



Universidade Federal Rural de Pernambuco  
(UFRPE)  
- Unidade Acadêmica de Garanhuns-

## Algoritmos de Ordenação

### ***Merge Sort***

**Prof. Priscilla Kelly Machado Vieira**

# Apresentação do Capítulo

2

- *Merge Sort*
  - Características
  - Análise
- Exercícios

# Algoritmos de Ordenação

3

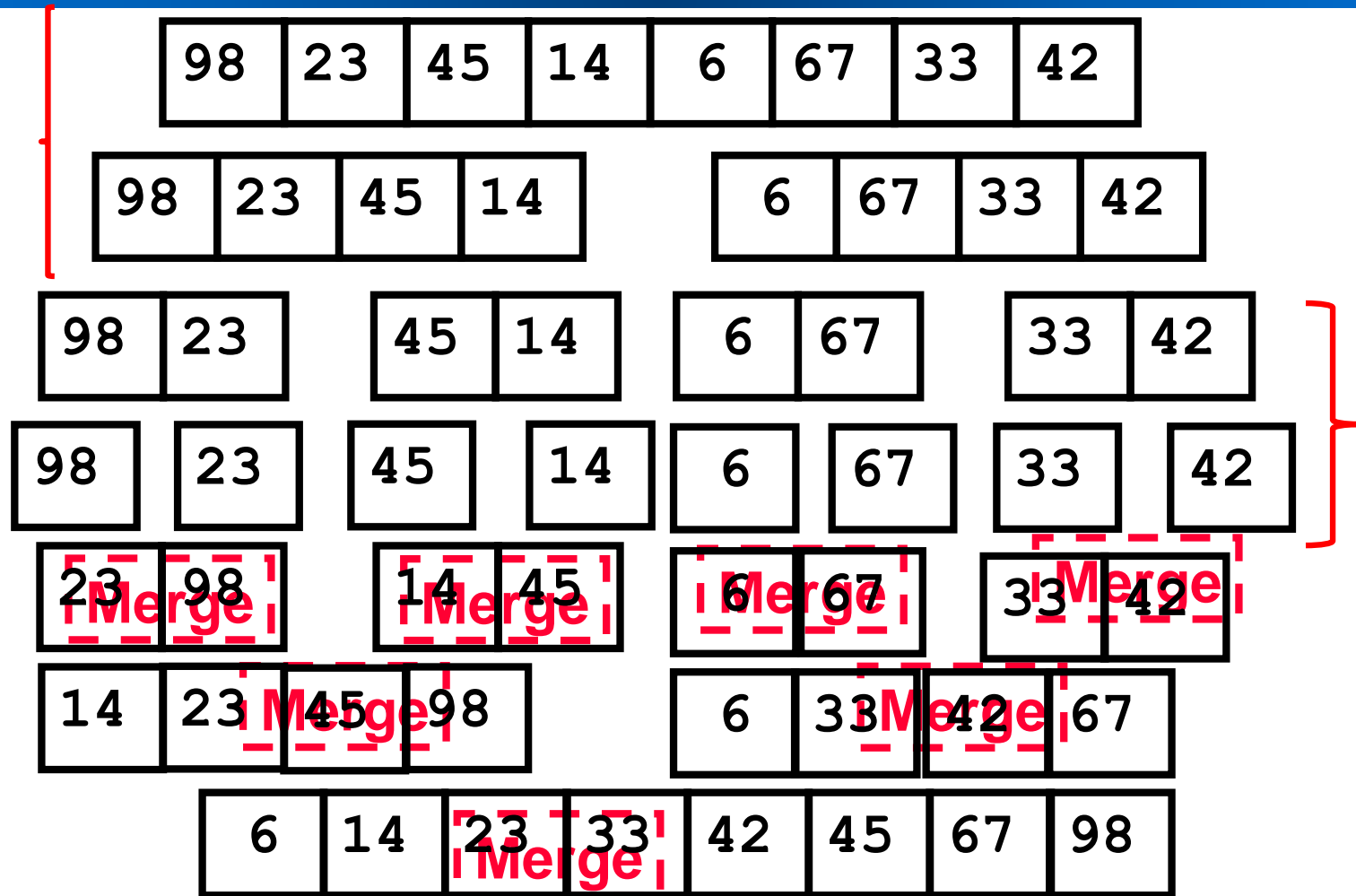
- Bubble Sort
- Selection Sort
- Insertion Sort
- Merge Sort
- Quick Sort

# Merge Sort

- Intuição
  - Algoritmo do tipo divisão e conquista
  - Dividir: se a sequência tiver mais de 1 elemento, divida em duas partes
  - Conquistar: ordene cada subsequência em separado usando merge-sort
  - Combinar: Junte as duas subsequências em uma sequência ordenada

# Merge Sort

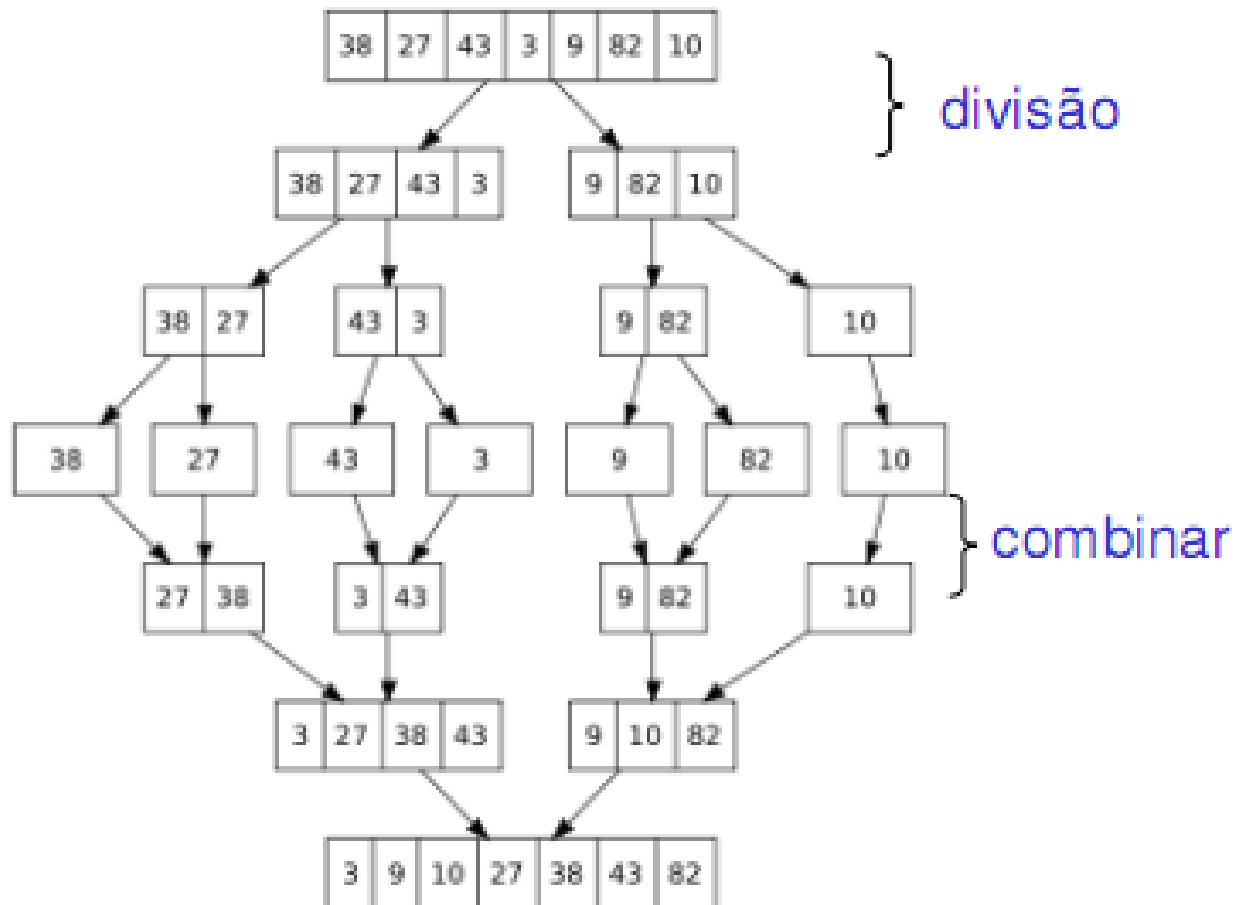
5



# Merge Sort

6

- Intuição



# Merge Sort

Número de elementos  
que serão intercalados



7

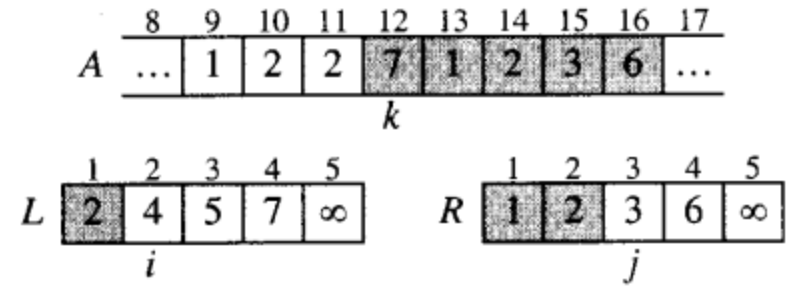
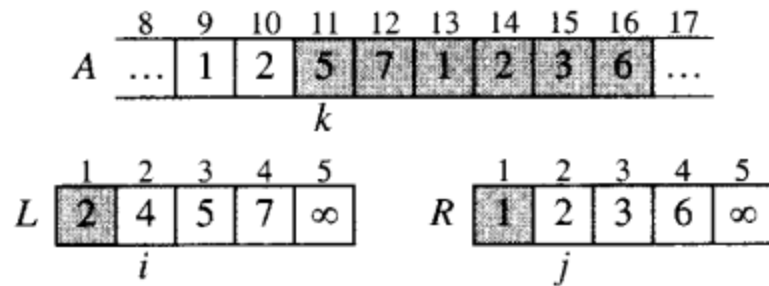
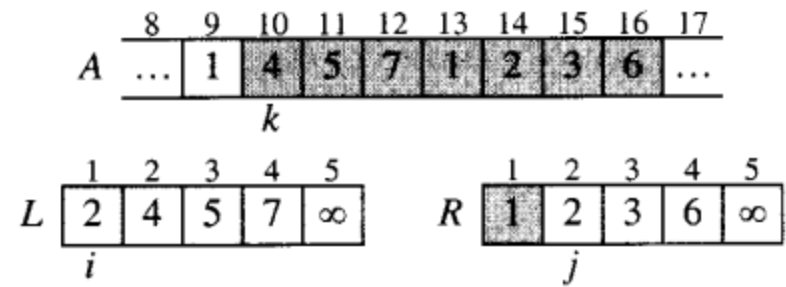
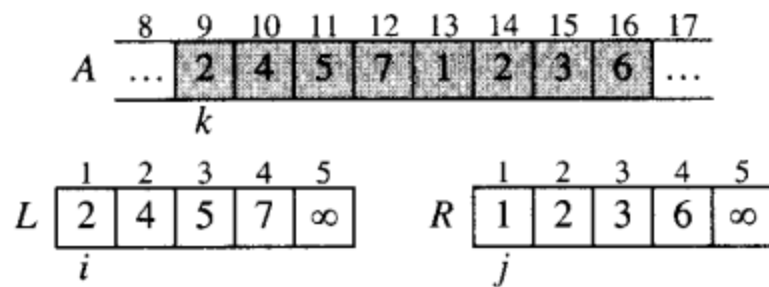
- Algoritmo

```
MERGE( $A, P, q, r$ )
1  $n_1 \leftarrow q - p + 1$ 
2  $n_2 \leftarrow r - q$ 
3 criar arranjos  $L[1..n_1 + 1]$  e  $R[1..n_2 + 1]$ 
4 for  $i \leftarrow 1$  to  $n_1$ 
5     do  $L[i] \leftarrow A[p + i - 1]$ 
6 for  $j \leftarrow 1$  to  $n_2$ 
7     do  $R[j] \leftarrow A[q + j]$ 
8  $L[n_1 + 1] \leftarrow \infty$ 
9  $R[n_2 + 1] \leftarrow \infty$ 
10  $i \leftarrow 1$ 
11  $j \leftarrow 1$ 
12 for  $k \leftarrow p$  to  $r$ 
13     do if  $L[i] \leq R[j]$ 
14         then  $A[k] \leftarrow L[i]$ 
15              $i \leftarrow i + 1$ 
16     else  $A[k] \leftarrow R[j]$ 
17          $j \leftarrow j + 1$ 
```

# Merge Sort

8

- Algoritmo





# Merge Sort

- Algoritmo

```
MERGE-SORT( $A, p, r$ )  
1  if  $p < r$   
2    then  $q \leftarrow \lfloor (p + r)/2 \rfloor$   
3        MERGE-SORT( $A, p, q$ )  
4        MERGE-SORT( $A, q + 1, r$ )  
5        MERGE( $A, p, q, r$ )
```

# Merge Sort

10

- Animação
  - <http://www.cse.iitk.ac.in/users/dsrkg/cs210/applets/sortingII/mergeSort/mergeSort.html>

# Características

- Boa performance
- Não é In place
  - Precisa de espaço extra de  $n$
- Existem algumas variações que podem melhorar
  - Complexidade aumenta
  - Não é stable
- Stable

# Laboratório 6

12

- Implemente em C o algoritmo do Merge Sort, junte-o aos demais algoritmos de ordenação e execute a avaliação de desempenho semelhante aos laboratórios anteriores
  - Os algoritmos de ordenação devem ser inseridos em um arquivo separado
  - Todos recebem como entrada um vetor que varia de tamanho (300, 3000 e 30000)
  - O tempo medido deve ser sobre a mesma entrada para todos os algoritmos