**Relatório de Teste de Desempenho de Algoritmos de Ordenação**

**Lucas Azevedo Dias**

**Introdução:**

Este relatório descreve os resultados de testes de desempenho para três algoritmos de ordenação: *Bubble Sort*, *Merge Sort* e *Quick Sort*. O objetivo desses testes é avaliar o desempenho de cada algoritmo em relação ao tempo de ordenação para diferentes quantidades de elementos a serem ordenados.

**Testes Realizados:**

Foram realizados testes em cinco diferentes tamanhos de conjunto de dados, com variação na quantidade de elementos a serem ordenados.

**Resultados:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Bubble Sort | **Times** | **DeltaTime (ms)** | **TotalIters** | **TotalSwaps** | **(TotalIters/Times)** | **(TotalSwaps/Times)** |
| 50 | 2 | 1862 | 649 | 3724,00% | 1298,00% |
| 500 | 8 | 229041 | 61311 | 45808,20% | 12262,20% |
| 1000 | 9 | 960039 | 244964 | 96003,90% | 24496,40% |
| 5000 | 45 | 24265146 | 6150125 | 485302,92% | 123002,50% |
| 10000 | 229 | 98940105 | 24889100 | 989401,05% | 248891,00% |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Merge Sort | **Times** | **DeltaTime (ms)** | **TotalIters** | **TotalSwaps** | **(TotalIters/Times)** | **(TotalSwaps/Times)** |
| 50 | 0 | 201 | 6 | 402,00% | 12,00% |
| 500 | 1 | 3627 | 124 | 725,40% | 24,80% |
| 1000 | 2 | 8226 | 240 | 822,60% | 24,00% |
| 5000 | 2 | 53175 | 1045 | 1063,50% | 20,90% |
| 10000 | 2 | 116285 | 2057 | 1162,85% | 20,57% |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Quick Sort | **Times** | **DeltaTime (ms)** | **TotalIters** | **TotalSwaps** | **(TotalIters/Times)** | **(TotalSwaps/Times)** |
| 50 | 0 | 93 | 82 | 186,00% | 164,00% |
| 500 | 0 | 1342 | 1208 | 268,40% | 241,60% |
| 1000 | 0 | 2841 | 2556 | 284,10% | 255,60% |
| 5000 | 2 | 17096 | 15630 | 341,92% | 312,60% |
| 10000 | 3 | 36103 | 2057 | 361,03% | 20,57% |

**Conclusão:**

De forma geral, o algoritmo de *Merge Sort* apresentou os melhores resultados, onde por mais que tenha tido mais iterações relativamente a quantidade de inserções, possuiu uma quantidade ínfima de trocas de posições. Assim, garantindo a ele uma vantagem de tempo sobre os outros dois algoritmos apresentados.

Link do GitHub: <https://github.com/lucas-azdias/Resolucao-de-Problemas-Estruturados-em-Computacao/tree/main/%5BTAREFAS%5D/TDE%2003%20-%20Ordena%C3%A7%C3%A3o%20RA%2004%20-%204%20hs%20Aula>