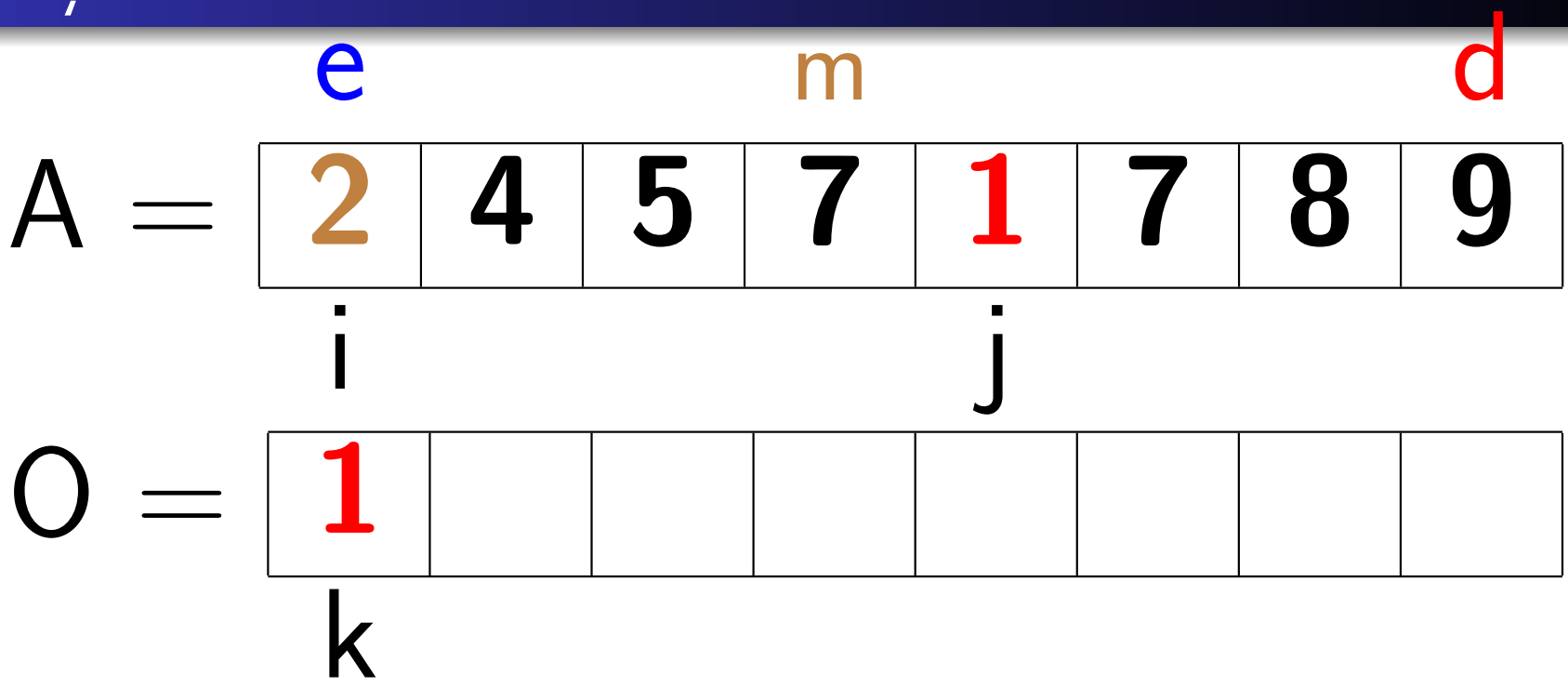
Senão

$O[k{+}{+}] \leftarrow A[j{+}{+}]$;

fim

...

Função intercala: funcionamento



INTERCALA (A, e, m, d, O)

Enquanto $(i \leq m)$ e $(j \leq d)$ **faça**

Se $A[i] \leq A[j]$ **então**

$O[k{+}{+}] \leftarrow A[i{+}{+}]$;

Senão

$\triangleright$ $O[k{+}{+}] \leftarrow A[j{+}{+}]$;

fim

...

Função intercala: funcionamento

INTERCALA (A, e, m, d, O)

Enquanto $(i \leq m)$ e $(j \leq d)$ **faça**

$\triangleright$ **Se** $A[i] \leq A[j]$ **então**

$O[k{+}{+}] \leftarrow A[i{+}{+}]$;

Senão

$O[k{+}{+}] \leftarrow A[j{+}{+}]$;

fim

...

Função intercala: funcionamento

INTERCALA (A, e, m, d, O)

Enquanto $(i \leq m)$ e $(j \leq d)$ **faça**

Se $A[i] \leq A[j]$ **então**

$\triangleright$ $O[k{+}{+}] \leftarrow A[i{+}{+}]$;

Senão

$O[k{+}{+}] \leftarrow A[j{+}{+}]$;

fim

...

Função intercala: funcionamento

INTERCALA (A, e, m, d, O)

Enquanto $(i \leq m)$ e $(j \leq d)$ **faça**

$\triangleright$ **Se** $A[i] \leq A[j]$ **então**

$O[k{+}{+}] \leftarrow A[i{+}{+}]$;

Senão

$O[k{+}{+}] \leftarrow A[j{+}{+}]$;

fim

...

Função intercala: funcionamento

INTERCALA (A, e, m, d, O)

Enquanto $(i \leq m)$ e $(j \leq d)$ **faça**

Se $A[i] \leq A[j]$ **então**

$\triangleright$ $O[k{+}{+}] \leftarrow A[i{+}{+}]$;

Senão

$O[k{+}{+}] \leftarrow A[j{+}{+}]$;

fim

...

Função intercala: funcionamento

INTERCALA (A, e, m, d, O)

Enquanto $(i \leq m)$ e $(j \leq d)$ **faça**

$\triangleright$ **Se** $A[i] \leq A[j]$ **então**

$O[k{+}{+}] \leftarrow A[i{+}{+}]$;

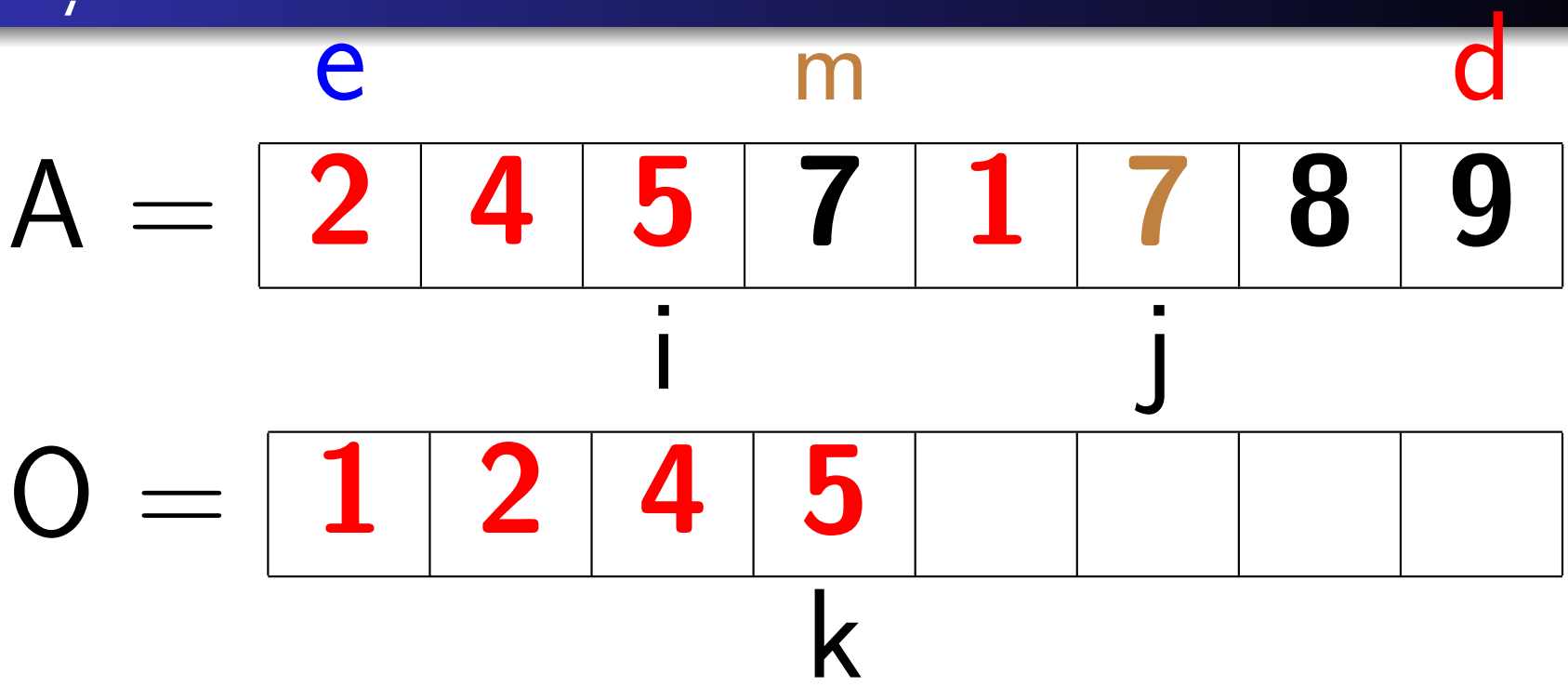
Senão

$O[k{+}{+}] \leftarrow A[j{+}{+}]$;

fim

...

Função intercala: funcionamento



INTERCALA (A, e, m, d, O)

Enquanto $(i \leq m)$ e $(j \leq d)$ **faça**

Se $A[i] \leq A[j]$ **então**

$\triangleright$ $O[k{+}{+}] \leftarrow A[i{+}{+}]$;

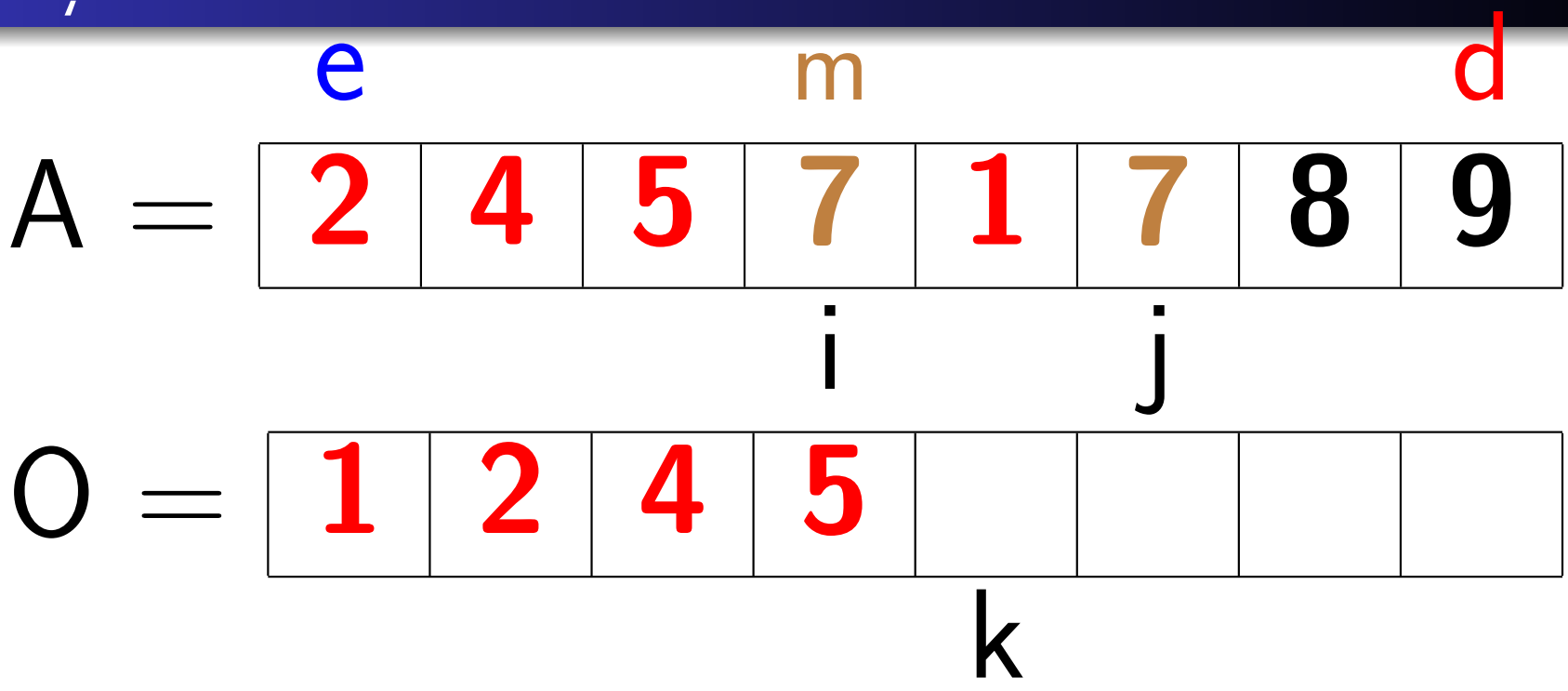
Senão

$O[k{+}{+}] \leftarrow A[j{+}{+}]$;

fim

...

Função intercala: funcionamento



INTERCALA (A, e, m, d, O)

Enquanto $(i \leq m)$ e $(j \leq d)$ **faça**

$\triangleright$ **Se** $A[i] \leq A[j]$ **então**

$O[k{+}{+}] \leftarrow A[i{+}{+}]$;

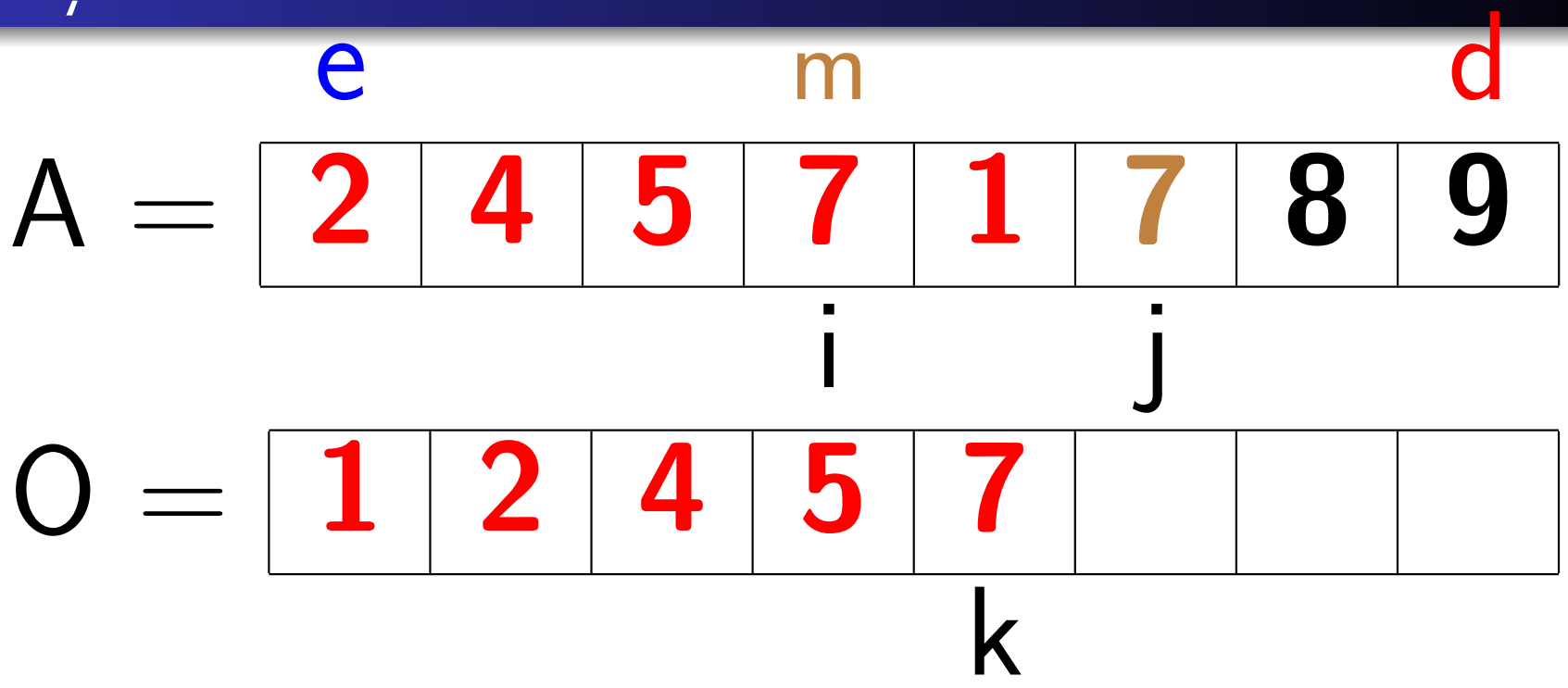
Senão

$O[k{+}{+}] \leftarrow A[j{+}{+}]$;

fim

...

Função intercala: funcionamento



INTERCALA (A, e, m, d, O)

Enquanto $(i \leq m)$ e $(j \leq d)$ **faça**

Se $A[i] \leq A[j]$ **então**

$\triangleright$ $O[k{+}{+}] \leftarrow A[i{+}{+}]$;

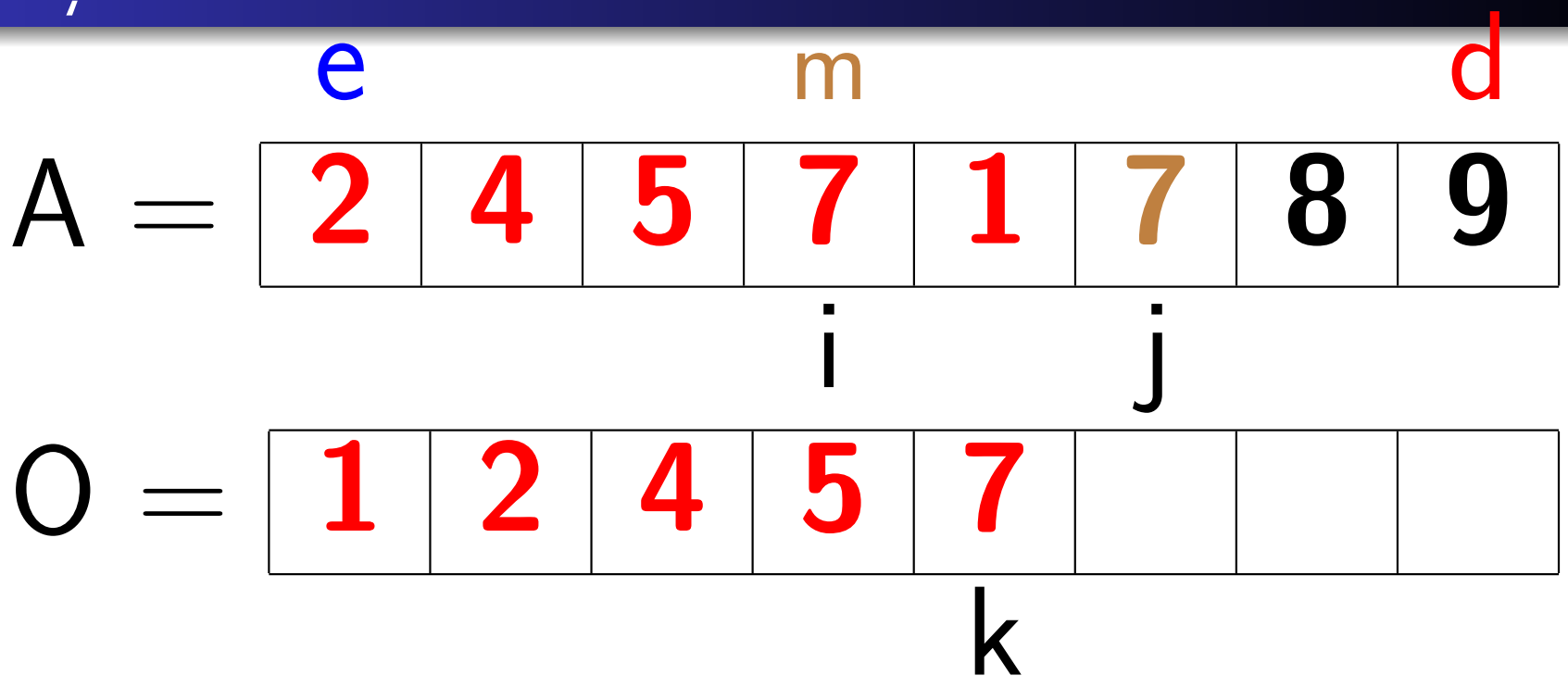
Senão

$O[k{+}{+}] \leftarrow A[j{+}{+}]$;

fim

...

Função intercala: funcionamento



INTERCALA (A, e, m, d, O)

▷ **Enquanto** $(i \leq m)$ e $(j \leq d)$ **faça**

Se $A[i] \leq A[j]$ **então**

$O[k{+}{+}] \leftarrow A[i{+}{+}]$;

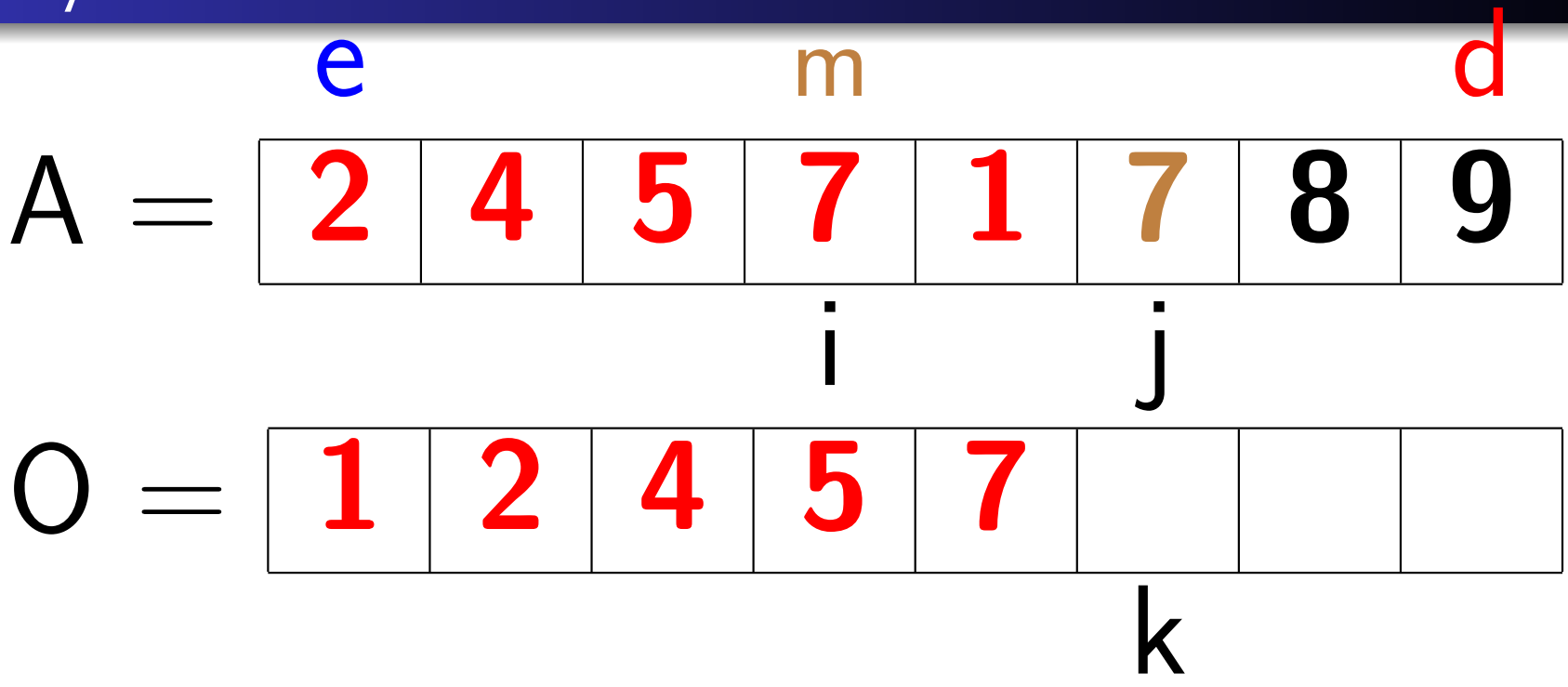
Senão

$O[k{+}{+}] \leftarrow A[j{+}{+}]$;

fim

...

Função intercala: funcionamento

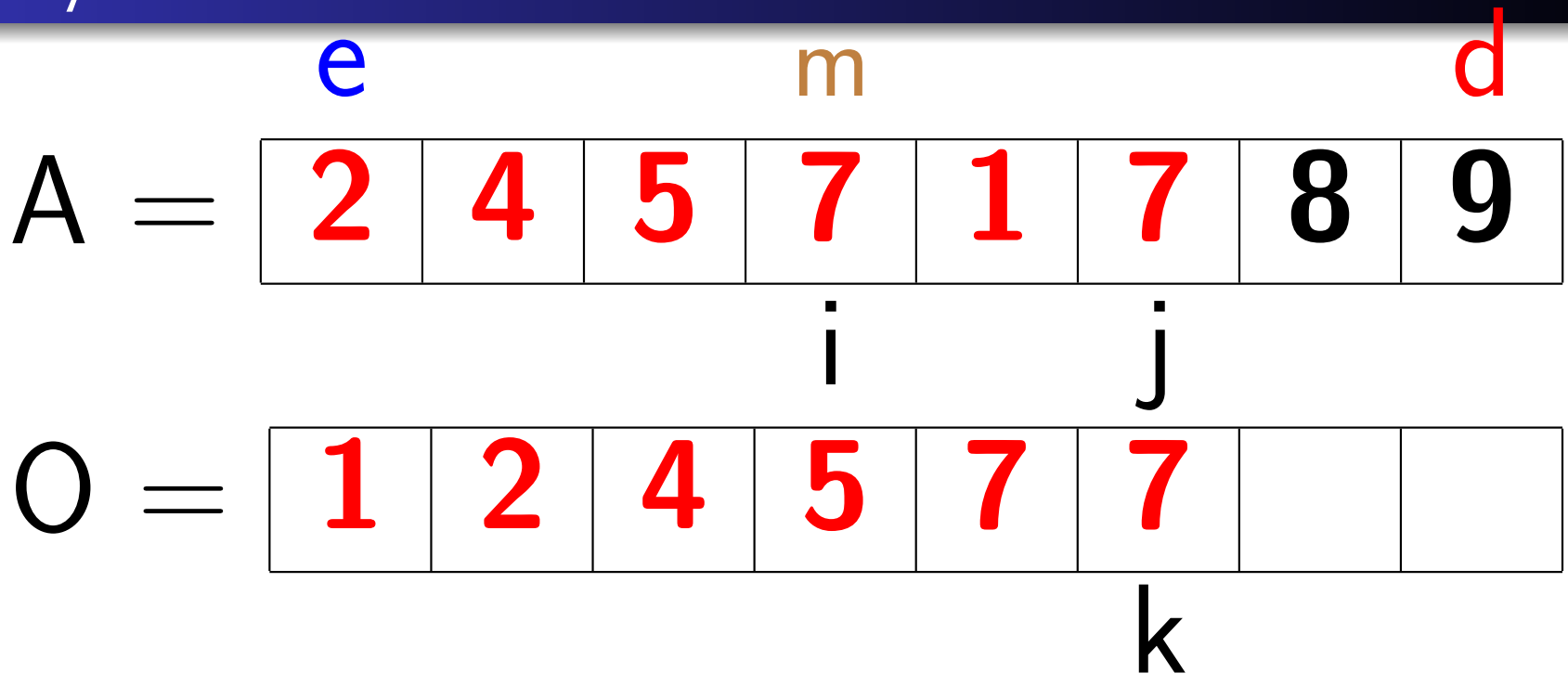


INTERCALA (A, e, m, d, O)

▷ **Enquanto** $(j \leq d)$ **faça**

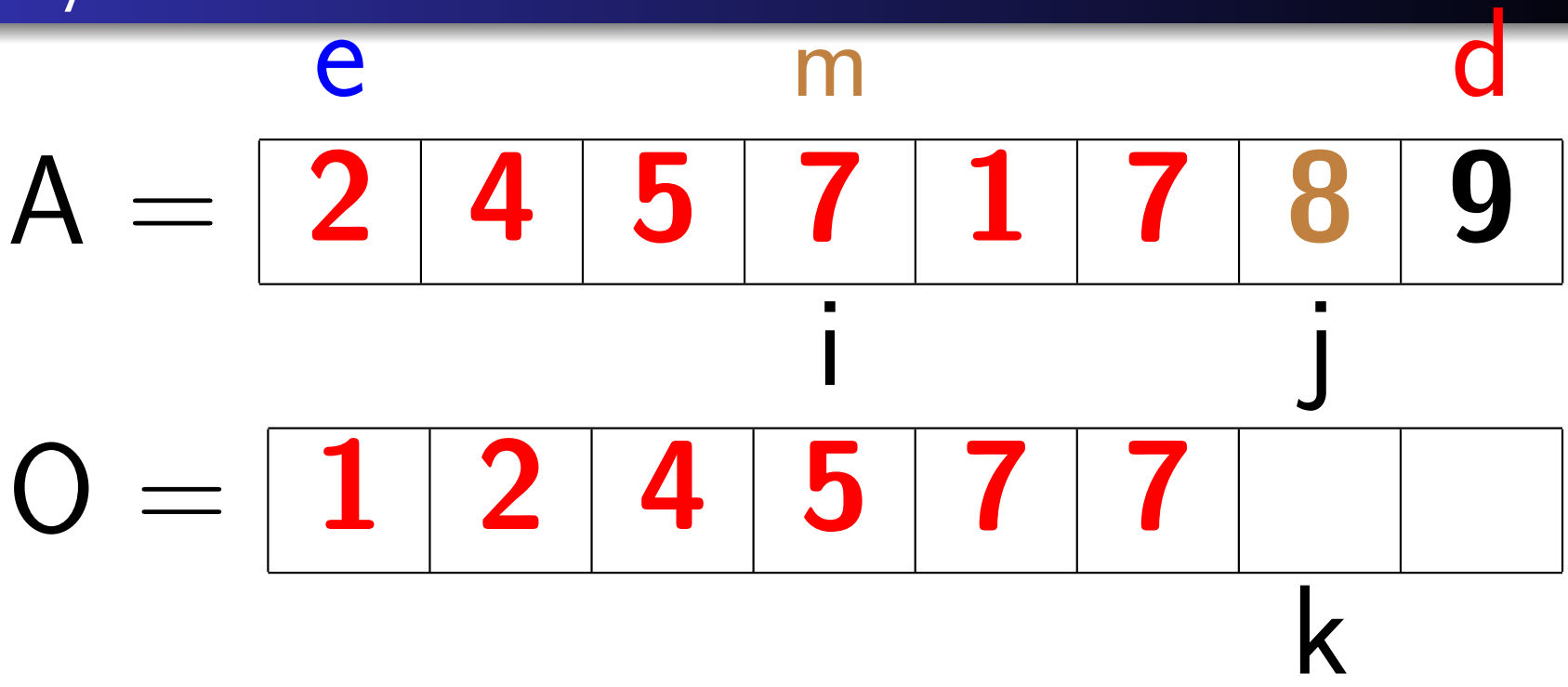
$O[k{+}{+}] \leftarrow A[j{+}{+}];$

Função intercala: funcionamento



INTERCALA (A, e, m, d, O)
Enquanto $(j \leq d)$ **faça**
$\triangleright$ $O[k{+}{+}] \leftarrow A[j{+}{+}]$;

Função intercala: funcionamento

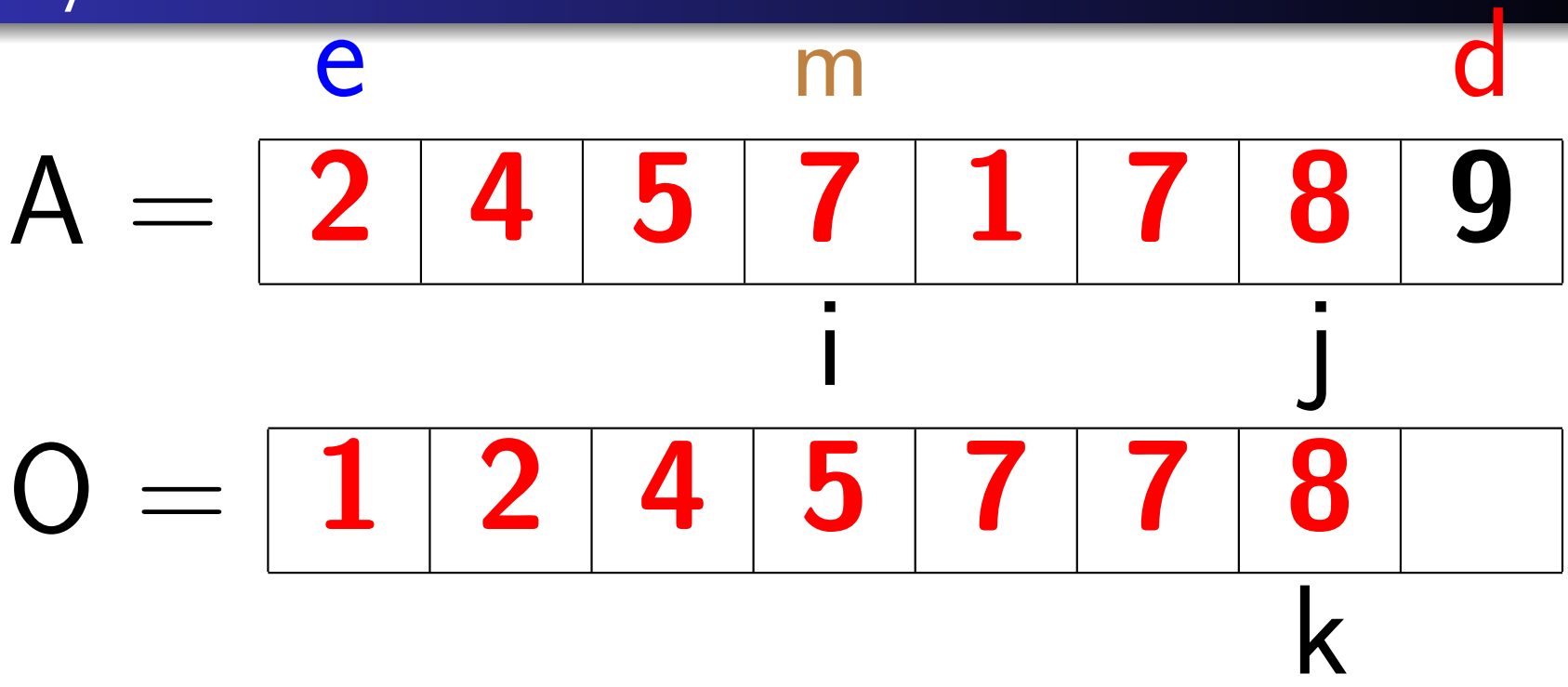


INTERCALA (A, e, m, d, O)

▷ **Enquanto** $(j \leq d)$ **faça**

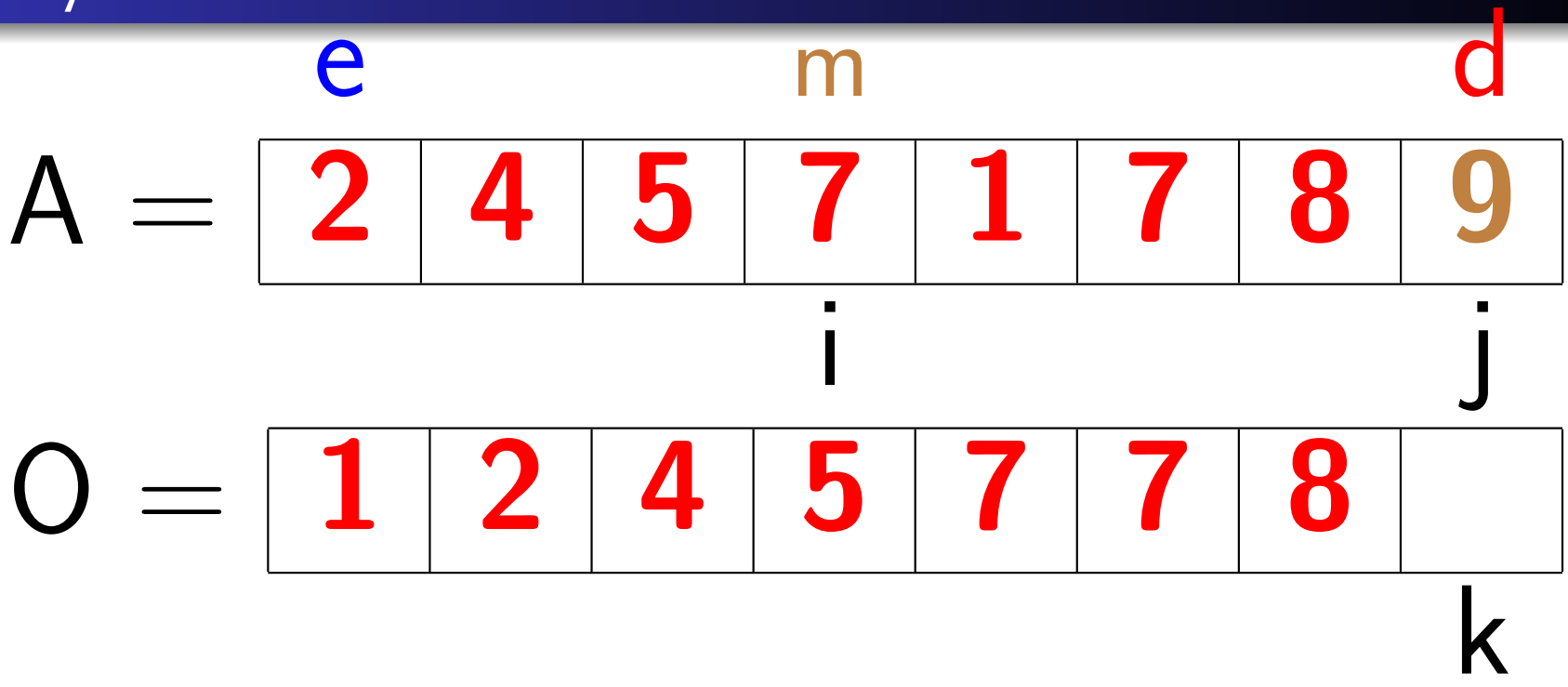
$O[k{+}{+}] \leftarrow A[j{+}{+}]$;

Função intercala: funcionamento



INTERCALA (A, e, m, d, O)
Enquanto $(j \leq d)$ **faça**
$\triangleright$ $O[k{+}{+}] \leftarrow A[j{+}{+}]$;

Função intercala: funcionamento

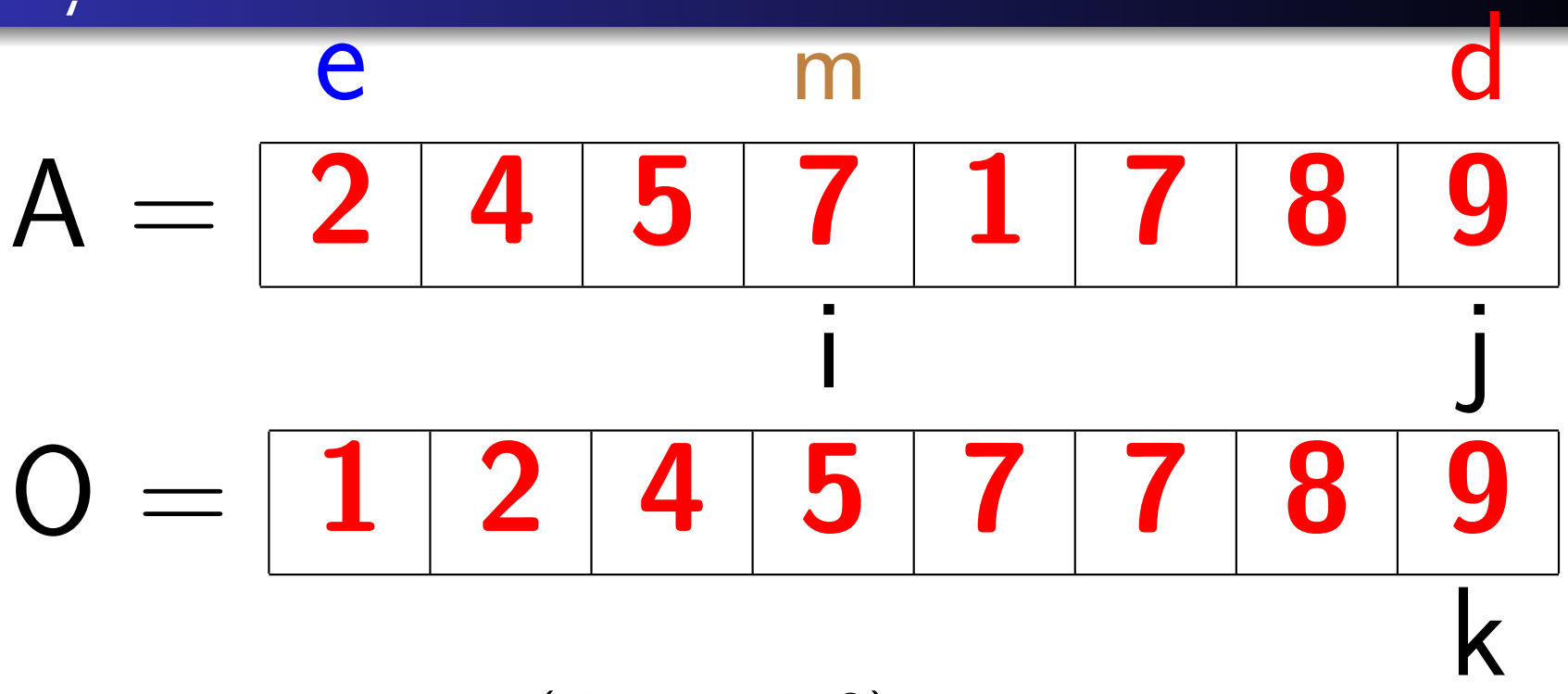


INTERCALA (A, e, m, d, O)

▷ **Enquanto** $(j \leq d)$ **faça**

$O[k{+}{+}] \leftarrow A[j{+}{+}]$;

Função intercala: funcionamento



INTERCALA (A, e, m, d, O)
Enquanto $(j \leq d)$ **faça**
$\triangleright$ $O[k{+}{+}] \leftarrow A[j{+}{+}]$;

Sumário

Complexidade

O algoritmo do Merge-Sort é estável e sua equação de recorrência é dado por:

$$T(\mathbf{n}) = \begin{cases} \Theta(1) & \text{se } \mathbf{n} = 1, \\ T(\lceil \mathbf{n}/2 \rceil) + T(\lfloor \mathbf{n}/2 \rfloor) + \Theta(\mathbf{n}) & \text{se } \mathbf{n} > 1. \end{cases}$$

Complexidade de tempo: $\Theta(n \log n)$.

Complexidade de espaço: $\Theta(n)$.