**EP05 - Algoritmos de Ordenação**

*<Guilherme Araújo Mendes de Souza – 156437>*  
  
Este Exercício-Programa compara algoritmos de ordenação

* Descreva o ambiente de execução utilizado:

|  |
| --- |
| Processador: AMD Ryzen 5 5500U with Radeon Graphics 2.10 GHz  RAM: 8,00 GB (utilizável: 5,86 GB) (3200 MHz)  Sistema operacional: Windows 10 Home Single Language (x64)  Compilador: Eclipse IDE for C and C++ Developers - 2022-09  Programas abertos durante a execução do código: Firefox (Com 2 abas abertas) e MS Word. |

* Descreva o método de medição do tempo:

|  |
| --- |
| -- O método de medição do tempo adotado foi o clock() da biblioteca time.h.  -- Na função principal é definido duas variáveis início e fim do tipo clock\_t.  -- A variável inicio marca o clock no momento que a função principal chama a função de ordenação, e a variável fim marca o clock após a execução da função de ordenação.  -- A variável tempo\_decorrido do tipo double, recebe o valor do clock final da variável fim e do clock inicial da variável inicio. E realiza a seguinte operação ((fim-inicio)/ CLOCKS\_PER\_SEC)\*1000  OBS: CLOCKS\_PER\_SEC é uma constante definida na biblioteca time.h |

* Tempos de execução em milessegundos:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Método/Tempo** | **n=5000** | **n=10000** | **n=15000** | **n=20000** |
| **InsertionSort** | 25.000000 milessegundos | 81.000000 milessegundos | 175.000000 milessegundos | 316.000000 milessegundos |
| **SelectionSort** | 29.000000 milessegundos | 116.000000 milessegundos | 266.000000 milessegundos | 495.000000 milessegundos |
| **MergeSort** | 2.000000 milessegundos | 4.000000 milessegundos | 6.000000 milessegundos | 10.000000 milessegundos |
| **HeapSort** | 1.000000 milessegundos | 2.000000 milessegundos | 4.000000 milessegundos | 5.000000 milessegundos |
| **QuickSort** | 1.000000 milessegundos | 2.000000 milessegundos | 2.000000 milessegundos | 3.000000 milessegundos |

* Código fonte:

|  |  |
| --- | --- |
| **Método** | **Link para a implementação no GitHub** |
| **InsertionSort** | https://github.com/Gu1lh3rm3-Arauj0/Algoritmos-e-Estruturas-de-Dados-I/blob/master/ep05%20-%20InsertionSort.c |
| **SelectionSort** | https://github.com/Gu1lh3rm3-Arauj0/Algoritmos-e-Estruturas-de-Dados-I/blob/master/ep05%20-%20SelectionSort.c |
| **MergeSort** | https://github.com/Gu1lh3rm3-Arauj0/Algoritmos-e-Estruturas-de-Dados-I/blob/master/ep05%20-%20Mergesort.c |
| **HeapSort** | https://github.com/Gu1lh3rm3-Arauj0/Algoritmos-e-Estruturas-de-Dados-I/blob/master/ep05%20-%20HeapSort.c |
| **QuickSort** | https://github.com/Gu1lh3rm3-Arauj0/Algoritmos-e-Estruturas-de-Dados-I/blob/master/ep05%20-%20Quicksort.c |