Ficha de comparação de algoritmos de ordenação 2022/2

|  |  |
| --- | --- |
| *Nome do(a) aluno(a):* | Rogério Rodrigues Silva |

## Especificação do equipamento

|  |  |
| --- | --- |
| **Sistema Operacional** | Windows 11 |
| **Processador** | AMD Ryzen 5 4600G |
| **Arquitetura (32 ou 64 bits)** | 64BITS |
| **Memória RAM Total** | DDR4 16GB |
| **Versão do Python** | 3.10.7 |

Para que a análise de eficiência (abaixo) tenha resultados comparáveis, todos os testes precisam ser realizados no mesmo equipamento (descrito acima).

## Análise de eficiência

Anotar, no quadro abaixo, o tempo e a memória dispendidos para a execução de cada algoritmo trabalhando sobre diferentes tamanhos de amostra.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Tamanho da amostra | | | | | | | |
| Algoritmo | **10.000** | | **25.000** | | **50.000** | | **100.000** | |
| **Bubble Sort** | *Tempo* | 107185.12773513794ms | *Tempo* | 686643.0926322937ms | *Tempo* | 2800089.185476303ms | *Tempo* | 50544387.26902008ms |
| *Memória* | 0.000240325927734375 MB | *Memória* | 0.000240325927734375 MB | *Memória* | 0.000247955322265625 MB | *Memória* | 0.000255584716796875 MB |
| **Selection Sort** | *Tempo* | 28026.343822479248ms | *Tempo* | 196702.25930213928ms | *Tempo* | 725110.6472015381ms | *Tempo* | 2976792.6099300385ms |
| *Memória* | 0.0003509521484375 MB | *Memória* | 0.0003509521484375 MB | *Memória* | 0.00035858154296875 MB | *Memória* | 0.00035858154296875 MB |
| **Merge Sort RECURSIVO** | *Tempo* | 141.03436470031738ms | *Tempo* | 413.3760929107666ms | *Tempo* | 905.8325290679932ms | *Tempo* | 1957.336187362671ms |
| *Memória* | 0.24070167541503906 MB | *Memória* | 0.5978260040283203 MB | *Memória* | 1.1947269439697266 MB | *Memória* | 2.2983264923095703 MB |
| **Merge Sort ITERATIVO** | *Tempo* | 212.19277381896973ms | *Tempo* | 583.6772918701172ms | *Tempo* | 1259.225845336914ms | *Tempo* | 2698.1592178344727ms |
| *Memória* | 0.08392715454101562 MB | *Memória* | 0.22433853149414062 MB | *Memória* | 0.4479408264160156 MB | *Memória* | 0.8951454162597656 MB |
| **Quick Sort RECURSIVO** | *Tempo* | 225.2044677734375ms | *Tempo* | 573.8661289215088ms | *Tempo* | 1304.8121929168701ms | *Tempo* | 2874.9330043792725ms |
| *Memória* | 0.020040512084960938 MB | *Memória* | 0.030492782592773438 MB | *Memória* | 0.09529685974121094 MB | *Memória* | 0.09790992736816406 MB |
| **Quick Sort ITERATIVO** | *Tempo* | 200.29735565185547ms | *Tempo* | 513.5576725006104ms | *Tempo* | 1206.0940265655518ms | *Tempo* | 2667.459487915039ms |
| *Memória* | 0.07791519165039062 MB | *Memória* | 0.1924896240234375 MB | *Memória* | 0.3832244873046875 MB | *Memória* | 0.7649078369140625 MB |
| **Quick Sort RECURSIVO** | *Tempo* | 225.2044677734375ms | *Tempo* | 573.8661289215088ms | *Tempo* | 1304.8121929168701ms | *Tempo* | 2874.9330043792725ms |
| *Memória* | 0.020040512084960938 MB | *Memória* | 0.030492782592773438 MB | *Memória* | 0.09529685974121094 MB | *Memória* | 0.09790992736816406 MB |

## Passadas, comparações e trocas

Anotar, no quadro abaixo, o tempo e a memória dispendidos para a execução de cada algoritmo trabalhando sobre diferentes tamanhos de amostra.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Tamanho da amostra | | | | | | | |
| Algoritmo | **10.000** | | **25.000** | | **50.000** | | **100.000** | |
| **Bubble Sort** | *Passadas* | 9.942 | *Passadas* | 24.675 | *Passadas* | 49.353 | *Passadas* | 99.259 |
| *Comparações* | 99.410.058 | *Comparações* | 616.850.325 | *Comparações* | 2.467.600.647 | *Comparações* | 9.925.800.741 |
| *Trocas* | 25.297.553 | *Trocas* | 157.485.392 | *Trocas* | 625.828.205 | *Trocas* | 2.498.177.562 |
| **Selection Sort** | *Passadas* | 9.999 | *Passadas* | 24.999 | *Passadas* | 49.999 | *Passadas* | 99.999 |
| *Comparações* | 49.995.000 | *Comparações* | 312.487.500 | *Comparações* | 1.249.975.000 | *Comparações* | 4.999.950.000 |
| *Trocas* | 9.990 | *Trocas* | 24.984 | *Trocas* | 49.991 | *Trocas* | 99.982 |
| **Quick Sort RECURSIVO** | *Passadas* | 6.739 | *Passadas* | 16.815 | *Passadas* | 33.606 | *Passadas* | 67.259 |
| *Comparações* | 247.491 | *Comparações* | 616.240 | *Comparações* | 1.435.726 | *Comparações* | 3.148.139 |
| *Trocas* | 74.724 | *Trocas* | 183.025 | *Trocas* | 442.266 | *Trocas* | 924.104 |
| **Quick Sort ITERATIVO** | *Passadas* | 6.710 | *Passadas* | 16.849 | *Passadas* | 33.597 | *Passadas* | 67.186 |
| *Comparações* | 172.911 | *Comparações* | 436.273 | *Comparações* | 996.399 | *Comparações* | 2.238.251 |
| *Trocas* | 81.687 | *Trocas* | 199.826 | *Trocas* | 486.840 | *Trocas* | 1.013.966 |
| **Quick Sort RECURSIVO** | *Passadas* | 6.739 | *Passadas* | 16.815 | *Passadas* | 33.606 | *Passadas* | 67.259 |
| *Comparações* | 247.491 | *Comparações* | 616.240 | *Comparações* | 1.435.726 | *Comparações* | 3.148.139 |
| *Trocas* | 74.724 | *Trocas* | 183.025 | *Trocas* | 442.266 | *Trocas* | 924.104 |

## Considerações finais

Descreva, no quadro abaixo, suas considerações acerca do resultado da comparação dos algoritmos de ordenação.

Com o objetivo de economizar tempo e ganhar performance com o consumo da memória do computador, o QUICK SORT RECURSIVO saiu na frente. Com o uso dele a quantidade de memória teve um consumo bem menor, porem o tempo relacionado ao ITERATIVO está quase igual, tendo uma pequena diferença entre os dois sendo o ITERATIVO um pouco mais rápido porem com um consumo muito maior de memória.

Os algoritmos de ordenação BUBBLE SORT e SELECTION SORT estão entre os piores, pois demandam de um tempo muito grande pra poderem ser executados, gerando assim um atraso no sistema para poder fazer a ordenação de uma lista, em contra partida tem um consumo muito baixo de memória.

O MERGE SORT por fim, tem uma eficiência muito boa, referente a velocidade, porem o consumo de memória ainda é elevado em comparação com seu concorrente QUICK SORT. Sendo assim se tornando ineficiente para listas muito grandes.