**Hypothesis**

Insertion Sort will run faster than Merge Sort for arrays with length not exceeding 50.

**Methods**

Describe step-by-step the experiments that you conduct. Provide the source code that you use, and details about which compiler you use, how you compile it (*e.g*., optimization flags), and the range of inputs that you feed into your program.  You may provide a link to a repository rather than copying the code into your file.  Your methods should accurately reflect how you actually generated your data **such that, someone reading them can replicate your experiment**.

**Results**

**A graph of a line

Description automatically generated with medium confidence**

After running the experiment, trying arrays with size from 1 to 2000, we see that the actual runtime of Insertion Sort follows its time complexity graph of O(n2), and actual runtime of Merge Sort follows its time complexity graph of O(n log n).

**Discussion**

The result from the experiment has proved my hypothesis to be incorrect. It’s quite surprising that Merge Sort outperforms Insertion Sort even at small-sized arrays (n < 50). Also, for array of size less than 50, the difference in runtime between the 2 sorting algorithms is indistinguishable from each other, indicating that Insertion Sort does not perform better than Merge Sort even for array with very small size.

**Conclusions**

Let n be size of the array. Insertion Sort and Merge Sort perform quite the same for n < 200. For n > 200, Merge Sort performs much better than Insertion Sort do.