

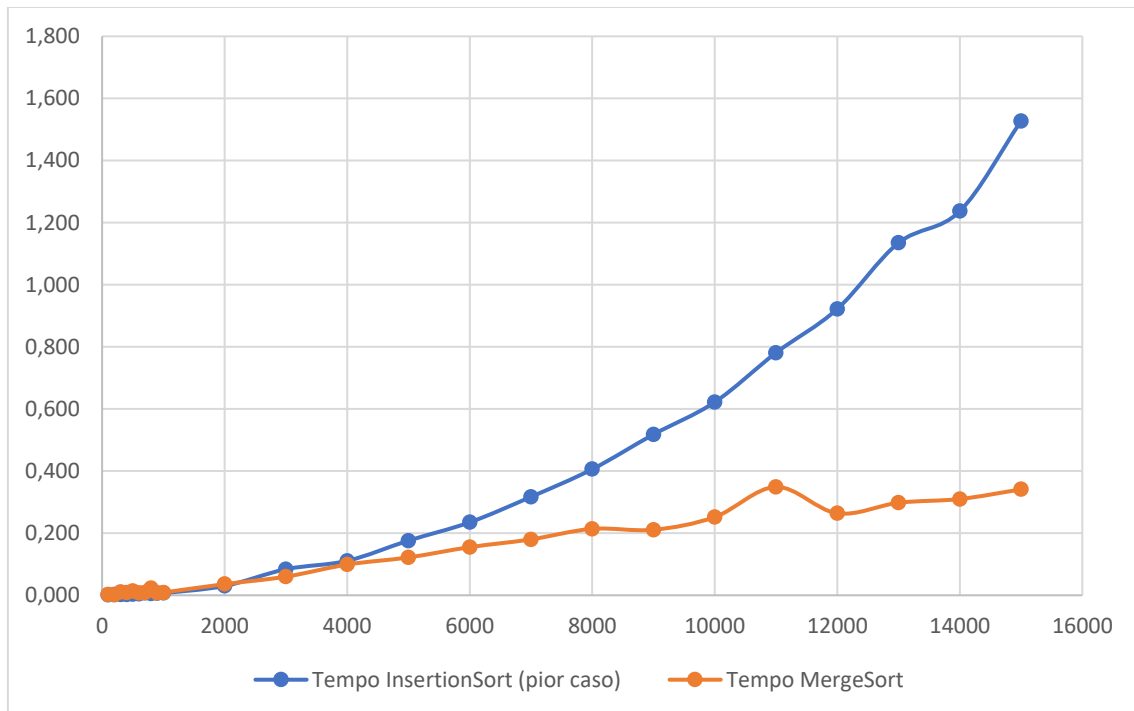
## INTRODUÇÃO

Neste relatório iremos fazer uma breve análise sobre dois algoritmos de ordenação vistos no decorrer do curso Introdução à Análise de Algoritmos: *Insertion Sort* e *Merge Sort*. Após testarmos o tempo de processamento para entradas de tamanhos diferentes, segue abaixo uma tabela na qual consta a comparação dos tempos obtidos para cada um dos algoritmos:

**Tabela 1: comparação dos tempos entre os dois algoritmos (tempo em segundos)**

Quantidade de valores	Tempo InsertionSort (pior caso)	Tempo MergeSort
100	0,001	0,002
200	0,002	0,001
300	0,002	0,011
400	0,002	0,009
500	0,003	0,014
600	0,004	0,007
700	0,007	0,009
800	0,005	0,023
900	0,006	0,007
1000	0,007	0,009
2000	0,030	0,036
3000	0,084	0,060
4000	0,111	0,099
5000	0,176	0,122
6000	0,235	0,155
7000	0,317	0,180
8000	0,406	0,214
9000	0,518	0,211
10000	0,622	0,252
11000	0,781	0,349
12000	0,922	0,264
13000	1,135	0,298
14000	1,237	0,310
15000	1,527	0,341

**Gráfico 1: comparação dos tempos entre os dois algoritmos (eixo vertical com o tempo em segundos e o eixo horizontal com a quantidade de valores do vetor)**



Seguem abaixo os modelos teóricos do tempo de processamento de pior caso de cada um dos algoritmos:

- 1) Insertion Sort:** para o *Insertion Sort*, o tempo de processamento no pior caso, ou seja, quando os valores estão em ordem decrescente, é  $\theta(n^2)$ ;
- 2) Merge Sort:** para o *Merge Sort*, o tempo de processamento no pior caso é  $\theta(n \log(n))$ .

Obs.: o algoritmo *Merge Sort* não tem pior caso, já que qualquer caso leva um tempo  $\theta(n \log(n))$  de processamento.

## ANÁLISE

De acordo com os dados obtidos e registrados na tabela 1 e no gráfico 1, podemos concluir que o algoritmo de ordenação mais eficiente é o *Merge Sort*, o que corresponde à comparação dos modelos teóricos expostos acima.