**HW 1 – Merge and Insertion Sort**

To collect these times, I modified my program to first prompt the user for a value which woul determine the size of the array to sort. I then created and populated an array of size with random numbers. I stored the time right before my call to the sorting algorithm, and subtracted it from the time right after the sorting algorithm completed to find the total time elapsed in sorting.

Written report and data analysis

* 1. Runtimes:
     1. Merge sort

|  |  |
| --- | --- |
| **n** | **Runtime (s)** |
| 100 | 0.0009 |
| 500 | 0.0022 |
| 1,000 | 0.0039 |
| 5,000 | 0.0183 |
| 10,000 | 0.0387 |
| 50,000 | 0.2252 |
| 100,000 | 0.4763 |
| 200,000 | 1.0131 |
| 500,000 | 2.8181 |
| 1,000,000 | 5.7429 |

* + 1. Insertion sort

|  |  |
| --- | --- |
| **n** | **Runtime (s)** |
| 100 | 0.0014 |
| 500 | 0.0117 |
| 1,000 | 0.0410 |
| 1,500 | 0.0917 |
| 2,000 | 0.1480 |
| 5,000 | 0.8959 |
| 10,000 | 3.4306 |
| 20,000 | 14.1836 |
| 30,000 | 31.7555 |
| 40,000 | 59.2988 |

* 1. Data plot and fit curve

For merge sort, although the algorithm is , I was not able to get excel to give me a correct logarithmic regression. However, when plotting an graph in Desmos, the shape of the graph matches the data for merge sort extremely well.

For insertion sort, the data confirms that the runtime is . The equation of the regression is given in the plot.

* 1. Combined graph
  2. Prediction

Since my data includes for merge sort, I will be predicting the runtime for . For merge sort, 4×10^(-13)×2000000^2 + 5×10^(-6)×2000000 - 0.0206 gives 11.5794 seconds. For insertion sort, 4×10^(-8)×2000000^2 - 1×10^(-4)×2000000 + 0.1706 gives 159800.1706 seconds, approximately 2 days.