

Brigham Campbell

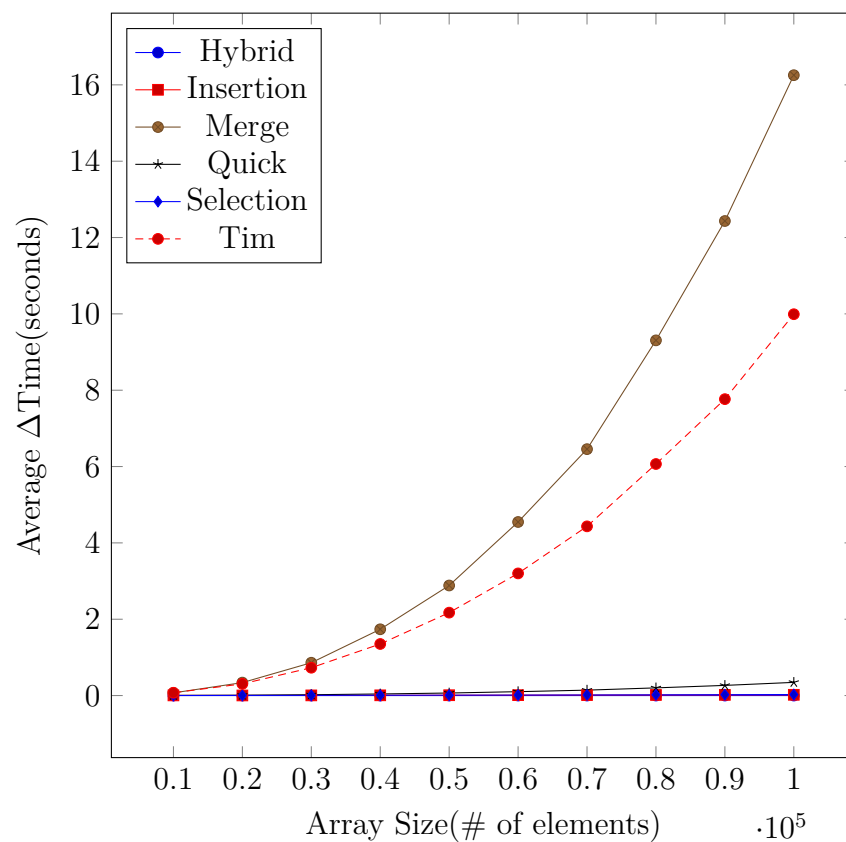
Dr. Griffith

CS 2235

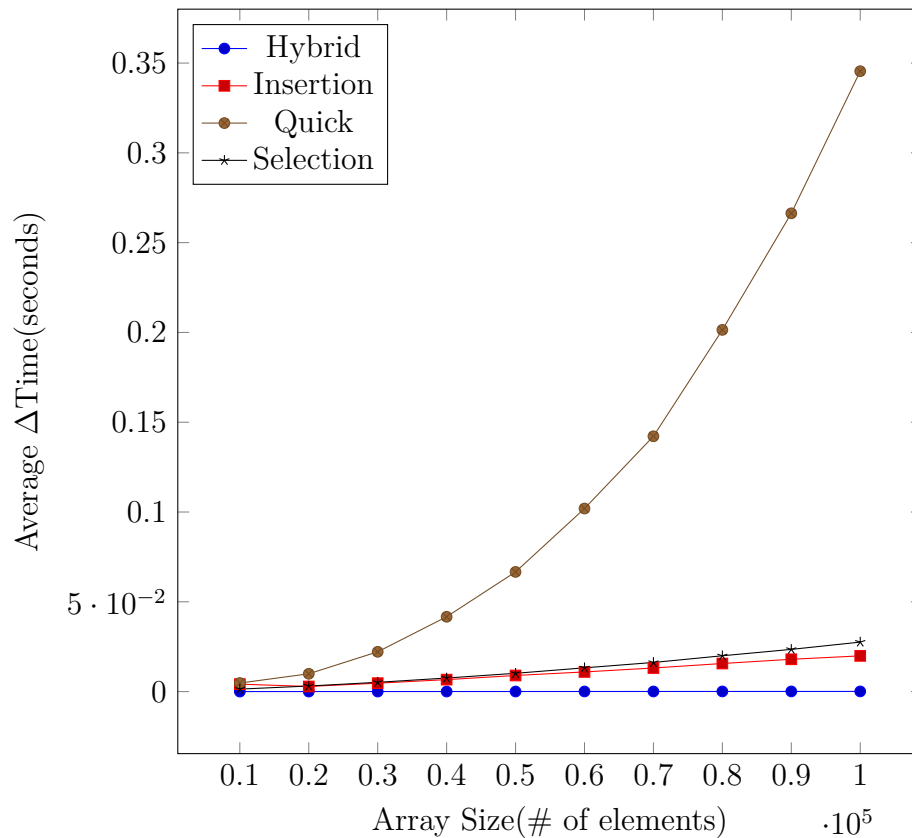
26 September 2019

Programming Project 01

The following figure is the result of plotting the averages of the length of time six different sorting algorithms take to sort an array of variable length. The averages have been computed by running each sorting algorithm 50 times per array size.



As the graph illustrates, MergeSort and TimSort take much more time than the other functions. Below is a graph excluding the two algorithms to better show how they compare to each other.



With these results, it's clear that HybridSort is by far the best sorting function among these implementations. This is strange because TimSort should be faster in theory. However, this makes sense because TimSort has been implemented recursively. In Java, function calls have significant overhead, so thousands of function calls will scale the time an algorithm takes to complete by a large factor.

Taking into account the results show here, I'd recommend HybridSort. However, I'd also consider that a non-recursive implementation of TimSort could be even faster than HybridSort as implemented here.