Tabela Comparativa

Utilizando a notação Big-O

|  |  |
| --- | --- |
| Algoritmo | Complexidade no pior caso |
| Bubble Sort | O (n2) |
| Selection Sort | O (n2) |
| Merge Sort | O (n log n) |
| Quick Sort | O (n2) |

|  |
| --- |
|  |

Resumo das Complexidades

1. Bubble Sort: Ele percorre o vetor diversas vezes, fazendo o maior elemento "flutuar" para o final. Tem O (n2) de comparações no pior caso (lista invertida).
2. Selection Sort: Seleciona o menor valor para troca, caso o primeiro valor seja maior que ele. O número de comparações é O (n2).
3. Merge Sort: Utiliza a ideia "dividir para conquistar", dividindo a lista e depois mesclando as sublistas ordenadas com recursividade. A complexidade no pior caso é O (n log n).
4. Quick Sort: Escolhe um dos elementos do vetor para ser o pivô. O pior caso ocorre quando o pivô é o menor ou o maior elemento, como resultado O (n2). Geralmente, tem desempenho próximo à O (n log n) com boas escolhas de pivô.

Gráfico

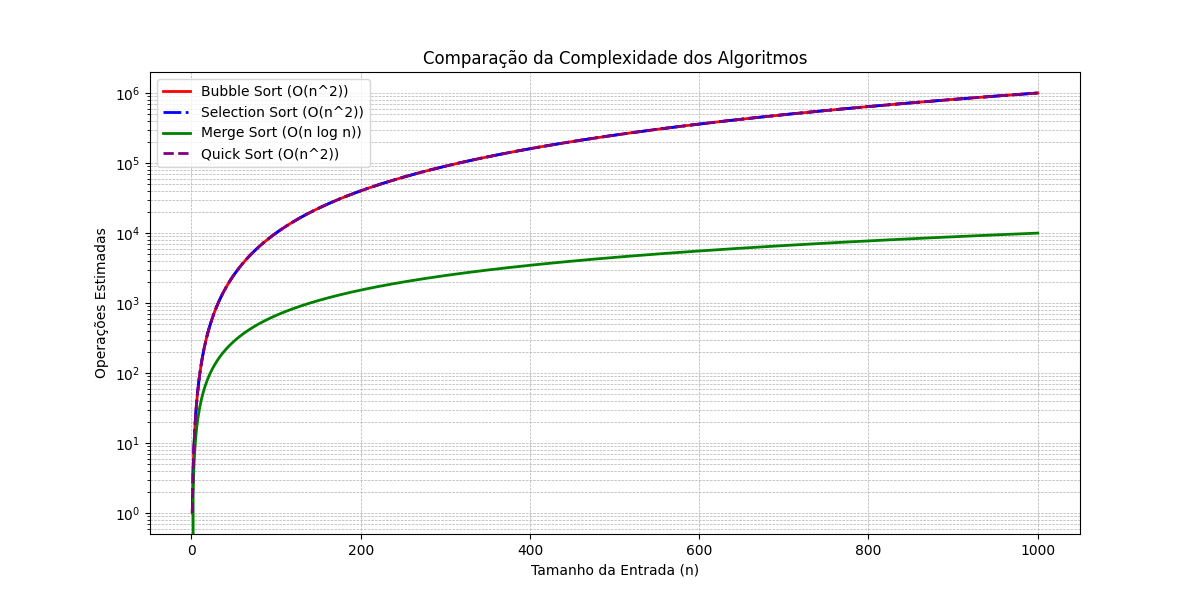


Gráfico representando o pior cenário para cada estrutura de ordenação em forma de equação

Discussão

Bubble Sort e Selection Sort:

Os dois apresentam uma complexidade de O (n2), o que apresenta nos gráficos pelo crescimento rápido para entradas maiores.

Em listas com números exorbitantes de dados, eles não conseguem ter uma eficiência boa, mesmo com o Selection Sort podendo as vezes, ter a metade da quantidade de comparações, que o Bubble Sort possui.

Merge Sort:

Sua complexidade de O (n log n) faz com que ele tenha um crescimento muito mais deavgar comparado ao O (n2).

Ele consegue ser muito mais eficiente com quantidades enormes de dados, dividindo-os em duas partes, consecutivamente até chegar no último valor dos dois lados, além da recursividade. Porém esses tipos de recurso, deixa o algoritmo muito pesado, ocupando muita memória do computador.

Quick Sort:

Embora sua complexidade no pior caso seja O (n2) seu funcionamento, geralmente é mais rápido que Merge Sort podendo ter a complexidade de O (n log n), dependendo da escolha do pivô ou se já está ordenada, além da menor sobrecarga sobre o sistema. No entanto, no pior cenário (lista já ordenada ou escolha ruim de pivô), o desempenho cai para O (n2).