Brady Cassada

Algorithms Lab 1

**Basic Times for All 3 Algorithms**

The full set of data can be found in the excel file, below are abbreviated tables.

**Best Case Avg. Time to Sort (s)**

|  |  |  |
| --- | --- | --- |
| **N** | **Insertion** | **Merge** |
| **10** | 7.11E-08 | 7.13E-07 |
| **50** | 5.88E-08 | 2.19E-06 |
| **100** | 6.44E-08 | 5.82E-06 |
| **500** | 2.11E-07 | 2.10E-05 |
| **1000** | 5.16E-06 | 5.00E-04 |
| **10000** | 6.22E-05 | 5.63E-03 |

**Worst Case Avg. Time to Sort (s)**

|  |  |  |
| --- | --- | --- |
| **N** | **Insertion** | **Merge** |
| **10** | 8.25E-08 | 6.16E-07 |
| **50** | 1.00E-06 | 1.86E-06 |
| **100** | 3.69E-06 | 3.85E-06 |
| **500** | 5.56E-05 | 2.07E-05 |
| **1000** | 2.62E-04 | 4.53E-05 |
| **10000** | 1.92 | 5.38E-03 |

**Random Sequence Avg. Time to Sort (s)**

|  |  |  |
| --- | --- | --- |
| **N** | **Insertion** | **Merge** |
| **10** | 1.32E-07 | 8.34E-07 |
| **50** | 8.21E-07 | 4.18E-06 |
| **100** | 2.81E-06 | 1.31E-05 |
| **500** | 5.63E-05 | 3.68E-05 |
| **1000** | 0.0111 | 8.16E-04 |
| **10000** | 1.11 | 9.15E-03 |

1000 Trials were conducted for each n value

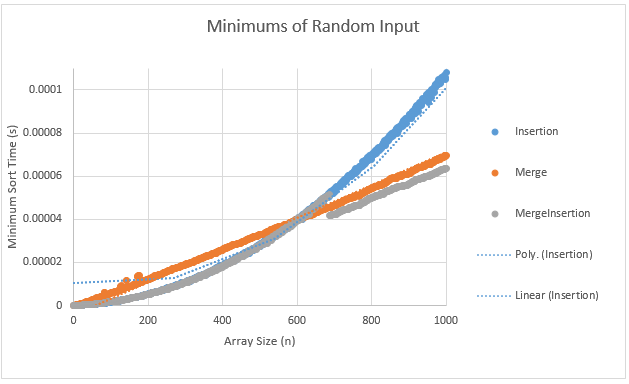
**Finding K**

In order to find a proper point to switch from merge sort to insertion sort, the two must be directly compared. I chose the data from the random inputs as I figured this would be closest to a “real world” estimate of input data (again, the average sort time for each n value was calculated over 1000 trials). One more thing to note is I decided to take the shortest time out of these trials because it produced the smoothest graph/data since it, more or less, ignores outliers.

A line of best fit was drawn for each of these plots and their equations are:

These functions meet at **n** = **687.9.** Theoretically, this is (an estimate of) where the insertion short algorithm becomes more efficient than the merge sort. Moving forward, the *K* value will be **688**.

Let’s look at the same plot with the addition of the merge-insertion times:



As you can see, the *K* value isn’t exactly spot on but it is certainly pretty close to the point we’d like to switch to insertion sort. Since all three graphs seem to be coming together at around 600, it might be best to switch our *K* value to ~600.

Ah, much better. The merge-insertion sort performs better than the plain merge sort and is overall the fastest.

Finally, let’s take a look at the average best, worst, and random cases for just merge-insertion: