Ficha de comparação de algoritmos de ordenação 2023/1

|  |  |
| --- | --- |
| *Nome do(a) aluno(a):* | João Paulo Falcuci Teixeira |

## Especificação do equipamento

|  |  |
| --- | --- |
| **Sistema Operacional** | Ubuntu 22.10 Kernel: 5.19.0 |
| **Processador** | Intel i5-5200U (4) @ 2.700GHz |
| **Arquitetura (32 ou 64 bits)** | 64 bits |
| **Memória RAM Total** | 8,00 GB DDR3 1600 MHz |
| **Versão do Python** | 3.11.2 |

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* | 0:01:43.652536 | *Tempo* | 0:11:22.774309 | *Tempo* | 0:45:45.062958 | *Tempo* | 3:08:49.366995 |
| *Memória* | **0.00029 MB** | *Memória* | **0.00029 MB** | *Memória* | **0.0003 MB** | *Memória* | **0.00031 MB** |
| **Selection Sort** | *Tempo* | 0:00:30.066958 | *Tempo* | 0:03:10.794170 | *Tempo* | 0:13:11.570059 | *Tempo* | 0:53:38.657204 |
| *Memória* | 0.00042 MB | *Memória* | 0.00042 MB | *Memória* | 0.00042 MB | *Memória* | 0.00042 MB |
| **Merge Sort RECURSIVO** | *Tempo* | **0:00:00.169351** | *Tempo* | **0:00:00.463508** | *Tempo* | **0:00:01.018003** | *Tempo* | **0:00:02.219619** |
| *Memória* | 0.24185 MB | *Memória* | 0.5916 MB | *Memória* | 1.18794 MB | *Memória* | 2.29099 MB |
| **Merge Sort ITERATIVO** | *Tempo* | 0:00:00.284596 | *Tempo* | 0:00:00.780980 | *Tempo* | 0:00:01.666968 | *Tempo* | 0:00:03.663735 |
| *Memória* | 0.08356 MB | *Memória* | 0.22397 MB | *Memória* | 0.44757 MB | *Memória* | 0.89478 MB |
| **Quick Sort RECURSIVO** | *Tempo* | 0:00:00.236792 | *Tempo* | 0:00:00.609582 | *Tempo* | 0:00:01.427936 | *Tempo* | 0:00:03.250276 |
| *Memória* | 0.02096 MB | *Memória* | 0.01416 MB | *Memória* | 0.07012 MB | *Memória* | 0.01786 MB |
| **Quick Sort ITERATIVO** | *Tempo* | 0:00:00.188127 | *Tempo* | 0:00:00.476993 | *Tempo* | 0:00:01.103665 | *Tempo* | 0:00:02.448788 |
| *Memória* | 0.07752 MB | *Memória* | 0.19207 MB | *Memória* | 0.3828 MB | *Memória* | 0.76448 MB |
| **MELHOR RESULTADO** | *Tempo* | Bubble Sort | *Tempo* | Bubble Sort | *Tempo* | Bubble Sort | *Tempo* | Bubble Sort |
| *Memória* | Merge Sort Recursivo | *Memória* | Merge Sort Recursivo | *Memória* | Merge Sort Recursivo | *Memória* | Merge Sort Recursivo |

## 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* | **6.709** | *Passadas* | 16.848 | *Passadas* | **33.596** | *Passadas* | **67.185** |
| *Comparações* | **159.494** | *Comparações* | **402.578** | *Comparações* | **929.208** | *Comparações* | **2.103.882** |
| *Trocas* | 88.396 | *Trocas* | 216.674 | *Trocas* | 520.436 | *Trocas* | 1.081.151 |
| **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.

Para os dados apresentados, o algoritmo que apresentou o menor consumo de memória, foi o Bubble Sort, enquanto o algoritmo que apresentou o menor tempo de execução foi o Merge Sort Recursivo. Por outro lado, o Bubble Sort foi o algoritmo que apresentou maior tempo de execução, enquanto o Merge Sort Recursivo o que mais consumiu memória no processo.

Destaca-se o Quick Sort como o algoritmo que, apesar de não ter tido os melhores resultados em tempo e em consumo de memória, se mostrou bastante rápido (principalmente na versão iterativa) e com baixo consumo de memória (principalmente na versão recursiva), sendo na minha opinião, o melhor “custo-benefício” entre os algoritmos testados.