Ficha de comparação de algoritmos de ordenação 2024/1

|  |  |
| --- | --- |
| *Nome do(a) aluno(a):* | Vinicius de Paula |

## Especificação do equipamento

|  |  |
| --- | --- |
| **Sistema Operacional** | Windows 10 |
| **Processador** | Intel(R) Core(TM) i5-10400F CPU @ 2.90GHz 2.90 GHz |
| **Arquitetura (32 ou 64 bits)** | 64bits |
| **Memória RAM Total** | 16GB RAM |
| **Versão do Python** | 3.12 (64bits) |

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* | **140932.3923587799ms** | *Tempo* | **890272.839307785ms** | *Tempo* | 3541332.727909088ms | *Tempo* | 13912225.165128708ms |
| *Memória* | **0.000240325927734375MB** | *Memória* | **0.000240325927734375MB** | *Memória* | 0.000247955322265625MB | *Memória* | 0.000255584716796875MB |
| **Selection Sort** | *Tempo* | **36915.65489768982ms** | *Tempo* | **240073.53472709656ms** | *Tempo* | 939012.8448009491ms | *Tempo* | 3701600.703239441ms |
| *Memória* | **0.0003509521484375MB** | *Memória* | **0.0003509521484375MB** | *Memória* | 0.00035858154296875MB | *Memória* | 0.00035858154296875MB |
| **Merge Sort RECURSIVO** | *Tempo* | **177.14786529541016ms** | *Tempo* | **499.18127059936523ms** | *Tempo* | 1074.1651058197021ms | *Tempo* | 2364.776134490967ms |
| *Memória* | **0.24131202697753906MB** | *Memória* | **0.5984363555908203MB** | *Memória* | 1.1953372955322266MB | *Memória* | 2.2989368438720703MB |
| **Merge Sort ITERATIVO** | *Tempo* | **242.36822128295898ms** | *Tempo* | **687.8573894500732ms** | *Tempo* | 1426.8312454223633ms | *Tempo* | 3401.686668395996ms |
| *Memória* | **0.08392715454101562MB** | *Memória* | **0.22433853149414062MB** | *Memória* | 0.4479408264160156MB | *Memória* | 0.8951454162597656MB |
| **Quick Sort RECURSIVO** | *Tempo* | **226.3946533203125ms** | *Tempo* | **568.4850215911865ms** | *Tempo* | 1304.3251037597656ms | *Tempo* | 2977.782726287842ms |
| *Memória* | 0.020399093627929688MB | *Memória* | 0.030851364135742188MB | *Memória* | 0.09565544128417969MB | *Memória* | 0.09826850891113281MB |
| **Quick Sort ITERATIVO** | *Tempo* | 218.48416328430176ms | *Tempo* | 578.9833068847656ms | *Tempo* | 1339.5168781280518ms | *Tempo* | 2908.337354660034ms |
| *Memória* | 0.07794189453125MB | *Memória* | 0.1924896240234375MB | *Memória* | 0.3832244873046875MB | *Memória* | 0.7649078369140625MB |
| **MELHOR RESULTADO** | *Tempo* | Merge Sort RECURSIVO | *Tempo* | Merge Sort RECURSIVO | *Tempo* | Merge Sort RECURSIVO | *Tempo* | Merge Sort RECURSIVO |
| *Memória* | Bubble Sort | *Memória* | Bubble Sort | *Memória* | Bubble Sort | *Memória* | Bubble Sort |

## 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* | **9942** | *Passadas* | **24675** | *Passadas* | 49353 | *Passadas* | 99259 |
| *Comparações* | **99410058** | *Comparações* | **616850325** | *Comparações* | 2467600647 | *Comparações* | 9925800741 |
| *Trocas* | **25297553** | *Trocas* | **157485392** | *Trocas* | 625828205 | *Trocas* | 2498177562 |
| **Selection Sort** | *Passadas* | **9999** | *Passadas* | **24999** | *Passadas* | 49999 | *Passadas* | 99999 |
| *Comparações* | **49995000** | *Comparações* | **312487500** | *Comparações* | 1249975000 | *Comparações* | 4999950000 |
| *Trocas* | **9990** | *Trocas* | **24984** | *Trocas* | 49991 | *Trocas* | 99982 |
| **Quick Sort RECURSIVO** | *Passadas* | **6739** | *Passadas* | **16815** | *Passadas* | 33606 | *Passadas* | 67259 |
| *Comparações* | 159922 | *Comparações* | 403744 | *Comparações* | 933897 | *Comparações* | 2105979 |
| *Trocas* | 74927 | *Trocas* | 183578 | *Trocas* | 443528 | *Trocas* | 926764 |
| **Quick Sort ITERATIVO** | *Passadas* | 6709 | *Passadas* | 16848 | *Passadas* | 33596 | *Passadas* | 67185 |
| *Comparações* | 159494 | *Comparações* | 402578 | *Comparações* | 929208 | *Comparações* | 2103882 |
| *Trocas* | 88396 | *Trocas* | 216674 | *Trocas* | 520436 | *Trocas* | 1081151 |
| **MELHOR RESULTADO** | *Passadas* | **Quick Sort**  **ITERATIVO** | *Passadas* | **Quick Sort RECURSIVO** | *Passadas* | **Quick Sort**  **ITERATIVO** | *Passadas* | **Quick Sort ITERATIVO** |
| *Comparações* | **Quick Sort**  **ITERATIVO** | *Comparações* | **Quick Sort**  **ITERATIVO** | *Comparações* | **Quick Sort**  **ITERATIVO** | *Comparações* | **Quick Sort ITERATIVO** |
| *Trocas* | **Selection Sort** | *Trocas* | Selection Sort | *Trocas* | Selection Sort | *Trocas* | Selection Sort |

## Considerações finais

Descreva, no quadro abaixo, suas considerações acerca do resultado da comparação dos algoritmos de ordenação.  
  
#############################################################################################  
  
Conseguimos perceber que o SelectionSort é melhor em todas as Trocas. E o QuickSortITERATIVO é melhor em todas as Comparações  
na questão de passadas ficamos entre QuickSortITERATIVO e QuickSortRECURSIVO.  
  
Percebemos que em questão de tempo o melhor é o MergeSortRECURSIVO  
e o melhor em uso de memória é o BubbleSort apesar de ser o mais demorado