

Assignment 4 - Sorts Write Up

Graph Observations:

Data Link: [Google Sheets](#)

Using the six different sorts I created, I ran them on 1,500 elements, 10,000 elements, and 100,000 elements. As you can see in the first graph, Selection and Stable Selection's times increase with steeper curve than insertion sort.

In the second graph, I didn't run either selection sorts because I want to focus on the curves of the other four sorts. As you can see, Insertion sort's time increases with a curve and the other three seems to increase in a straight line as the number of elements increases to 10,000.

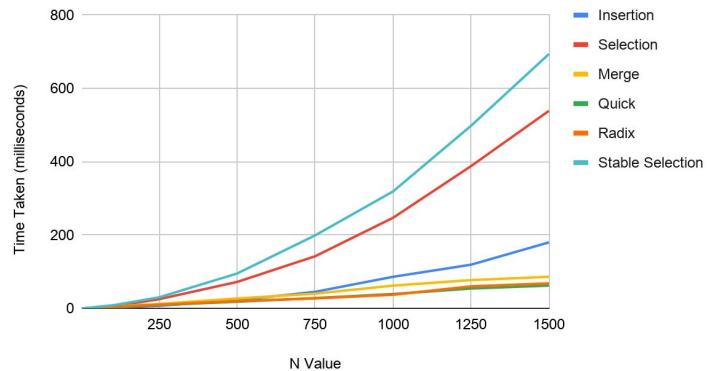
In the third graph, I only ran Merge, Quick, and Radix as the number of elements goes up to 100,000. I would like to bring your attention to around the 25,000 point. Every time before that, quick sort is faster than radix but as the number increases, Radix's time becomes smaller. Merge has always been the slowest out of the three so quick sort is always better than merge.

I ran these sorts multiple times and got the average for each run for those number of elements to find the average time taken.

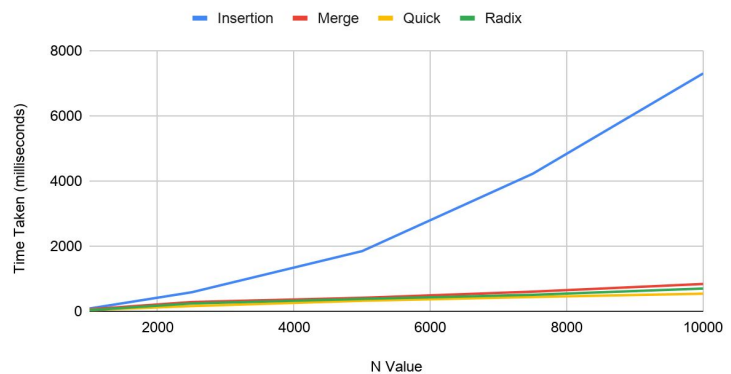
Analysis:

Insertion runs on big O of N^2 as it has to traverse through the list twice. The first is to compare numbers from right to left to make sure that all numbers on the left side are sorted. The second is to actually traverse numbers from left to right.

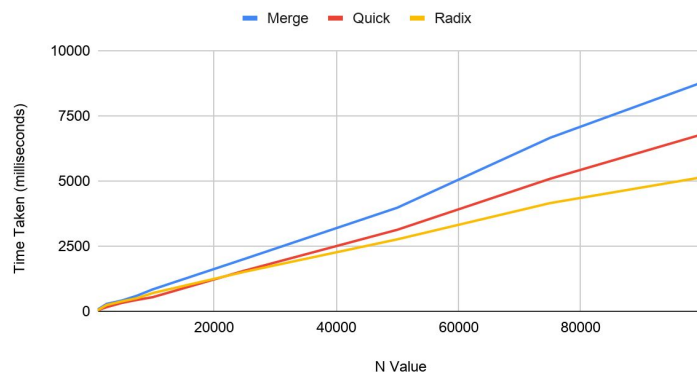
Time Taken to Sort 1,500 Elements



Time Taken to Sort 10,000 Elements



Time Taken to Sort 100,000 Elements



Selection runs on big O of N^2 . Finding the minimum number every time requires one traverse and finding comparing it with every other number requires the over traverse.

Merge and Quick runs on big O of $N\log N$. Since these sorts split an array in half. Each side will be $N/2$, and since it has to be partitioned N times, $2^x=N$. Then since there will be N comparisons, $N\log N$.

Error:

The main error in this set of data is the random variables generated, but I was able to remove that problem by running multiple iterations and finding the average time. Other than that, there are nothing else to consider.