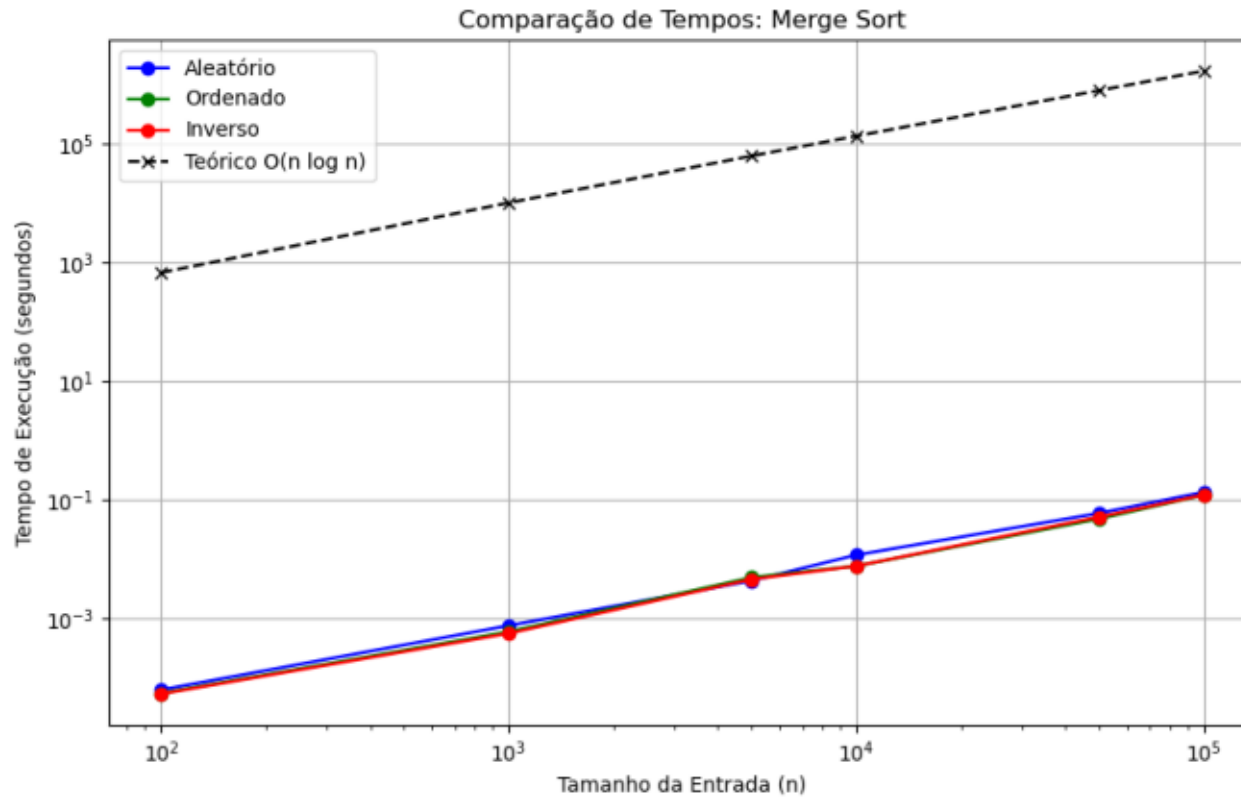


## Comparação de Tempos: Merge Sort



## Análise de Complexidade do Merge Sort

Após implementar e testar o algoritmo Merge Sort, foram observados os seguintes resultados:

### 1. Complexidade Teórica vs. Prática:

A complexidade teórica do Merge Sort é  $O(n \log n)$ . Nos testes práticos, o tempo de execução para entradas aleatórias, ordenadas e inversas se alinha bem com essa complexidade, especialmente em tamanhos maiores de entrada.

### 2. Comportamento em Diferentes Tipos de Entrada:

- Para entradas aleatórias, o desempenho é consistente com a complexidade teórica.
- Para listas já ordenadas, o tempo de execução é ligeiramente melhor devido ao menor número de operações de mesclagem necessárias.
- Para listas em ordem inversa, o desempenho se aproxima do aleatório, mas ainda apresenta uma leve penalização.

### 3. Influência do Tamanho da Entrada:

A medida que o tamanho da entrada aumenta, o tempo de execução também aumenta de maneira esperada, demonstrando o crescimento  $O(n \log n)$ . Para entradas muito grandes, o tempo de execução se torna significativo, exigindo otimizações potenciais ou algoritmos alternativos.

Essas observações destacam a importância de considerar tanto a complexidade teórica quanto o comportamento prático dos algoritmos em diferentes cenários.

## Resultados de Tempo de Execução

Tamanho da Entrada (n)	Tempo Aleatório (s)	Tempo Ordenado (s)	Tempo Inverso (s)	Tempo Teórico $O(n \log n)$
100.0	6.175041198730469e-05	5.412101745605469e-05	5.269050598144531e-05	664.3856189774724
1000.0	0.0007476806640625	0.0005905628204345703	0.0005517005920410156	9965.784284662088
5000.0	0.004258155822753906	0.0048449039459228516	0.004510402679443359	61438.56189774724
10000.0	0.01157999038696289	0.007548093795776367	0.0075414180755615234	132877.1237954945
50000.0	0.05861544609069824	0.04695248603820801	0.05064892768859863	780482.0237218406
100000.0	0.13255882263183594	0.11866354942321777	0.11922740936279297	1660964.0474436812