**CCDSALG Term 3, AY 2019 – 2020**

Project 1 – Comparing Sorting Algorithms

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Section** | **Names** | **Task 1** | **Task 2** | **Task 3** | **Task 4** |
| S12 | Gan, John Matthew Ong | X | X | X |  |
| S16 | Noblefranca, Jose Noel Cleofe | X | X |  |  |
| S15 | Remudaro, Angelo Alvarez |  | X |  | X |

Fill this part with your section and names. For the tasks, put an X mark if you have performed the specified task. Please refer to the project specifications for the tasks.

**LIST OF SORTING ALGORITHMS**

|  |  |  |
| --- | --- | --- |
| **Sorting Algorithm** | **Author (if available)** | **Downloaded From** |
| 1. Bubble sort | programmingsimplified.com | <https://www.programmingsimplified.com/c/source-code/c-program-bubble-sort> |
| 2. Insertion sort | hackerearth.com | <https://www.hackerearth.com/practice/algorithms/sorting/insertion-sort/tutorial/> |
| 3. Selection sort | geeksforgeeks.org | <https://www.geeksforgeeks.org/selection-sort/> |
| 4.  Merge sort | geeksforgeeks.org | <https://www.geeksforgeeks.org/merge-sort/> |
| 5.  Quick sort | geeksforgeeks.org | <https://www.geeksforgeeks.org/quick-sort/> |
| 6.  Radix sort | geeksforgeeks.org | <https://www.geeksforgeeks.org/radix-sort/> |

**COMPARISON TABLE**

**M = (9)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Size** | **Average Machine Execution Time (miliseconds)** | | | | | |
| **Bubble** | **Insertion** | **Selection** | **Merge** | **Quick** | **Radix** |
| 1024 | **2.292** | **0.674** | **1.204** | **0.251** | **0.199** | **0.0** |
| 2048 | **8.282** | **2.096** | **4.993** | **0.399** | **0.298** | **0.399** |
| 4096 | **35.876** | **9.872** | **20.748** | **0.4** | **0.397** | **0.399** |
| 8192 | **245.05** | **0.058808** | **136.192** | **2.294** | **1.993** | **1.395** |
| 16384 | **1110.002** | **213.791** | **453.668** | **4.083** | **3.695** | **2.587** |
| 32768 | **4354.261** | **772.954** | **1537.165** | **7.59** | **6.191** | **4.292** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Size** | **Average Counter Value (in millions)** | | | | | |
| **Bubble** | **Insertion** | **Selection** | **Merge** | **Quick** | **Radix** |
| 1024 | **524799** | **273881** | **524800** | **11276** | **12477** | **10285** |
| 2048 | **2098175** | **1043734** | **2098176** | **24589** | **27369** | **20525** |
| 4096 | **8390655** | **4246568** | **8390656** | **53262** | **58675** | **41005** |
| 8192 | **33558528** | **16522178** | **33558528** | **114703** | **128216** | **81965** |
| 16384 | **134225920** | **66937032** | **134225920** | **245776** | **290103** | **163885** |
| 32768 | **536887296** | **266814240** | **536887296** | **524305** | **617586** | **327725** |

**GRAPHS**

Copy/paste the graphs here, make sure it is big enough to see the trend in the increase of the average Machine Execution Time (MET) and the average counter value.

**DISCUSSION**

Explain interesting findings based on your experiments.ss

**DISCUSSION**

Explain interesting findings based on your experiments.

**DISCUSSION**

Explain interesting findings based on your experiments.