

## Comparação Teórica vs. Prática

O algoritmo Merge Sort tem uma **complexidade de tempo garantida de  $O(n \log n)$** , tanto no melhor, pior, quanto no caso médio. Isso significa que, teoricamente, o número de operações realizadas pelo algoritmo em todos esses cenários será aproximadamente o mesmo.

- **Melhor caso:** O array já está ordenado. O Merge Sort ainda faz comparações e divide o array, então a complexidade teórica continua sendo  $O(n \log n)$ .
- **Caso médio:** O array é aleatório. Esse caso reflete o comportamento típico do algoritmo. O número de divisões se mantém o mesmo
- **Pior caso:** O array está ordenado de forma inversa. O Merge Sort continua com a mesma complexidade  $O(n \log n)$ , com o mesmo número de divisões.

Portanto, na teoria, os tempos de execução para qualquer um dos casos deve ser o mesmo.

Abaixo estão os resultados de testes feitos com 1000 entradas em cada um dos casos:

Caso	Número de entradas	Tempo de execução
Melhor	1000	0.000993s
Médio	1000	0.001000s
Pior	1000	0.001053s

## 5. Conclusões

É possível perceber, que na prática existe uma diferença mínima entre os tempos de execução, que pode ser causada por diversos fatores, como gestão de memória, cache, cpu e sistema operacional.

```
Melhor Caso (ordenado):  
Tempo de execução: 0.000993 segundos
```

```
Caso Médio (aleatório):  
Tempo de execução: 0.001000 segundos
```

```
Pior Caso (decrescente):  
Tempo de execução: 0.001053 segundos
```