

Mergesort: Estratégia e Análise

1. Introdução

O **Mergesort** é um algoritmo de ordenação baseado na estratégia **dividir para conquistar**. Criado por **John von Neumann em 1945**, ele divide recursivamente a lista em partes menores até obter sublistas unitárias e, em seguida, as combina de forma ordenada. É um algoritmo **estável**, mantendo a ordem relativa de elementos iguais, e possui complexidade **$O(n \log n)$** em todos os casos.

2. Funcionamento

O algoritmo segue três etapas principais:

- **Dividir:** a lista é separada em duas metades até alcançar sublistas unitárias.
- **Conquistar:** cada sublista é considerada ordenada.
- **Combinar:** as sublistas ordenadas são mescladas comparando elementos sucessivos.

A fusão ocorre utilizando ponteiros que percorrem ambas as sublistas, sempre inserindo o menor elemento na lista final.

3. Exemplo de Execução

Dada a lista [7, 5, 1, 8, 4, 9], o Mergesort a divide até obter [1], [5], [7], etc., e então as combina ordenadamente até resultar em [1, 4, 5, 7, 8, 9].

Este método também pode ordenar **strings**, como ["zebra", "macaco", "elefante", "banana"], retornando ["banana", "elefante", "macaco", "zebra"].

4. Comparativo com Outros Algoritmos

Algoritmo	Melhor Caso	Caso Médio	Pior Caso	Espaço Extra	Estável?
Mergesort	$O(n \log n)$	$O(n \log n)$	$O(n \log n)$	$O(n)$	✓ Sim
Bubblesort	$O(n)$	$O(n^2)$	$O(n^2)$	$O(1)$	✓ Sim
Insertionsort	$O(n)$	$O(n^2)$	$O(n^2)$	$O(1)$	✓ Sim
Quicksort	$O(n \log n)$	$O(n \log n)$	$O(n^2)$	$O(\log n)$	✗ Não

O **Mergesort** é mais previsível que o **Quicksort**, pois sempre tem complexidade **$O(n \log n)$** , independentemente do caso. No entanto, requer **espaço extra $O(n)$** , enquanto o **Quicksort** pode ser implementado in-place com **$O(\log n)$** de espaço adicional.

5. Complexidade

- **Tempo:** $O(n \log n)$ no **melhor, médio e pior caso**.
- **Espaço:** $O(n)$, devido à necessidade de armazenamento auxiliar para a fusão.

Apesar do consumo extra de memória, sua **estabilidade e eficiência garantida** tornam o Mergesort uma escolha ideal para ordenação de grandes volumes de dados e para casos onde a preservação da ordem relativa é essencial.