

TABLE I
RESULTS

Algorithm	Start	End	Elapsed Time	Rank
QuickSort	4.255	4.286	0.031	1
InsertionSort	4.29	4.336	0.046	2
BubbleSort	4.34	4.474	0.134	4
SelectionSort	4.479	4.568	0.089	3

Comparison of Four Sorting Algorithms in C++

Grady Lynch, CPSC 350 Section 1

Quick Sort, Insertion Sort, Bubble Sort, and Selection Sort tested for an array of 200 doubles.

1.

For this test, I expected Quick Sort to be the fastest, and it was. It has the best big O run time of the algorithms I tested. However, I expected the difference to be greater. 200 seems like a large input size to me, but all the algorithms sorted it in less than one millisecond. I was also surprised by the performance of Selection Sort. I was under the impression that its performance was very bad because it is so intuitive and easy to implement. This was not the case. It was significantly faster than Bubble Sort, although it was also significantly slower than Quick and Insertion Sort. I implemented these algorithms in C++, but it would be interesting to see what sort of difference in the performance you would observe using another language. Unfortunately, this empirical analysis doesn't account for partially sorted data. Insertion Sort has a massive advantage when chunks of the array are already sorted, and this could skew results.