­­­­Ficha de comparação de algoritmos de ordenação 2022/2

|  |  |
| --- | --- |
| *Nome do(a) aluno(a):* | Pamela Cristina Borelli |

## Especificação do equipamento

|  |  |
| --- | --- |
| **Sistema Operacional** | Windows 11 |
| **Processador** | Intel Core i7 |
| **Arquitetura (32 ou 64 bits)** | 64 bits |
| **Memória RAM Total** | 8MB |
| **Versão do Python** |  |

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* | **135373.93498420715ms** | *Tempo* | **751625.6291866302ms** | *Tempo* | 3.702.813.9576911926ms | *Tempo* | 15.220.834.599018097ms |
| *Memória* | **0.000240325927734375MB** | *Memória* | **0.000240325927734375 MB** | *Memória* | 0.000247955322265625 MB | *Memória* | 0.000255584716796875 MB |
| **Selection Sort** | *Tempo* | **26509.589672088623ms** | *Tempo* | **187677.3271560669ms** | *Tempo* | 850201.1330127716ms | *Tempo* | 3516813.152551651ms |
| *Memória* | **0.0003509521484375** | *Memória* | **0.0003509521484375 MB** | *Memória* | 0.00035858154296875 MB | *Memória* | 0.00035858154296875 MB |
| **Merge Sort RECURSIVO** | *Tempo* | **108.47258567810059ms** | *Tempo* | **127.10881233215332ms** | *Tempo* | 105.61585426330566ms | *Tempo* | 111.78135871887207ms |
| *Memória* | **0.24070167541503906 MB** | *Memória* | **0.24070167541503906 MB** | *Memória* | 0.2407016754150390 | *Memória* | 0.24070167541503906 |
| **Merge Sort ITERATIVO** | *Tempo* | **347.2440242767334ms** | *Tempo* | **1013.7443542480469ms** | *Tempo* | 2147.2671031951904ms | *Tempo* | **4606.170654296875ms** |
| *Memória* | **0.08392715454101562 MB** | *Memória* | **0.22433853149414062 MB** | *Memória* | 0.4479408264160156 MB | *Memória* | **0.8951454162597656 MB** |
| **Quick Sort RECURSIVO** | *Tempo* | **185.24742126464844ms** | *Tempo* | **447.7686882019043ms** | *Tempo* | 1166.6054725646973ms | *Tempo* | 2793.597936630249ms |
| *Memória* | 0.020040512084960938 | *Memória* | **0.030492782592773438 MB** | *Memória* | 0.09529685974121094 MB | *Memória* | 0.09790992736816406 MB |
| **Quick Sort ITERATIVO** | *Tempo* | 326.53284072875977ms | *Tempo* | 752.9201507568359ms | *Tempo* | 1757.6756477355957ms | *Tempo* | 3612.7748489379883ms |
| *Memória* | 0.07794189453125 MB | *Memória* | 0.1924896240234375 MB | *Memória* | 0.3832244873046875 MB | *Memória* | 0.7649078369140625 MB |
| **MELHOR RESULTADO** | *Tempo* | Merge sort recursivo | *Tempo* | Merge sorte recursivo | *Tempo* | Merge sort recursovo | *Tempo* | Merge sorte recursivo |
| *Memória* | Bubble sort | *Memória* | Bubble sorte | *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* | 247491 | *Comparações* | 616240 | *Comparações* | 1435726 | *Comparações* | 3148139 |
| *Trocas* | 74724 | *Trocas* | 183025 | *Trocas* | 442266 | *Trocas* | 924104 |
| **Quick Sort ITERATIVO** | *Passadas* | 6710 | *Passadas* | 16849 | *Passadas* | 33597 | *Passadas* | 67186 |
| *Comparações* | 172911 | *Comparações* | 436273 | *Comparações* | 996399 | *Comparações* | 2238251 |
| *Trocas* | 81687 | *Trocas* | 199826 | *Trocas* | 486840 | *Trocas* | 1013966 |
| **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* | Quick sort iterativo | *Trocas* | Quick sort recursivo | *Trocas* | Quick sort recursivo | *Trocas* | Quick sort recursivo |

## Considerações finais

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

Os melhores resultados em relação ao tempo são do algoritmo Merge Sort Recursivo, porém em relação à memória utilizada o melhor foi Bubbe Sort, mesmo que nesse caso esse último algoritmo tenha o pior desempenho em relação ao tempo.

Já em relação às passadas, comparações e trocas, o Quick Sort Iterativo foi melhor em vários casos, e sem dúvida o melhor em relação às comparações. Porém o Quick Sort Recursivo apresentou melhores resultados nas trocas, principalmente em listas maiores; e um caso nas passadas.