



GRUPO 2

# *MergeSort*

Cefas Daniel Pereira Corrêa, Gabriel Bruno Oliveira Pereira, Gustavo Raasch,  
João Vitor Dellarmelina, Maria Eduarda Fiorio, Pedro Henrique Ferreira, Stefano  
Silvestri



# *Lógica de Funcionamento*

**MergeSort** é um algoritmo que segue a lógica de **dividir para conquistar**.

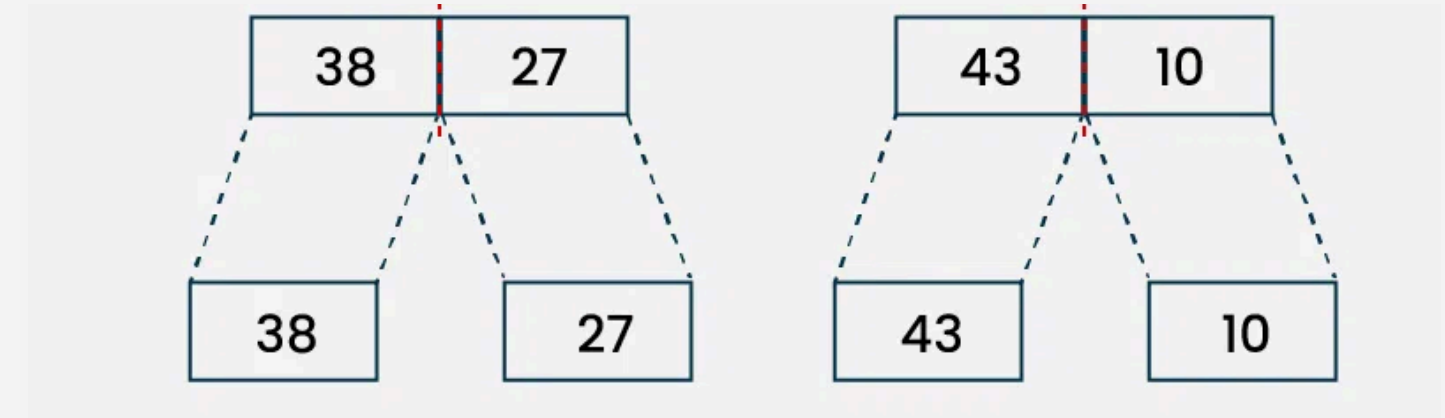
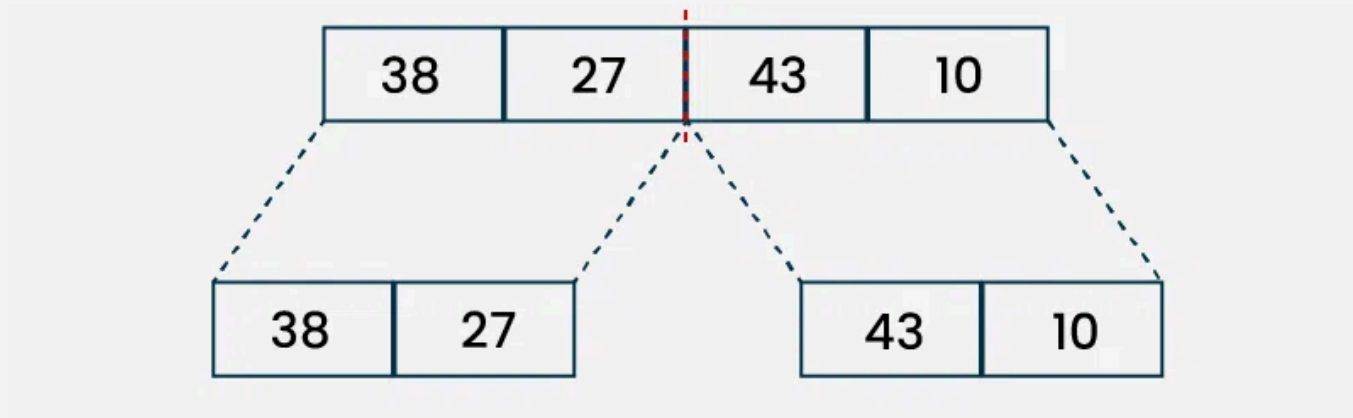
O funcionamento consiste em dividir o array em duas metades, classificar cada metade e, então, mesclar as metades classificadas novamente, até que o array inteiro esteja classificado.

# *Lógica de Funcionamento*

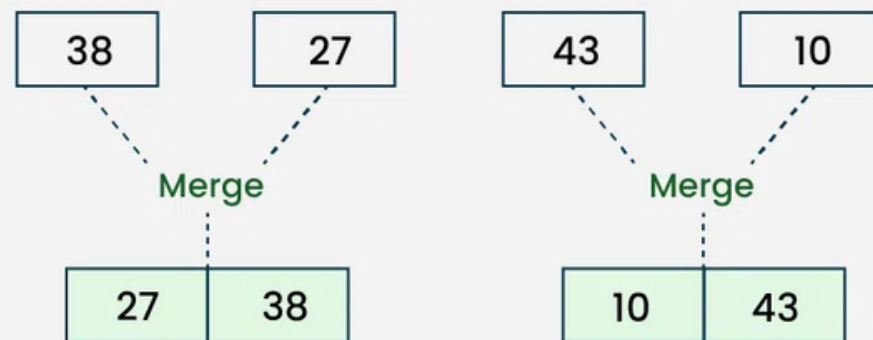
O método de divisão e conquista consiste em 3 passos:

1. **Dividir:** Dividir o problema original em subproblemas menores.
2. **Conquistar:** Resolver cada subproblema recursivamente.
3. **Mesclar:** Combinar as soluções encontradas, compondo uma solução para o problema original.

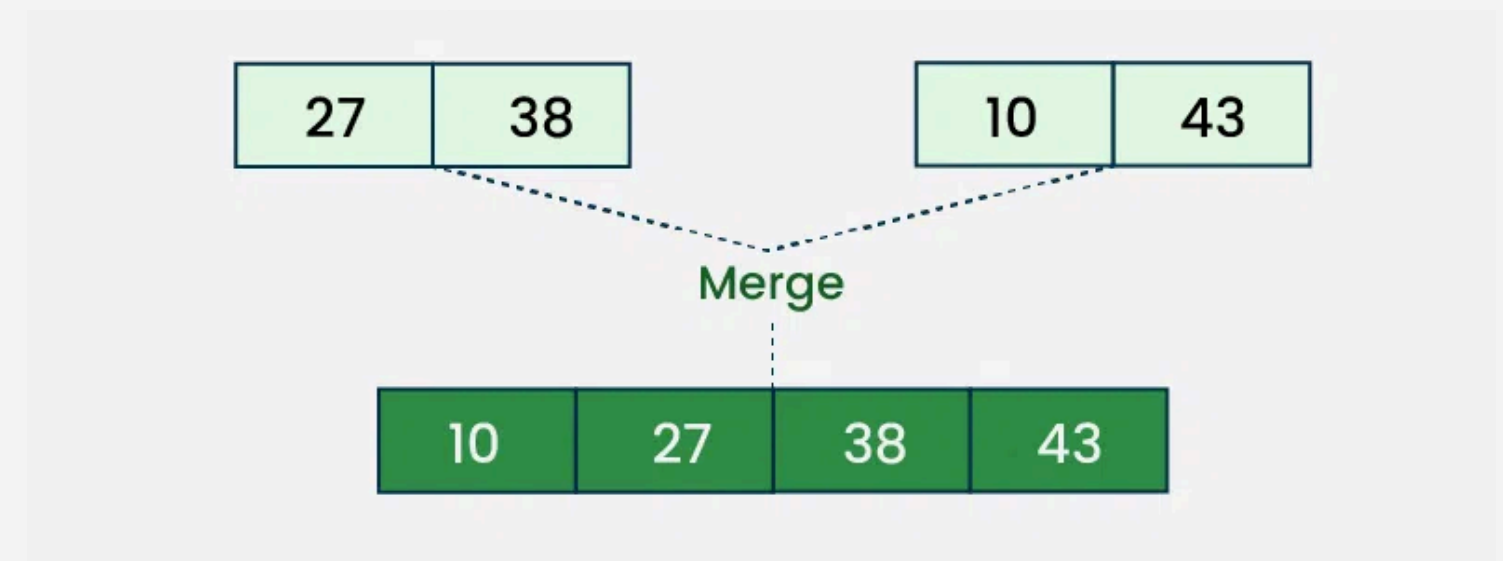
1.Dividir o array em duas metades, e os subarrays até que não possam ser mais divididos



2.Ordena as partes recursivamente



3. Mescla os subarrays ordenados em um array ordenado



# *Análise de Complexidade*

## **Complexidade de tempo:**

O Merge Sort tem uma complexidade de tempo igual em todos os casos, devido ao padrão de divisão e mesclagem do algoritmo

### **Melhor caso:**

$O(n \log n)$

### **Médio caso:**

$O(n \log n)$

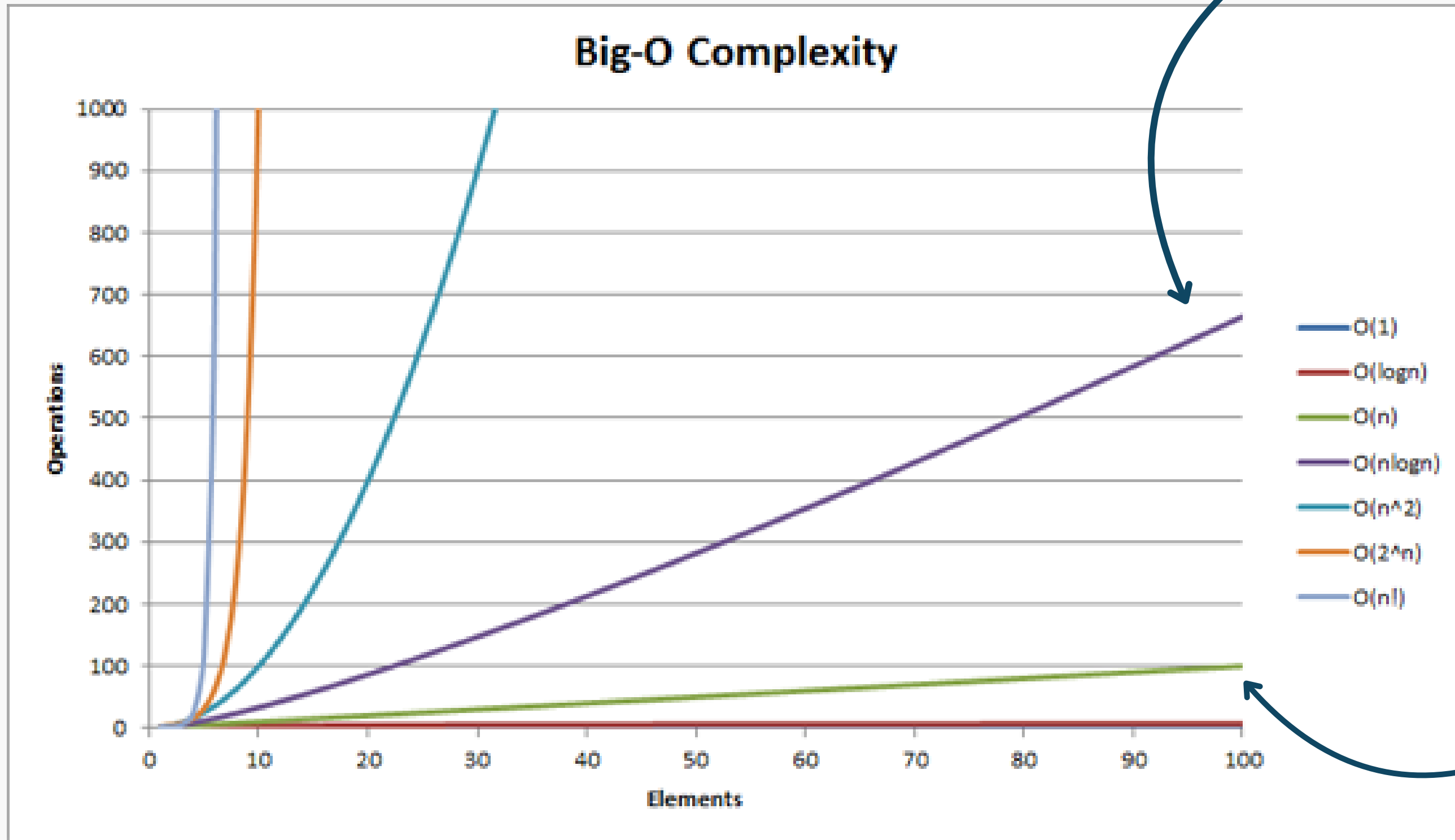
### **Pior caso:**

$O(n \log n)$

## **Complexidade de espaço: $O(n)$**

O algoritmo precisa de espaço adicional para armazenar temporariamente as sublistas durante a fase de mesclagem

# Análise de Complexidade



## Complexidade de tempo

## Complexidade de espaço

## Gráfico de análise de complexidade de tempo

# *Vantagens*

**Estabilidade:** O algoritmo é estável, ou seja, preserva a ordem dos elementos iguais.

**Desempenho garantido no pior caso:** tem uma complexidade de tempo no pior caso de  $O(N \log N)$ , o que significa que ela tem um bom desempenho mesmo em grandes conjuntos de dados.

**Altamente paralelizáveis:** mesclamos submatrizes de forma independente, o que as torna adequadas para processamento paralelo.

# *Desvantagens*

**Uso de memória:** requer memória adicional para armazenar as submatrizes mescladas durante o processo de classificação.

**Complexidade de Implementação:** A implementação do Merge Sort pode ser mais complexa do que outros algoritmos simples de ordenação, como o Quick Sort ou o Bubble Sort.





GRUPO 2

*Obrigado!*

