

**ESTRUTURA DE DADOS**  
**ALGORITMO DE ORDENAÇÃO POR INTECALAÇÃO**  
**(MERGESORT)**

**Prof. Enyo José**

# Mergesort

- Complexidade:
  - $n \log n$  (Melhor caso, Caso médio e Pior caso)
- Lógica:
  - Quebra/divide o vetor ao meio repetidas vezes até que o sub-vetor tenha tamanho 1
  - Em seguida desempilha as chamadas recursivas intercalando os elementos dos sub-vetores
    - Os sub-vetores chegam na intercalação parcialmente ordenados

Diferente do Quicksort que faz a ordenação parcial antes de fazer a chamada recursiva

# Mergesort - Funcionamento

- Duas funções:
  - Mergesort = responsável pelas chamadas recursivas e pela chamada a função que faz as ordenações parciais
  - Merge = Intercala 2 sub-vetores
- Função Mergesort
  - Calcula meio
  - Chamadas recursivas – quebra vetor ao meio
  - Chamada para o merge

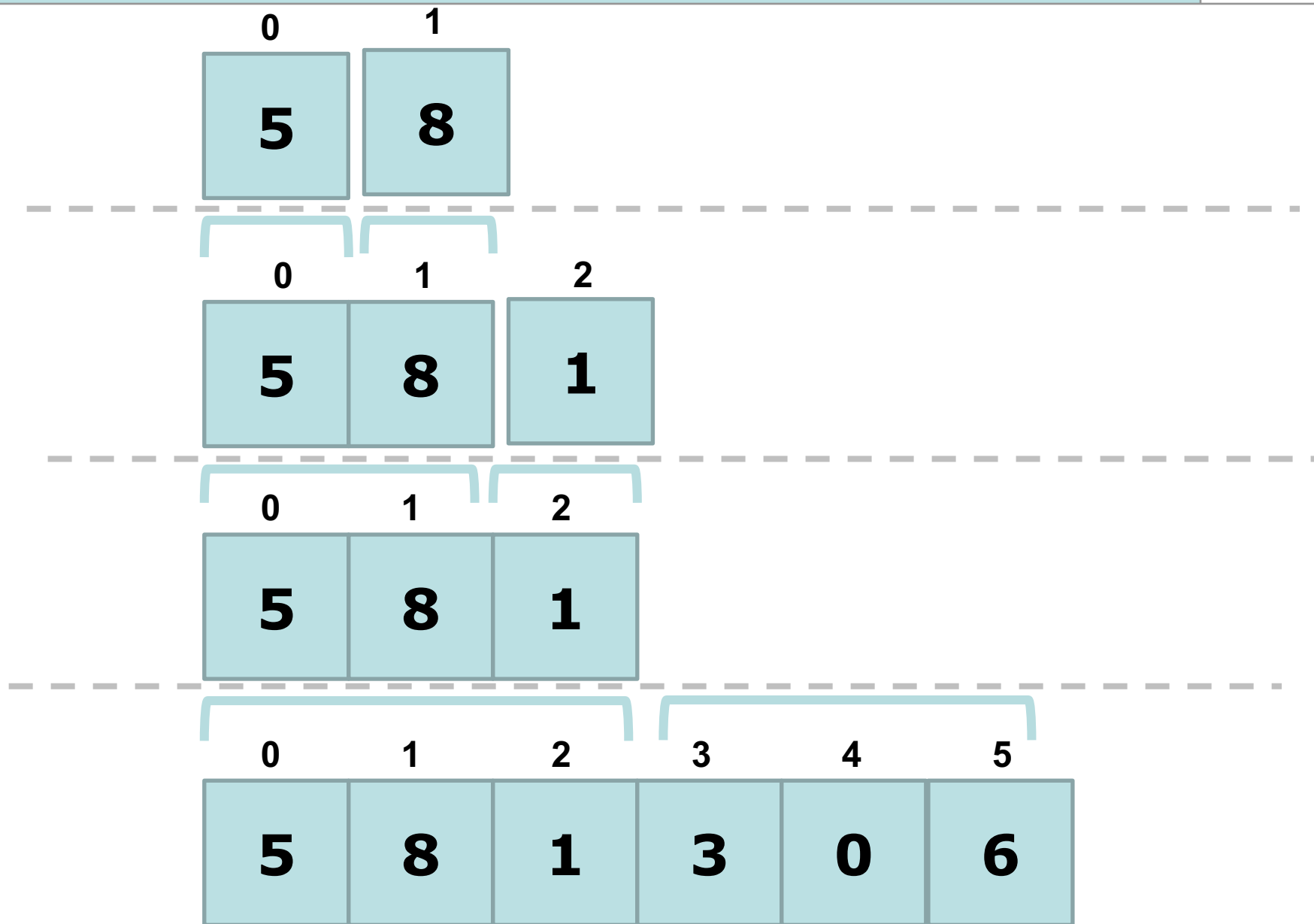
# Mergesort - Funcionamento

- Função Merge
  1. Definir dois vetores auxiliares para cada uma das partes do vetor base
    - A: meio - início + 1
    - B: fim - meio
  2. Preenche os vetores auxiliares
  3. Percorre o vetor base preenchendo-o com os elementos dos vetores auxiliares (intecalando)
  4. Ao final, coloca os elementos que sobraram nos sub-vetores

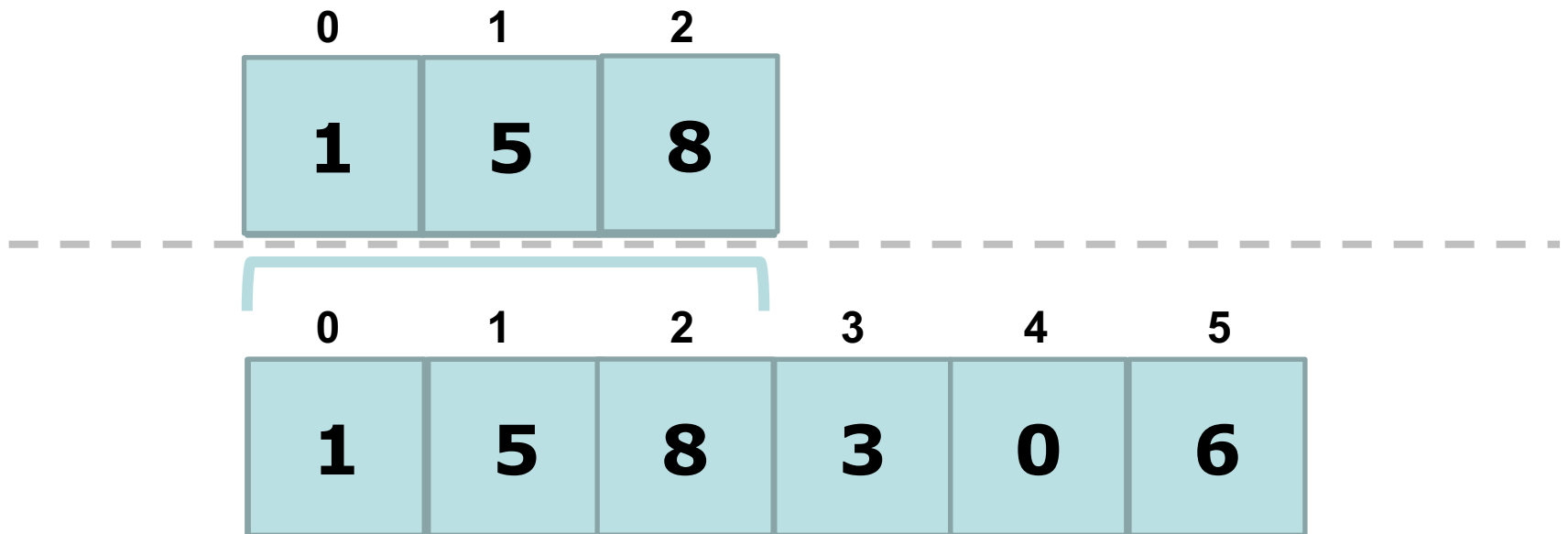
# Mergesort – Ilustração

- A ilustração dos próximos slides está disponível em <https://youtu.be/uC4HNZ8sj4M>

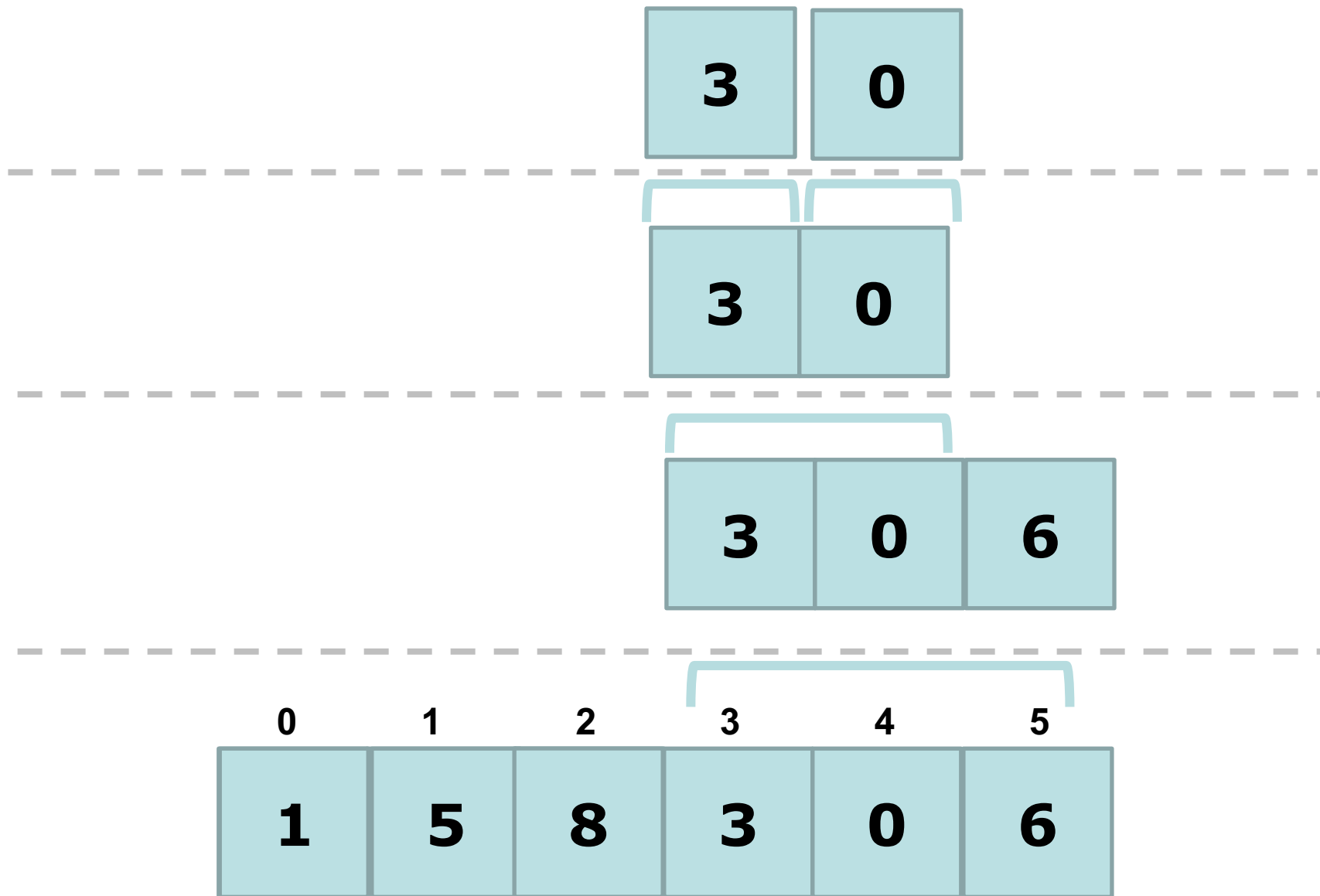
# Ilustração



# Ilustração



# Ilustração





# Ilustração

A **3**      B **0**

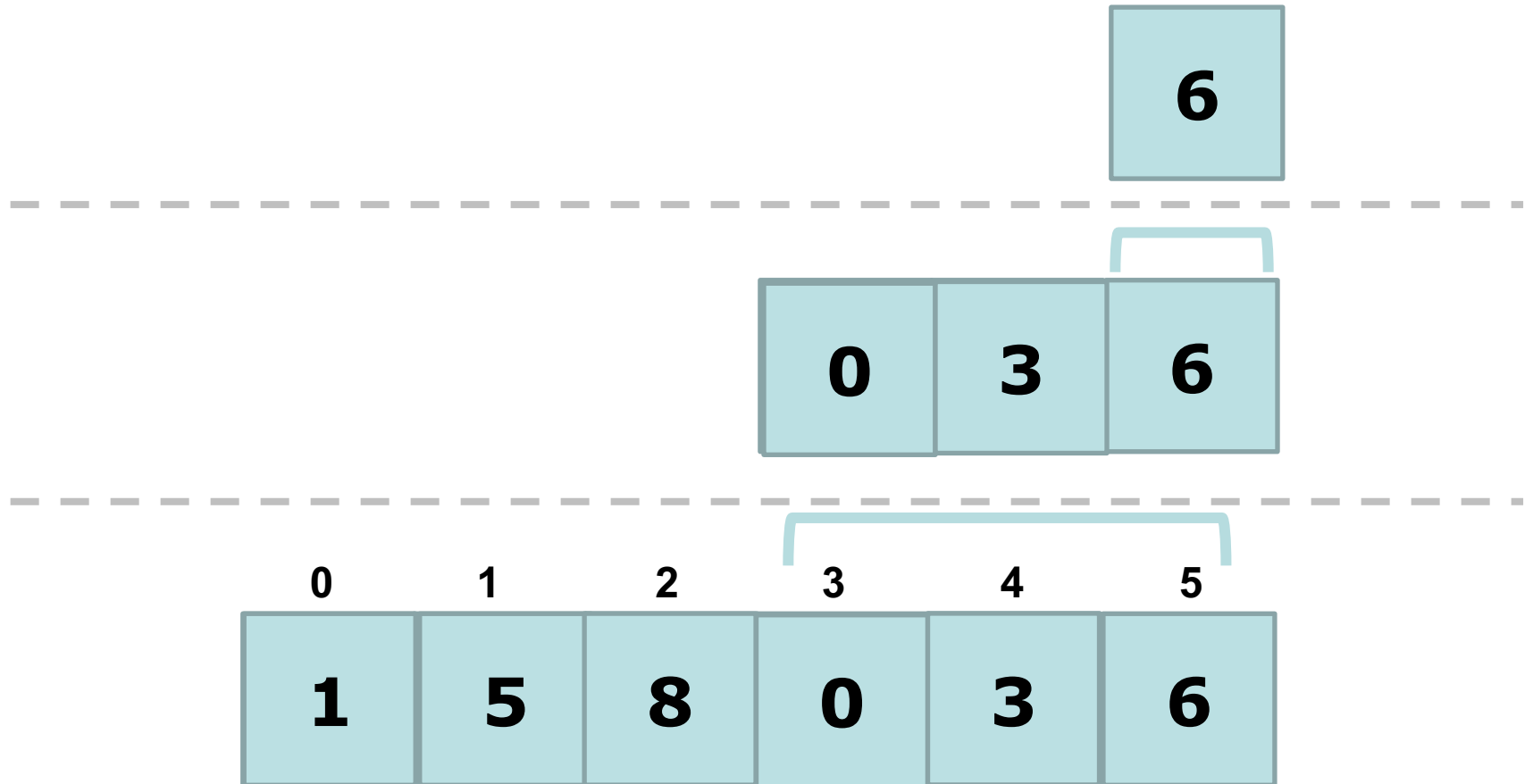
**0**   **3**

**0**   **3**   **6**

0      1      2      3      4      5

**1**   **5**   **8**   **0**   **3**   **6**

# Ilustração

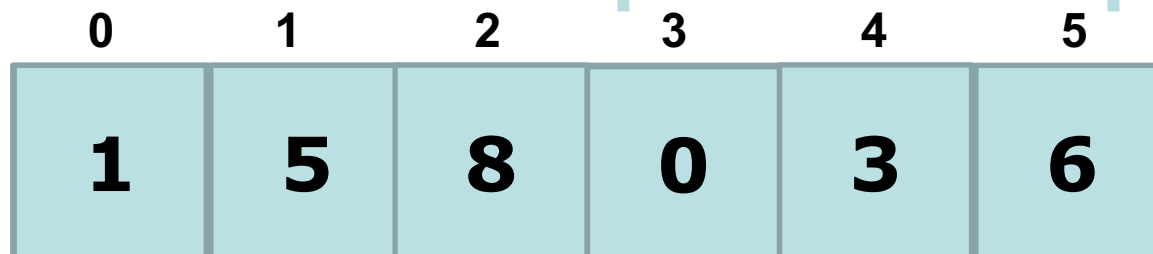
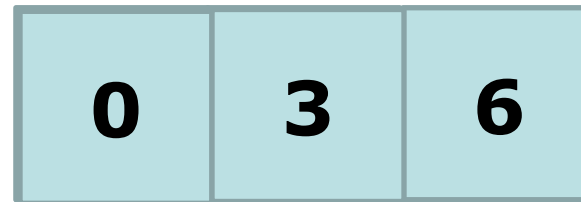


# Ilustração

**A**



**B**



# Ilustração

**A**

**1**

**5**

**8**

**B**

**0**

**3**

**6**

**0**

**1**

**2**

**3**

**4**

**5**

**0**

**1**

**3**

**5**

**6**

**8**

# Atividade

1. Desenhe o comportamento do Quicksort e do Mergesort para o seguinte vetor de entrada

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

1.

2. Implemente o Quicksort e do Mergesort para ordenação em ordem decrescente