

# Mergesort

Análise de Algoritmos – Ciência da Computação



Prof. Daniel Saad Nogueira  
Nunes

IFB – Instituto Federal de Brasília,  
Campus Taguatinga



# Sumário

---

1 Mergesort

2 Análise



# Sumário

---

## 1 Mergesort



# Mergesort

---

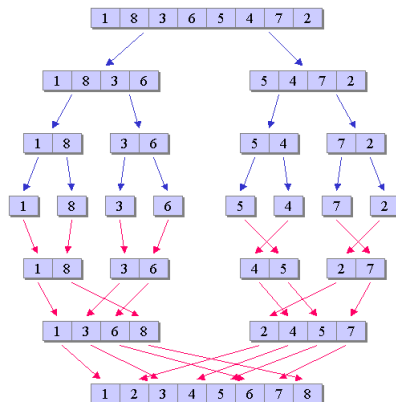
## Mergesort

- O Mergesort se baseia no conceito de Merge (junção) de duas sequências ordenadas. Primeiramente ele subdivide a sequência original na metade e ordena recursivamente essas sequências.
- Caso base: sequência unitária ou vazia, pois essas já são ordenadas.
- Por fim, faz a junção das duas sequências ordenadas para compor uma sequência maior ordenada.
- $(1, 3, 5, 7, 9) + (0, 2, 4, 6, 8) \xrightarrow{\text{merge}} (0, 1, 2, 3, 4, 5, 6, 7, 8, 9)$



# Mergesort

## Exemplo





# Mergesort

---

---

## Function Mergesort

---

**Input:**  $V[0, n - 1]$

**Output:**  $V[0, n - 1], \quad V[i] \leq V[i + 1], 0 \leq i < n - 1$

```
1 if(  $n > 1$  )
2    $V_1 \leftarrow V[0, \lfloor n/2 \rfloor]$ 
3    $V_2 \leftarrow V[\lfloor n/2 \rfloor + 1, n - 1]$ 
4   MERGESORT( $V_1$ )
5   MERGESORT( $V_2$ )
6   MERGE( $V, V_1, V_2$ )
```

---



# Mergesort

---

---

## Function Merge

---

**Input:**  $V[0, n - 1], V_1[0, \lfloor n/2 \rfloor], V_2[0, n - \lfloor n/2 \rfloor - 1]$

**Output:**  $V[0, n - 1], \quad V[i] \leq V[i + 1], 0 \leq i < n - 1$

```
1 for(  $i \leftarrow 0, j \leftarrow 0, k \leftarrow 0; j < V_1.SIZE() \wedge k < V_2.SIZE(); i++$  )
2   |   if(  $V_1[j] \leq V_2[k]$  )
3     |    $V[i] \leftarrow V_1[j++]$ 
4   |   else
5     |    $V[i] \leftarrow V_2[k++]$ 
6 while  $j < V_1.SIZE()$  do  $V[i++] \leftarrow V_1[j++]$ 
7 while  $k < V_2.SIZE()$  do  $V[i++] \leftarrow V_2[k++]$ 
```

---



# Sumário

---

## 2 **Análise**





# Mergesort

---

## Análise

A relação de recorrência do Mergesort corresponde à:

$$T(n) = 2 \cdot T(n/2) + O(n) \in \Theta(n \lg n)$$

In-place	Estável
✗	✓

## Observação

- Requer uma quantidade de memória superior a  $O(1)$  (vetores auxiliares).
- Recursivo!