Michael Lees

November 1, 2022

0001-Project\_EOT

|  |  |
| --- | --- |
| Num. of Elements/array | Average Loop Time: (ms) |
| 500 | 436.45 |
|  |
| 2500 | 9673.83 |  |
|  |
| 5000 | 39452.43 |  |
|  |

Chart, line chart

Description automatically generated The Bubble Sort algorithm has a time complexity value of and when applied to one thousand arrays each with five hundred, two thousand and five hundred, and five thousand elements, the function returns the following data. As expected, with each increase in the number of elements within each array, the function returns an increasing average execution time.

Chart, line chart

Description automatically generated The Selection Sort algorithm also has a time complexity value of and when applied to one thousand arrays each with five hundred, two thousand and five hundred, and five thousand elements, the function returns the following data. As expected, with each increase in the number of elements within each array, the function returns an increasing average execution time.

|  |  |
| --- | --- |
| Num. of Elements/array | Average Loop Time: (ms) |
| 500 | 231.16 |
|  |
| 2500 | 4399.85 |  |
|  |
| 5000 | 16730.96 |  |
|  |

Chart

Description automatically generated with medium confidence

After analyzing the data side by side, the selection sort algorithm performs at a faster average execution rate than the bubble sort algorithm. The bubble sort algorithm returns a best fit line of and the selection sort algorithm returns a best fit line of . These sorting functions performed as expected using the C++ programming language on a 2020 MacBook Pro 13” with a 2.3Ghz Quad-core Intel Core i7 with 16GB of 3733MHz LPDDR4x Ram running MacOS Ventura 13.0. (It has been compiled again for a Windows machine) The results were to be expected for the sorting algorithms after countless research into both algorithms.