



Métodos de Ordenação Lineares

Prof. Lilian Berton

São José dos Campos, 2018

Baseado no material de Thomas Cormen

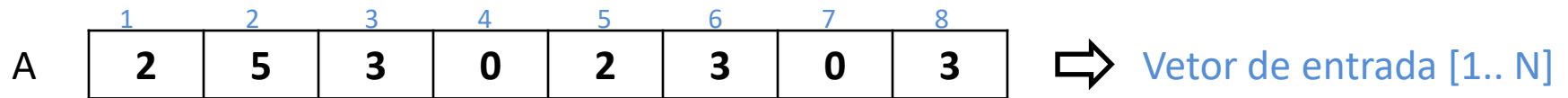
Algoritmos de ordenação em memória interna

- **Quadráticos $O(n^2)$:**
 - Ordenação por Seleção (Selection Sort)
 - Ordenação por Inserção (Insertion Sort)
 - Ordenação por Bolha (BubbleSort)
- **$O(n \log n)$:**
 - Quicksort
 - Heapsort
 - Shellsort
- **Lineares $O(n)$:**
 - Ordenação por contagem
 - Radix sort
 - Bucket sort

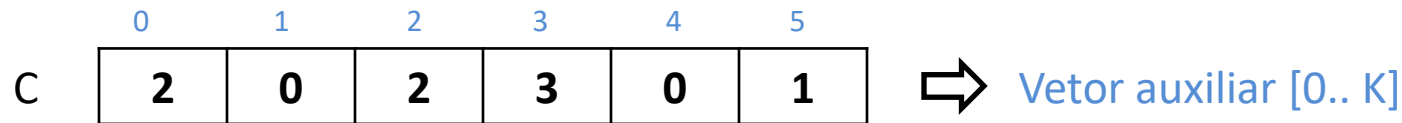
Ordenação por contagem

- **Algoritmo:**
- Pressupõe que cada um dos n elementos de entrada é um inteiro no intervalo de 1 a k .
- Parte-se da suposição que a entrada é um array **$A[1..n]$** .
- Necessita-se outros dois arrays **$B[1..n]$** que contém a saída ordenada e o arranjo **$C[0..k]$** que fornece um espaço de armazenamento temporário.
- É estável. Por isso é usada como subrotina no radixsort.

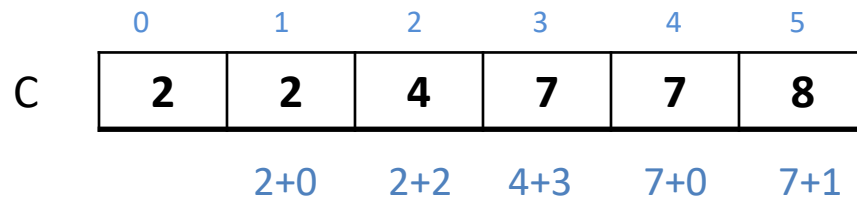
Exemplo ordenação por contagem



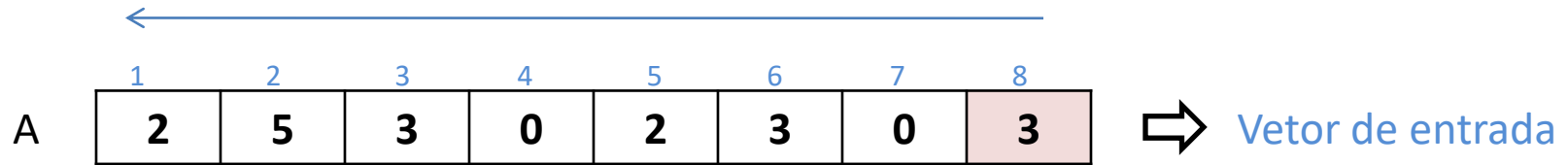
Calcule a quantidade de cada elemento e armazene em cada posição de C:



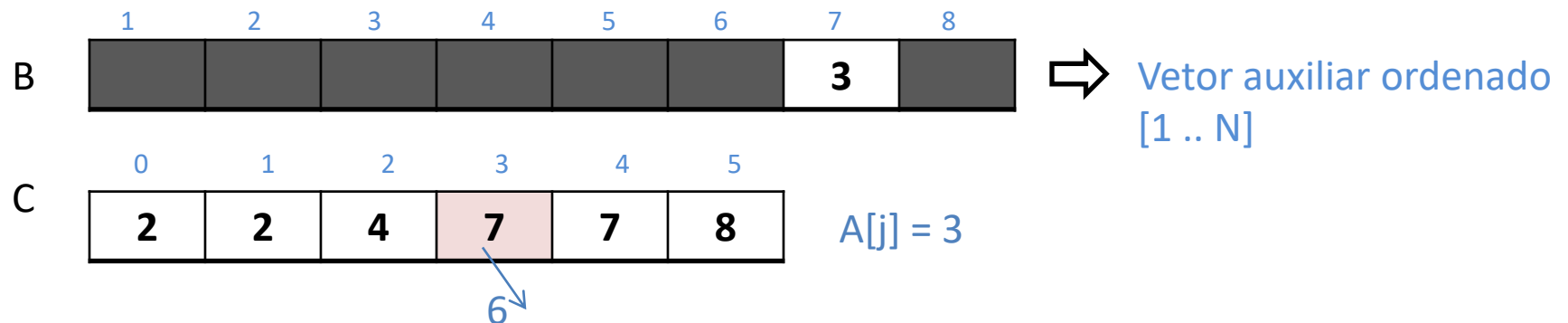
Percorra C e some cada i-ésimo valor na posição seguinte:



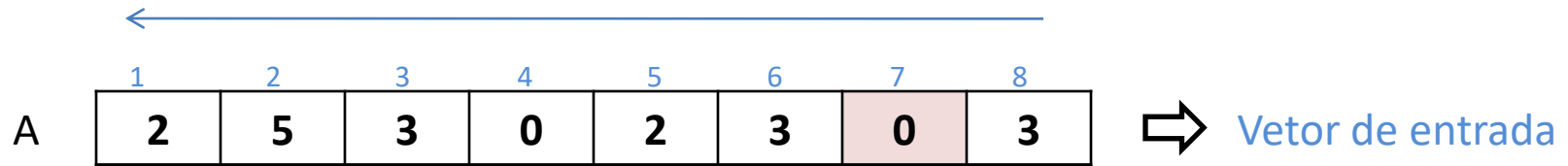
Exemplo ordenação por contagem



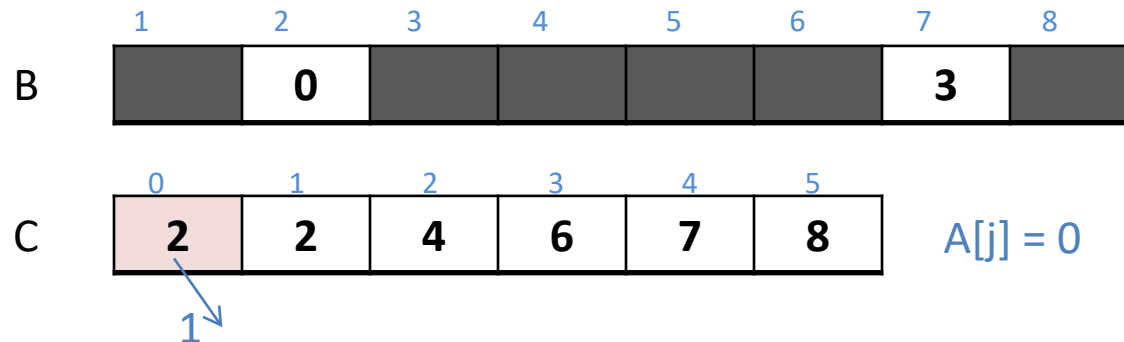
- Percorra o vetor de entrada A de n-1 até 0
- Insira cada j-ésimo elemento na posição correta em B usando o índice em C
- Atualiza o valor em C



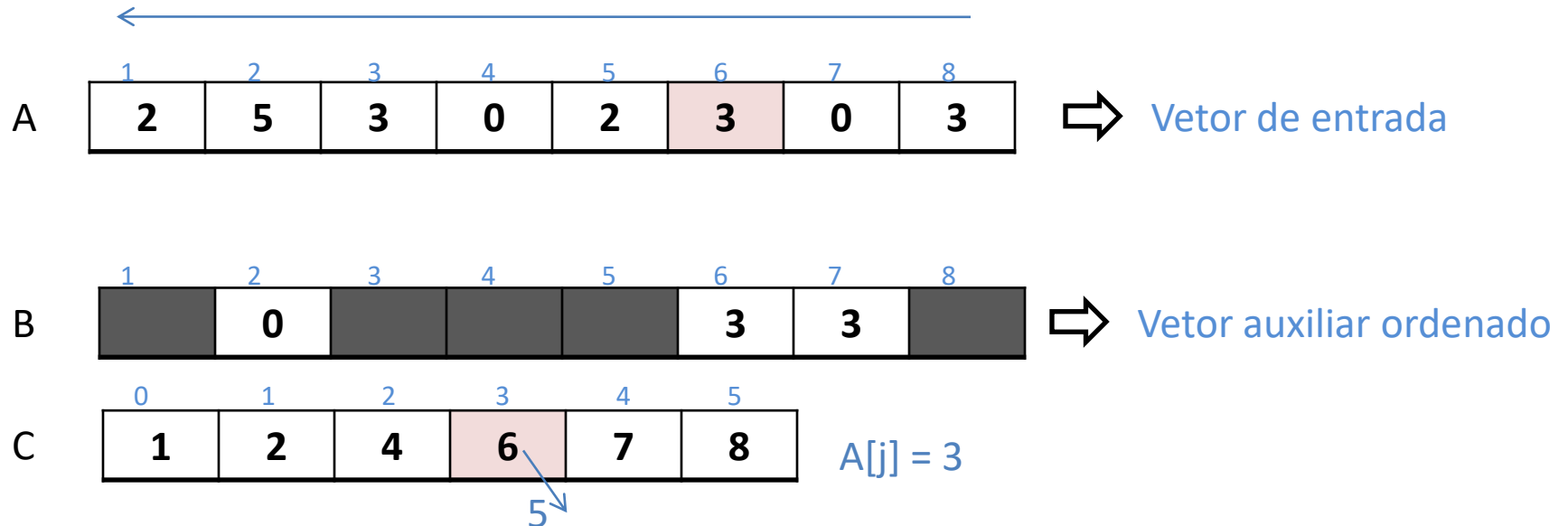
Exemplo ordenação por contagem



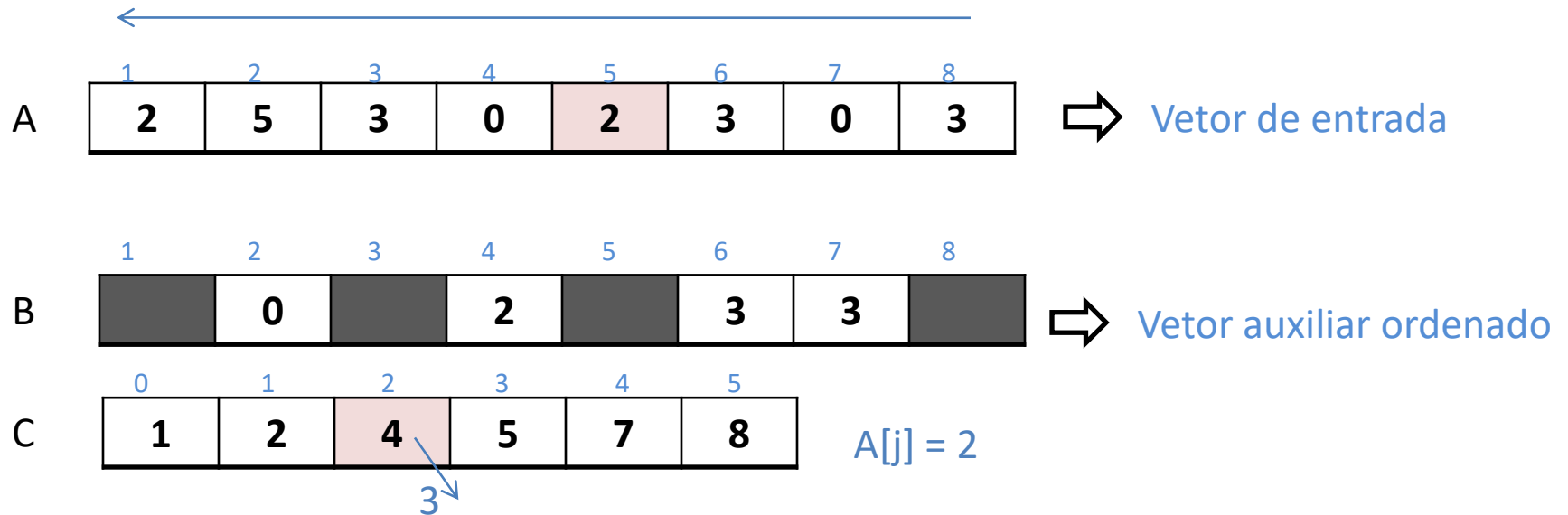
Repita esse processo, até o vetor A estar ordenado:



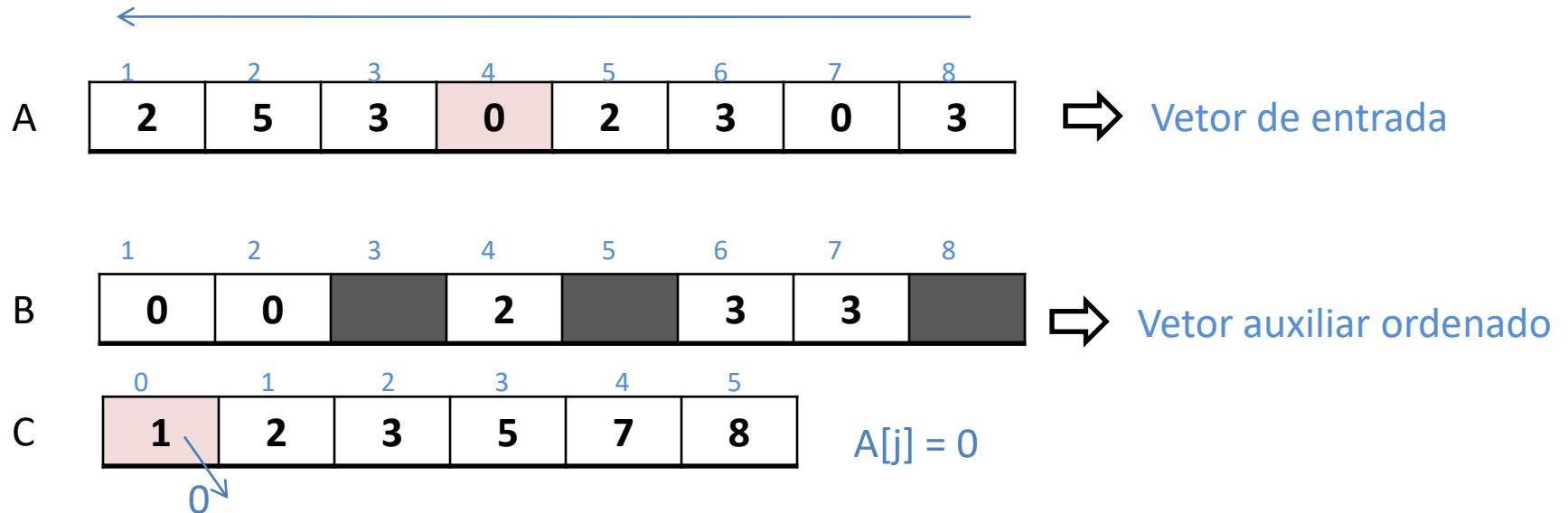
Exemplo ordenação por contagem



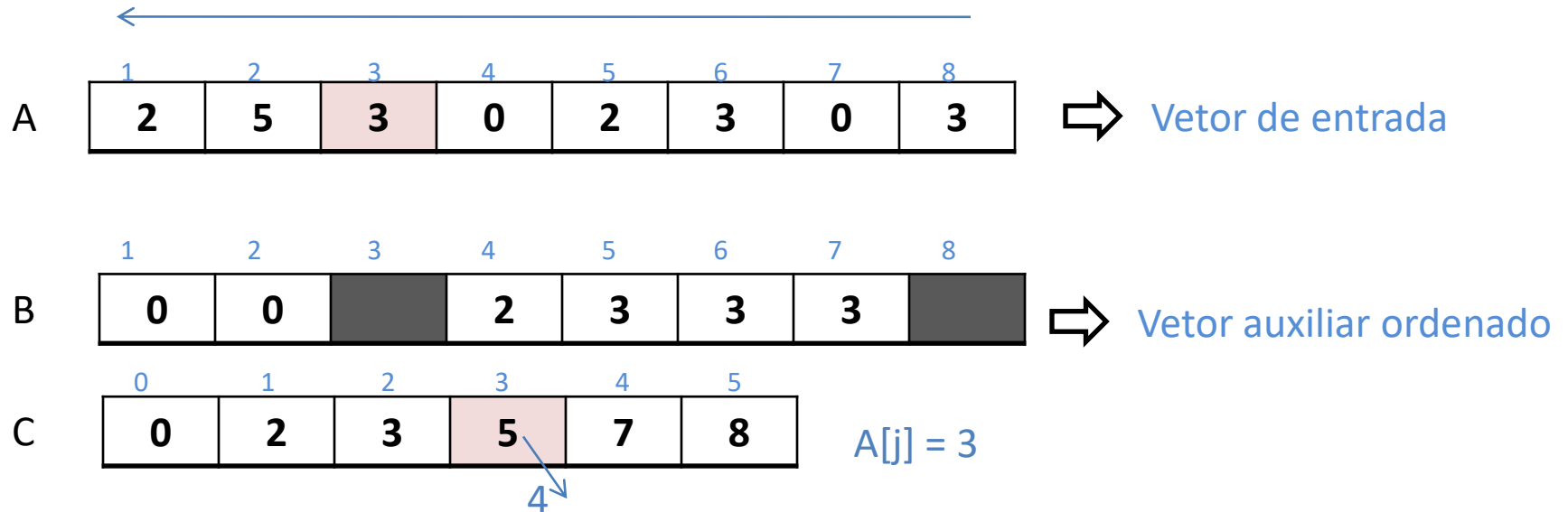
Exemplo ordenação por contagem



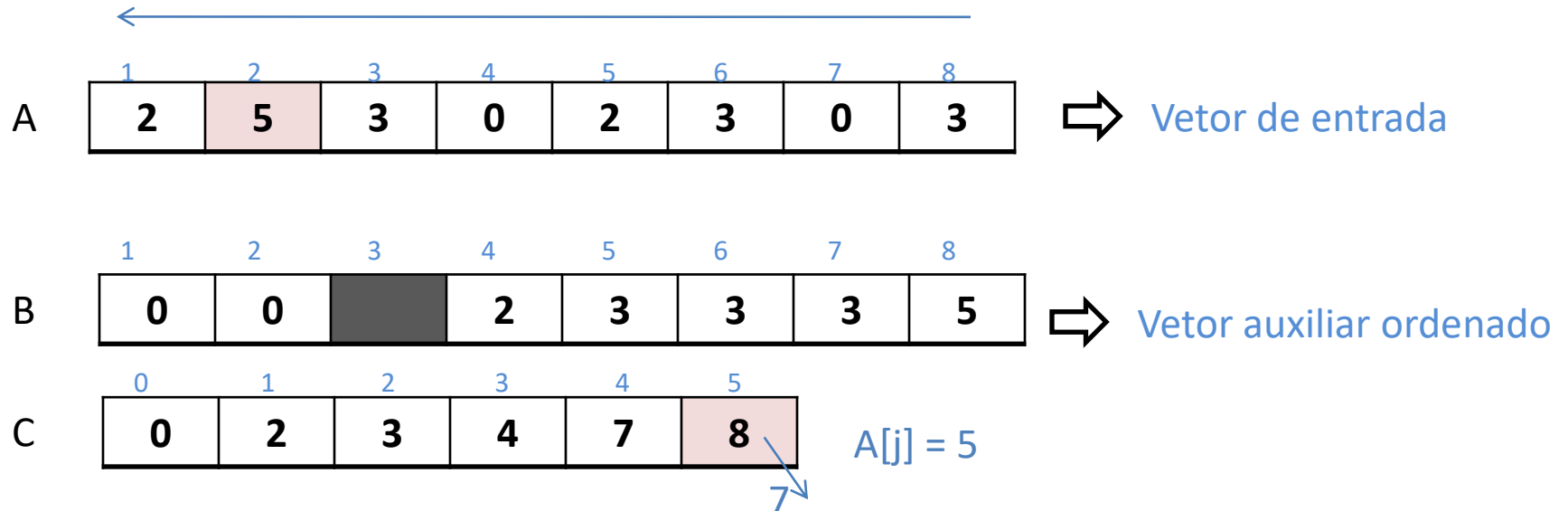
Exemplo ordenação por contagem



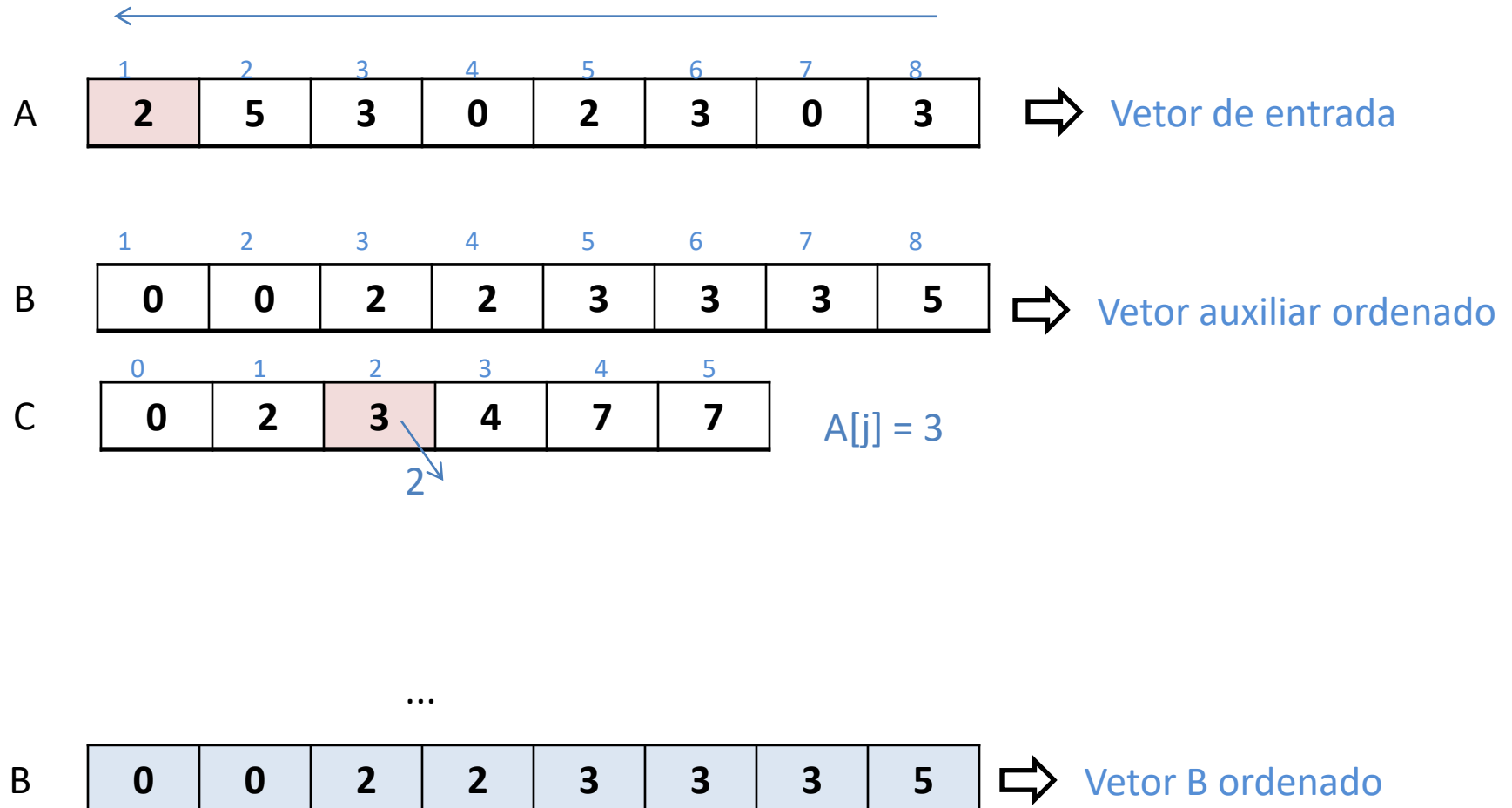
Exemplo ordenação por contagem



Exemplo ordenação por contagem



Exemplo ordenação por contagem



Ordenação por contagem

COUNTING-SORT(A, B, k)

$O(k)$ { 1 **for** $i \leftarrow 0$ **to** k
2 **do** $C[i] \leftarrow 0$

$O(n)$ { 3 **for** $j \leftarrow 1$ **to** *comprimento* $[A]$
4 **do** $C[A[j]] \leftarrow C[A[j]] + 1$

$O(k)$ { 5 \triangleright Agora $C[i]$ contém o número de elementos iguais a i .
6 **for** $i \leftarrow 2$ **to** k
7 **do** $C[i] \leftarrow C[i] + C[i - 1]$
8 \triangleright Agora $C[i]$ contém o número de elementos menores que ou iguais a i .

$O(n)$ { 9 **for** $j \leftarrow$ *comprimento* $[A]$ **downto** 1
10 **do** $B[C[A[j]]] \leftarrow A[j]$
11 $C[A[j]] \leftarrow C[A[j]] - 1$

Total = $O(k+n) = O(n)$

Ordenação por contagem

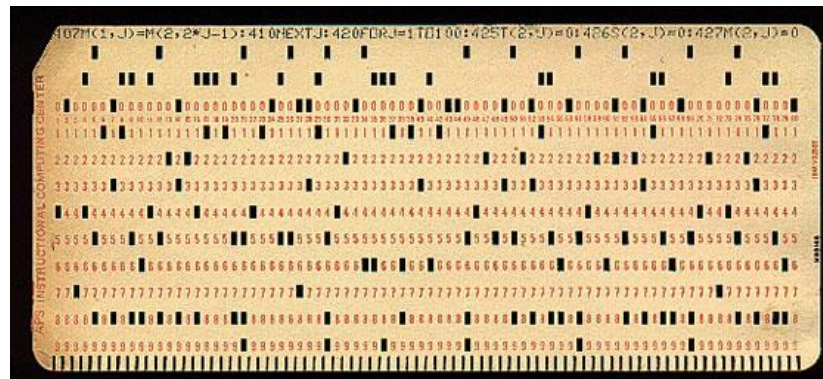
- Não realiza comparações entre os elementos.
- Utiliza os valores dos elementos para indexar em outro vetor.
- Utiliza memória adicional
 - vetor $A[1..N]$, vetor $B[1..N]$, vetor $C[1..k]$
 - Qual pode ser o tamanho de k ?
- É estável e pode ser utilizada no radix sort.

Algoritmos de ordenação em memória interna

- **Quadráticos $O(n^2)$:**
 - Ordenação por Seleção (Selection Sort)
 - Ordenação por Inserção (Insertion Sort)
 - Ordenação por Bolha (BubbleSort)
- **$O(n \log n)$:**
 - Quicksort
 - Heapsort
 - Shellsort
- **Lineares $O(n)$:**
 - Ordenação por contagem
 - Radix sort
 - Bucket sort

Radix sort

- É o algoritmo usado pelas máquinas de ordenação de **cartões perfurados**.
- Os cartões são organizados em 80 colunas e cada coluna pode ser feita uma perfuração em uma das 12 posições. No caso de dígitos decimais, apenas 10 posições são utilizadas em cada coluna.
- O radixsort ordena primeiro sobre o dígito menos significativo. O processo continua até os cartões terem sido ordenados sobre todos os dígitos.
- A ordenação precisa ser estável, para não alterar a ordem dos cartões.
- Pode ser usado para ordenar registros com vários campos, como data.

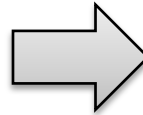


Exemplo radix sort

Algum algoritmo de ordenação

Ordena pelo 3º dígito

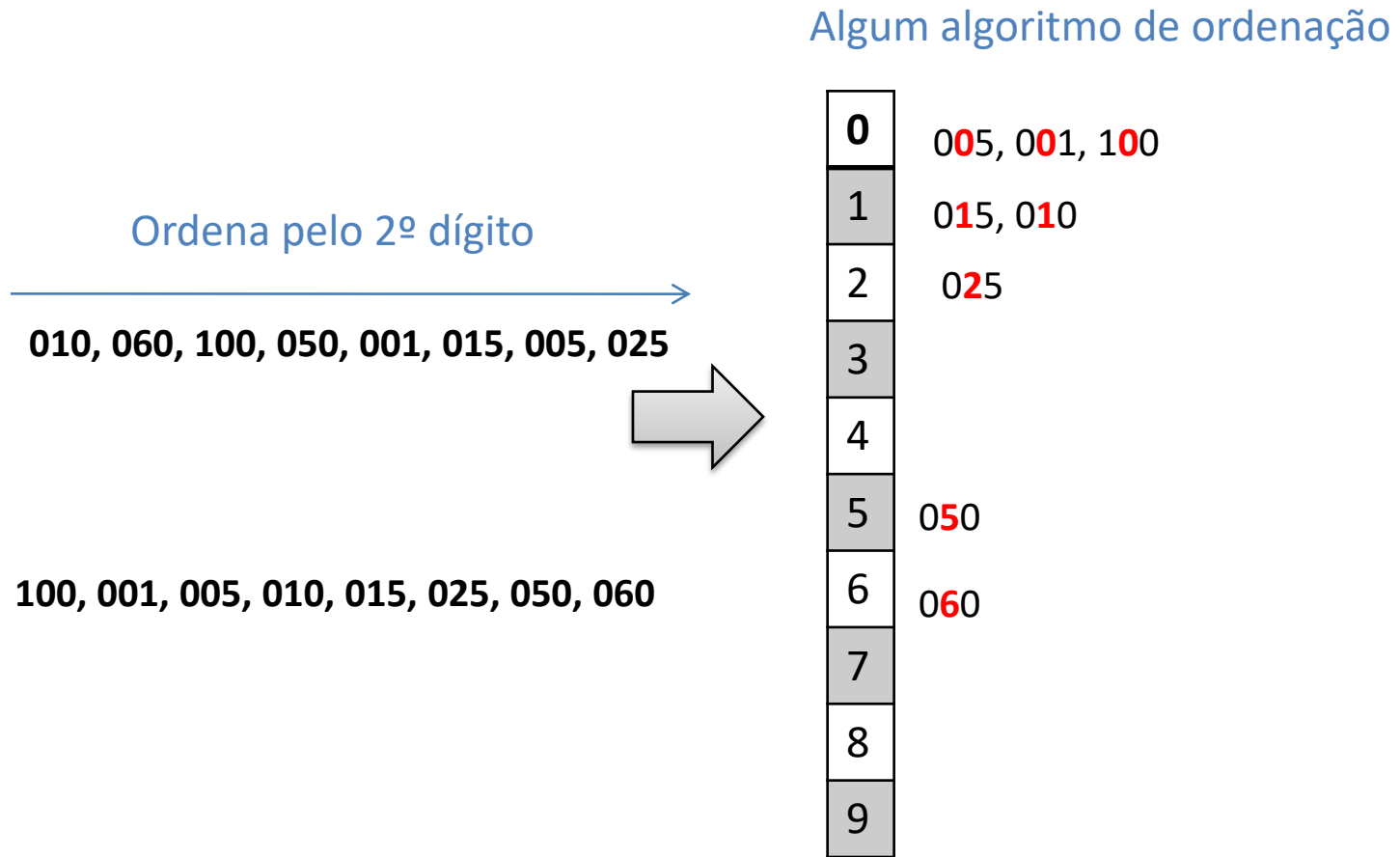
010, 015, 001, 060, 005, 100, 025, 050



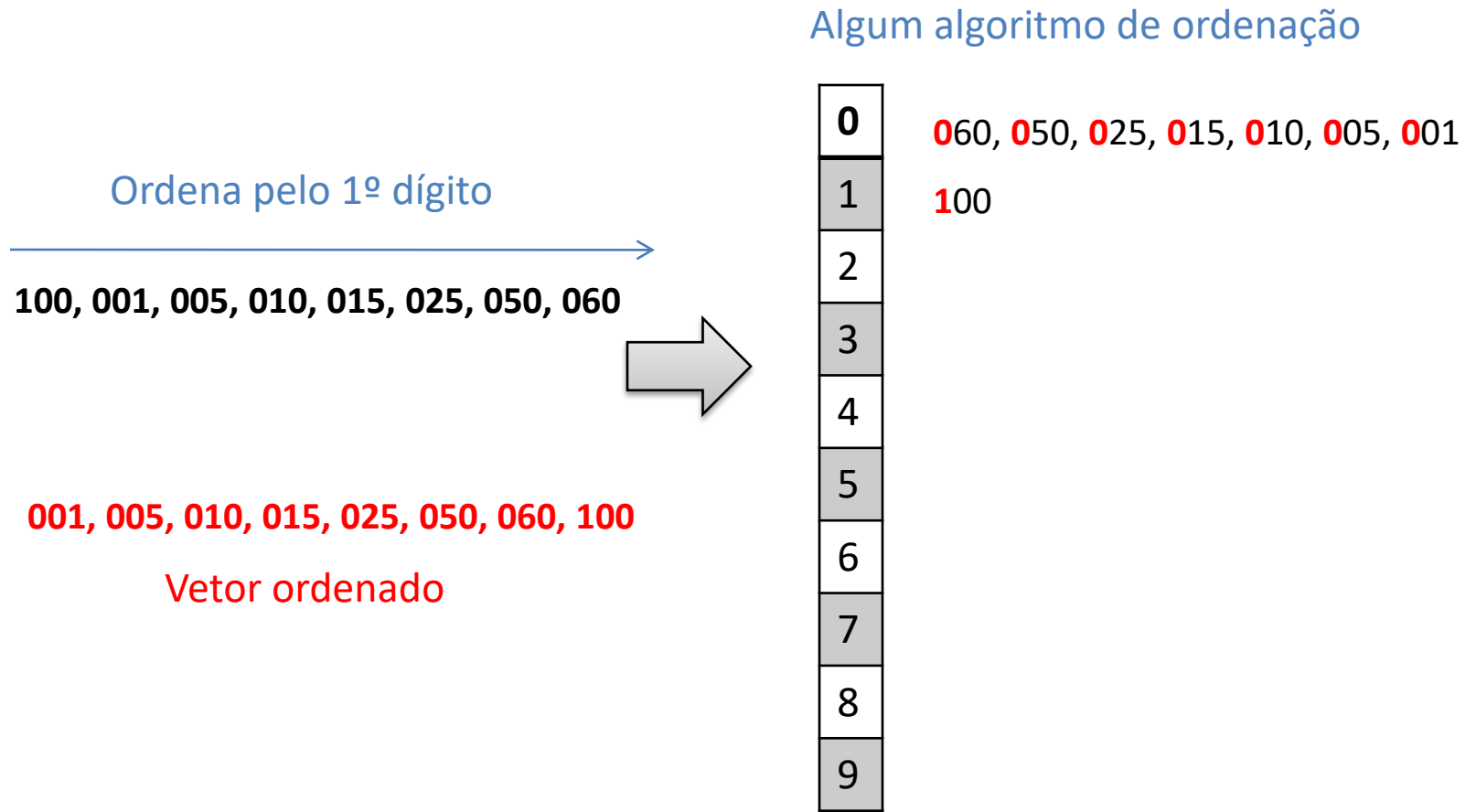
010, 060, 100, 050, 001, 015, 005, 025

0	050, 100, 060, 010
1	001
2	
3	
4	
5	025, 005, 015
6	
7	
8	
9	

Exemplo radix sort



Exemplo radix sort



Algoritmo radix sort

	1º	2º	3º
	↓	↓	↓
329	720	720	329
457	355	329	355
657	436	436	436
839	457	839	457
436	657	355	657
720	329	457	720
355	839	657	839

RADIX-SORT(A, d)

$O(d)$

1 for $i \leftarrow 1$ to d

$O(n)$

2 do usar uma ordenação estável para ordenar o arranjo A sobre o dígito i



Ordenação por contagem

Análise

- **Lema 1:** dados n números de d dígitos em que cada dígito pode assumir até k valores possíveis, radix sort ordena corretamente esses números no tempo $O(d(n+k))$.
- Quando d é constante e $k = O(n)$ radix sort é executado em tempo linear.
- **Lema 2:** Dados n números de b bits e qualquer inteiro positivo $r \leq b$, radix sort ordena corretamente esses números no tempo $O((b/r)(n + 2^r))$.
- Exemplo: dada uma palavra de 32 bits tendo 4 dígitos de 8 bits, temos $b = 32$, $r = 8$, $k = 2^r - 1 = 255$ e $d = b/r = 4$.

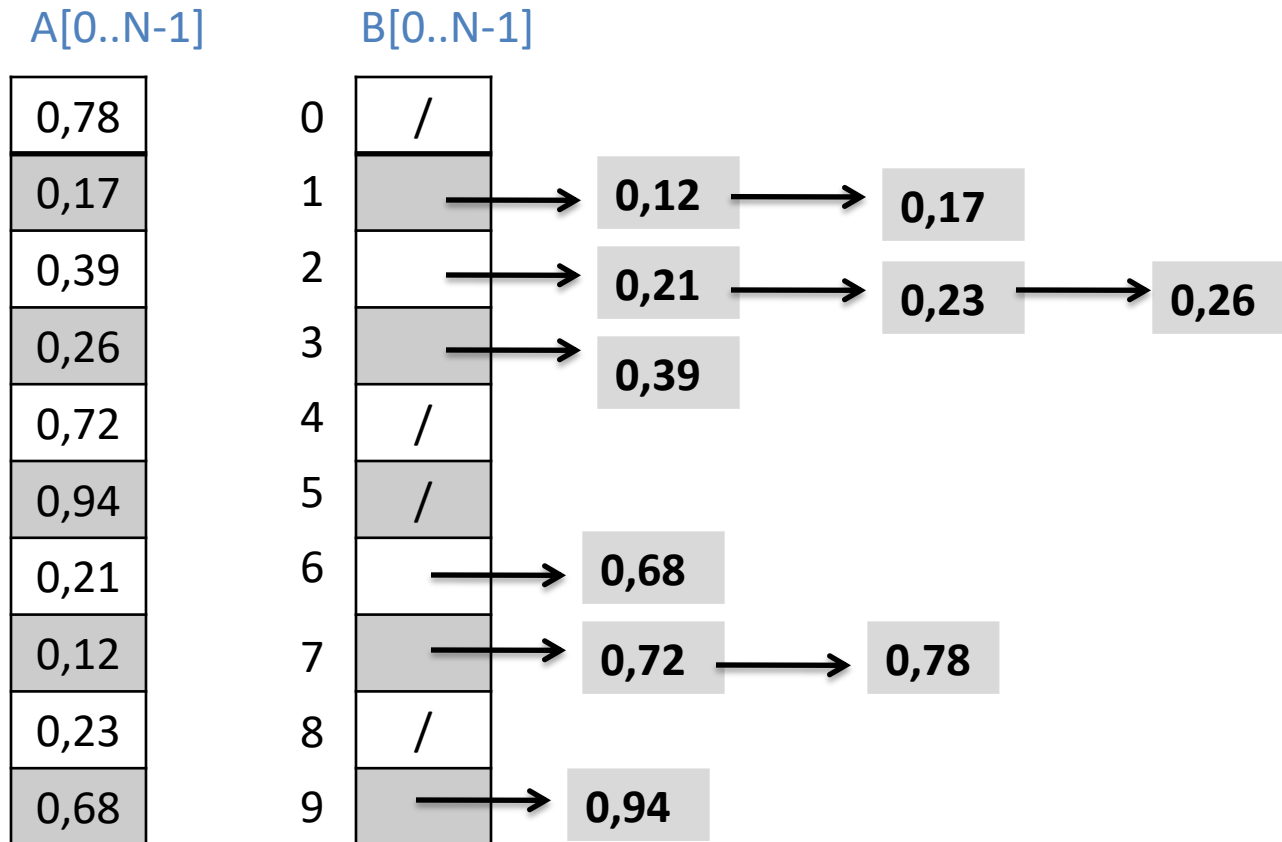
Algoritmos de ordenação em memória interna

- **Quadráticos $O(n^2)$:**
 - Ordenação por Seleção (Selection Sort)
 - Ordenação por Inserção (Insertion Sort)
 - Ordenação por Bolha (BubbleSort)
- **$O(n \log n)$:**
 - Quicksort
 - Heapsort
 - Shellsort
- **Lineares $O(n)$:**
 - Ordenação por contagem
 - Radix sort
 - Bucket sort

Bucket sort

- **Algoritmo:**
- Pressupõe que a entrada é gerada sobre um processo aleatório que distribui elementos uniformemente sobre o **intervalo $[0,1)$** .
- Divide o intervalo $[0,1)$ em n subintervalos (ou baldes) e distribui os n números entre os baldes.
- Como as **entradas são uniformemente distribuídas**, não espera-se que muitos números caiam em cada balde.
- A entrada é um vetor de n elementos $A[0..N-1]$ e cada elemento satisfaz $0 \leq A[i] \leq 1$. Utiliza-se um vetor auxiliar $B[0..N-1]$ de listas ligadas.

Exemplo bucket sort



Algoritmo bucket sort

BUCKET-SORT(A)

1 $n \leftarrow \text{comprimento}[A]$

$O(n)$ 2 **for** $i \leftarrow 1$ **to** n

3 **do** inserir $A[i]$ na lista $B[\lfloor nA[i] \rfloor]$

$O(n)$ 4 **for** $i \leftarrow 0$ **to** $n - 1$

5 **do** ordenar lista $B[i]$ com ordenação por inserção

6 concatenar as listas $B[0], B[1], \dots, B[n - 1]$ juntas em ordem

- Todas as linhas, exceto a linha 5 demoram $O(n)$ no pior caso.
- O que é necessário para que a linha 5 não seja quadrática?
(ver demonstração matemática no Cormen)

Exercício

- 1) Pesquisar outro algoritmo de ordenação em memória primária que não foi abordado nas aulas e descrevê-lo (ideia/análise complexidade/código).

Ex: mergesort, combsort, gnomesort, cocktailsort, timsort, etc.

- 2) Resolver lista de exercício sobre ordenação.

**O trabalho espanta três males: o vício, a
pobreza e o tédio.**

Voltaire

