

# Estrutura de Dados Avançada

Daniel de Sousa Moraes  
danielmoraes14@gmail.com

# Radix Sort

- Ordena os elementos processando dígito a dígito (bit a bit)
- Supondo que os elementos são definidos em um sistema numérico e com uma certa quantidade de dígitos
- Ordena em múltiplas passadas
  - Distribui cada elemento em listas (compartimentos) de acordo com o valor do dígito verificado
- Os elementos processado são frequentemente chamados de “chaves”, e podem ser strings de caracteres ou números

# Radix Sort

- Pode ser usado para ordenar a partir do dígito menos significativo (LSD) ou do dígito mais significativo (MSD)
- No LSD, chaves curtas vem antes das chaves longas e chaves de mesmo tamanho são ordenadas lexicograficamente.
- O MSD é o inverso

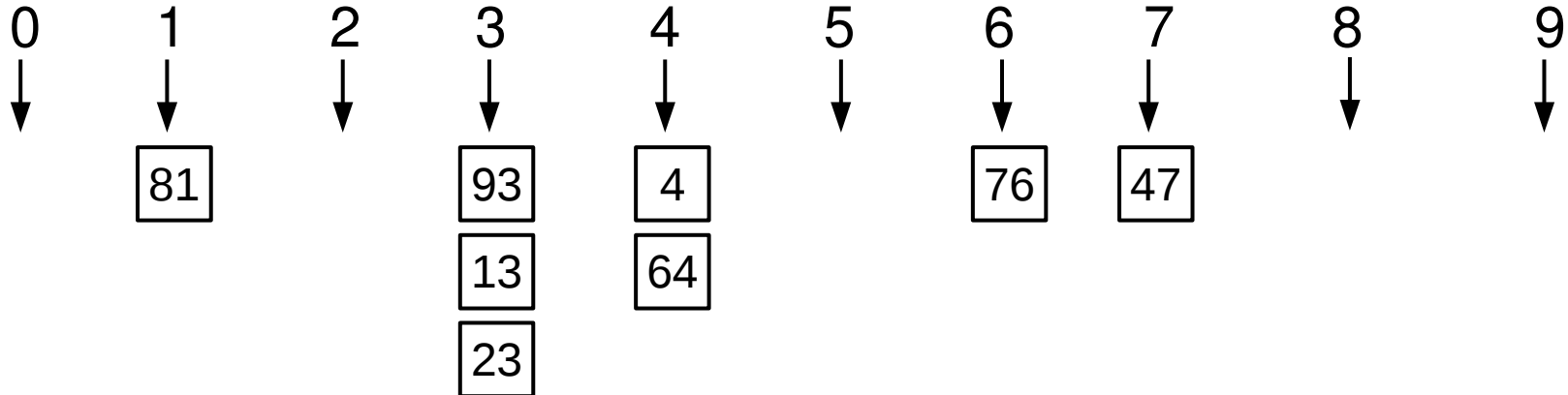
# Radix Sort

```
radixSort(V, d)  
  for  $i = 1$  to  $d$   
    ordenar  $V$  sobre o dígito  $i$ 
```

# Exemplo Radix Sort

- Para inteiros, o número de compartimentos (listas) é 10, 0 a 9.

9 <u>3</u>	1 <u>3</u>	4 <u>7</u>	<u>4</u>	7 <u>6</u>	2 <u>3</u>	6 <u>4</u>	8 <u>1</u>
------------	------------	------------	----------	------------	------------	------------	------------



# Radix Sort – LSD

- 134
- 286
- 130
- 467
- 981
- 723
- 832
- 67

# Radix Sort – LSD



- 13<sup>4</sup>
- 28<sup>6</sup>
- 13<sup>0</sup>
- 46<sup>8</sup>
- 98<sup>1</sup>
- 72<sup>3</sup>
- 83<sup>2</sup>
- 6<sup>7</sup>
- 130
- 981
- 832
- 723
- 134
- 286
- 67
- 468

# Radix Sort – LSD



- |       |       |       |
|-------|-------|-------|
| • 134 | • 130 | • 723 |
| • 286 | • 981 | • 130 |
| • 130 | • 832 | • 832 |
| • 468 | • 723 | • 134 |
| • 981 | • 134 | • 67  |
| • 723 | • 286 | • 468 |
| • 832 | • 67  | • 981 |
| • 67  | • 468 | • 286 |



# Radix Sort – LSD



- |       |       |               |       |
|-------|-------|---------------|-------|
| • 134 | • 130 | • <b>7</b> 23 | • 67  |
| • 286 | • 981 | • <b>1</b> 30 | • 130 |
| • 130 | • 832 | • <b>8</b> 32 | • 134 |
| • 468 | • 723 | • <b>1</b> 34 | • 286 |
| • 981 | • 134 | • <b>0</b> 67 | • 468 |
| • 723 | • 286 | • <b>4</b> 68 | • 723 |
| • 832 | • 67  | • <b>9</b> 81 | • 832 |
| • 67  | • 468 | • <b>2</b> 86 | • 981 |

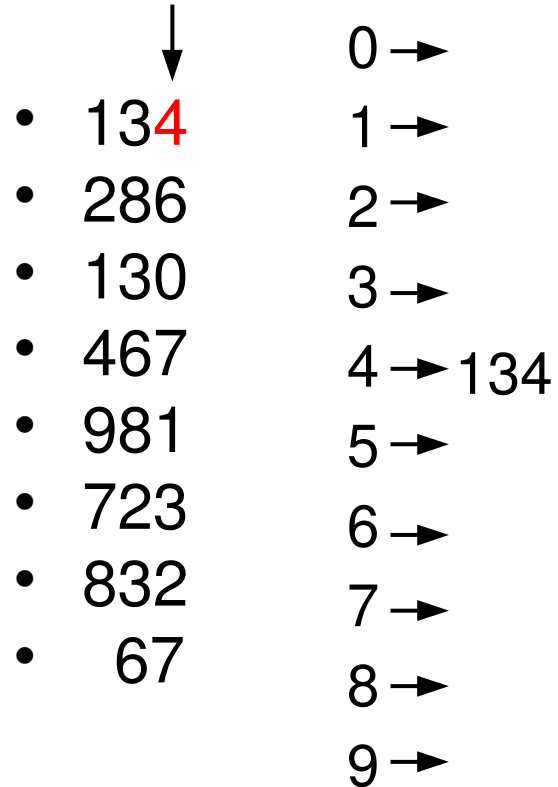
# Bucket Sort

- 134
  - 286
  - 130
  - 467
  - 981
  - 723
  - 832
  - 67
- Qual o intervalo das chaves? (quantidade de compartimentos) – Inteiros

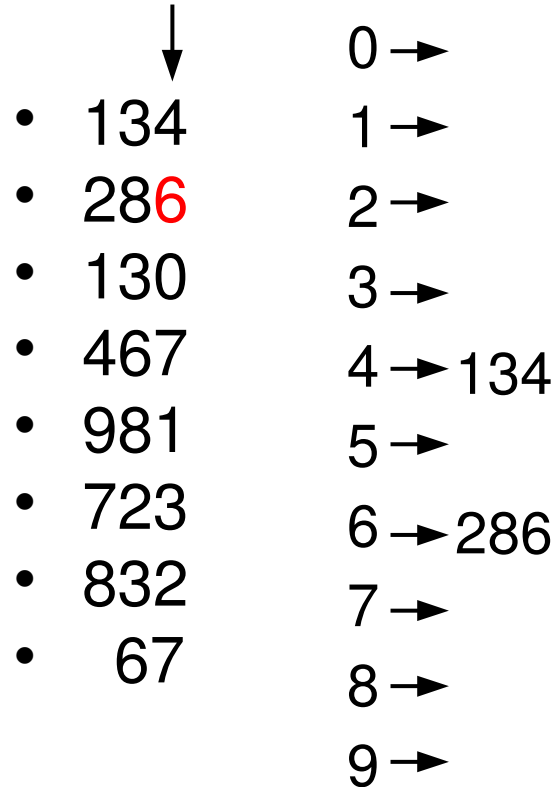
# Bucket Sort

- 134
  - 286
  - 130
  - 467
  - 981
  - 723
  - 832
  - 67
- 0 →
- 1 →
- 2 →
- 3 →
- 4 →
- 5 →
- 6 →
- 7 →
- 8 →
- 9 →

# Bucket Sort



# Bucket Sort



# Bucket Sort

	↓	0 → 130
• 134		1 →
• 286		2 →
• 130		3 →
• 467		4 → 134
• 981		5 →
• 723		6 → 286
• 832		7 →
• 67		8 →
		9 →

# Bucket Sort

	↓	0 → 130
• 134		1 →
• 286		2 →
• 130		3 →
• 467		4 → 134
• 981		5 →
• 723		6 → 286
• 832		7 → 467
• 67		8 →
		9 →

# Bucket Sort

	↓	0 → 130
• 134		1 → 981
• 286		2 →
• 130		3 →
• 467		4 → 134
• 981		5 →
• 723		6 → 286
• 832		7 → 467
• 67		8 →
		9 →



# Bucket Sort

↓	0 → 130
• 134	1 → 981
• 286	2 →
• 130	3 → 723
• 467	4 → 134
• 981	5 →
• 723	6 → 286
• 832	7 → 467
• 67	8 →
	9 →

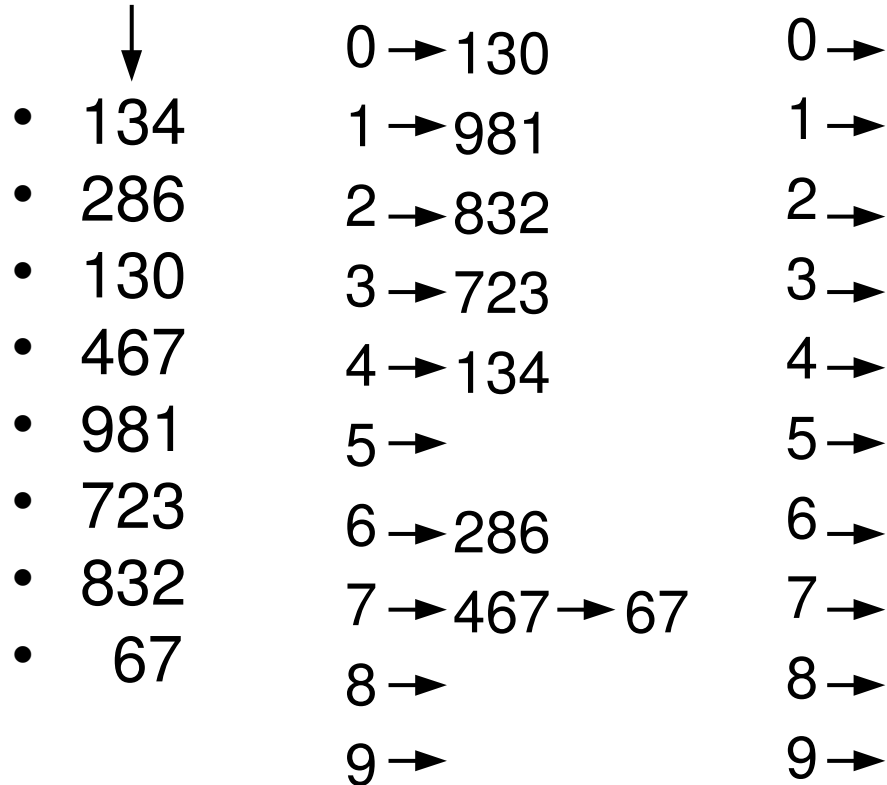
# Bucket Sort

↓	0 → 130
• 134	1 → 981
• 286	2 → 832
• 130	3 → 723
• 467	4 → 134
• 981	5 →
• 723	6 → 286
• 832	7 → 467
• 67	8 →
	9 →

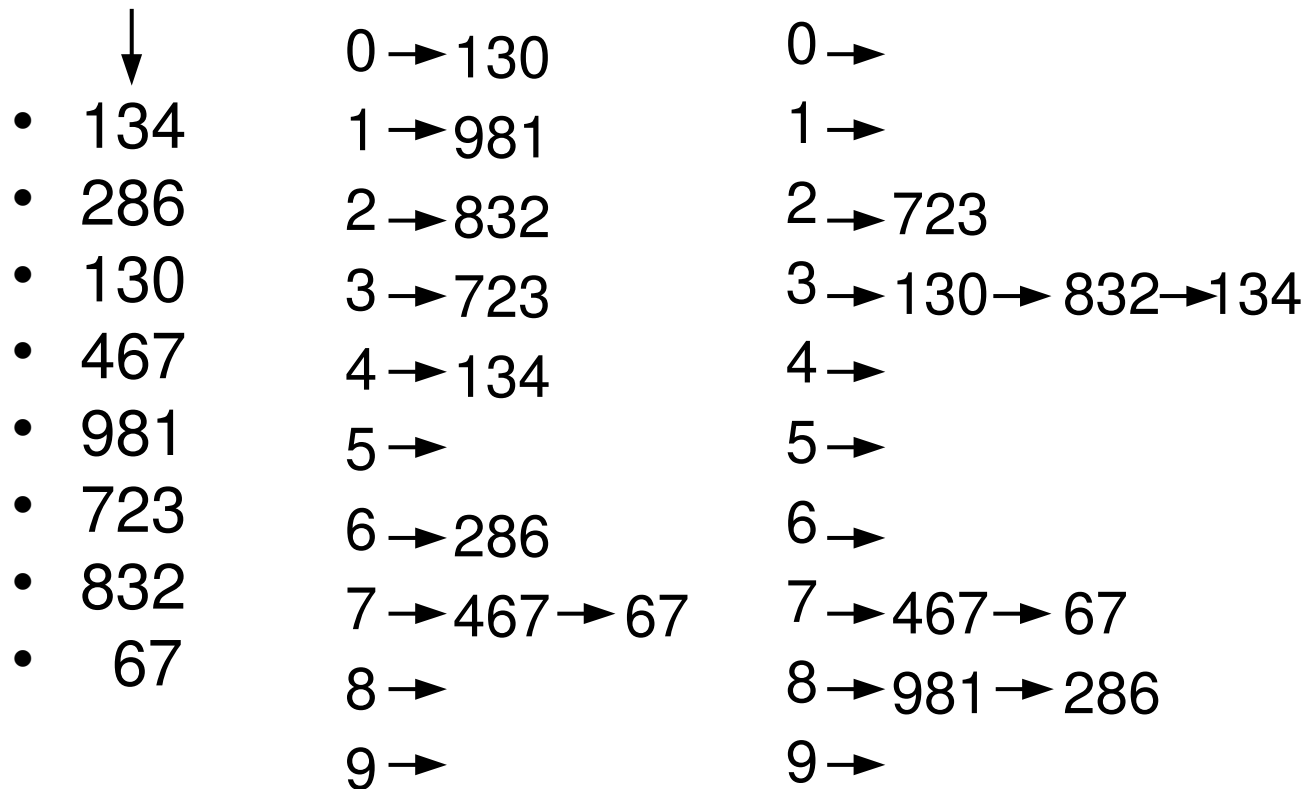
# Bucket Sort

↓	0 → 130
• 134	1 → 981
• 286	2 → 832
• 130	3 → 723
• 467	4 → 134
• 981	5 →
• 723	6 → 286
• 832	7 → 467 → 67
• 67	8 →
	9 →

# Bucket Sort



# Bucket Sort



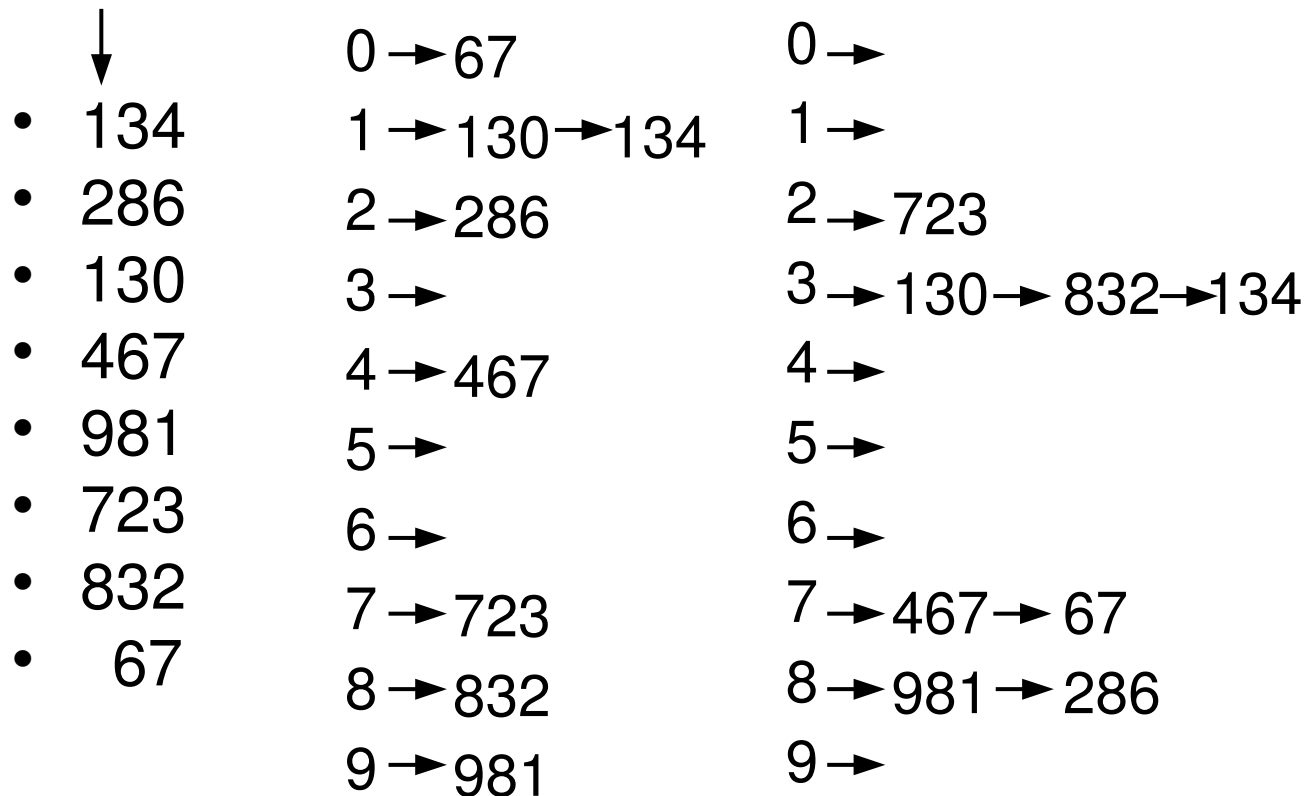
# Bucket Sort

- ↓
- 134
  - 286
  - 130
  - 467
  - 981
  - 723
  - 832
  - 67

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

0 →  
1 →  
2 → 723  
3 → 130 → 832 → 134  
4 →  
5 →  
6 →  
7 → 467 → 67  
8 → 981 → 286  
9 →

# Bucket Sort



# Bucket Sort

- ↓
- 134
  - 286
  - 130
  - 467
  - 981
  - 723
  - 832
  - 67
- 0 → 67  
1 → 130 → 134  
2 → 286  
3 →  
4 → 467  
5 →  
6 →  
7 → 723  
8 → 832  
9 → 981

67 130 134 286 467 723 832 981



# Radix Sort e Bucket Sort

- Usando o algoritmo estável (Counting Sort) para ordenar os números por cada bit, temos  **$O(d(n+k))$** , onde  $d$  é o número de dígitos e que o algoritmo estável leva  $O(n+k)$

# Bibliografia

Cormen, Thomas H. et al. Algoritmos.; [tradução Arlete Simille]. 3ª ed  
- Rio de Janeiro - Elsevier, 2011.

Carlos de Salles Soares Neto – Notas de Aula da Disciplina de ED II  
- UFMA