Bubble sort:

The average time complexity of bubble sort is O(n^2)
The best case for bubble sort is O(n)
The worst case for bubble sort is O(n^2)

Shell sort:

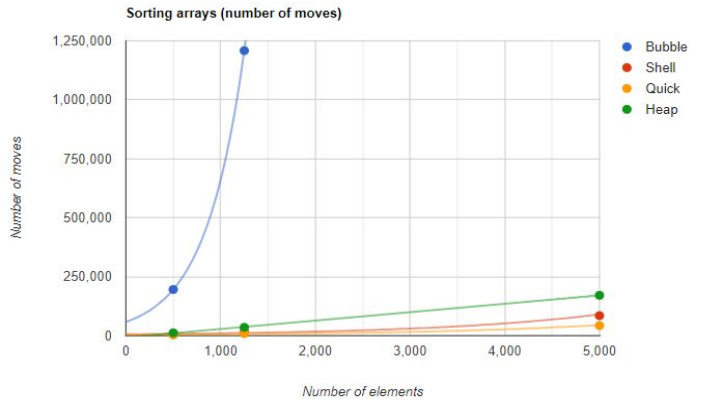
The average time complexity of shell sort is O(n^5/3)
The best case for shell sort is O(n logn)
The worst case for shell sort is O(n)

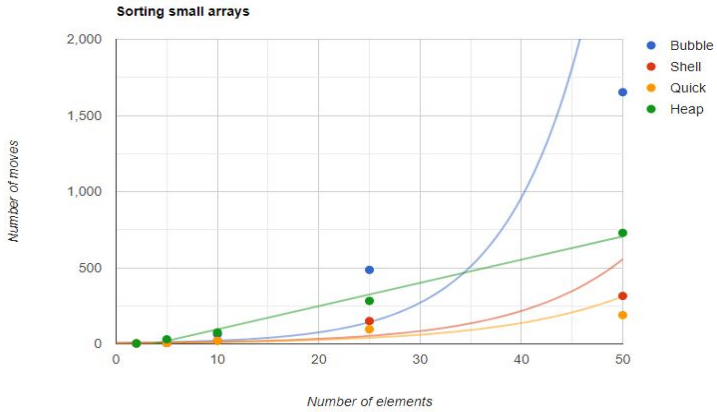
Heap sort:

The average time complexity of heap sort is O(n logn)
The best case for heap sort is O(n logn)
The worst case for heap sort is O(n logn)

Quick sort:

The average time complexity of quick sort is O(nlogn)
The best case for quicksort is O(nlogn)
The worst case for quicksort is O(n^2)





The graphs on the previous page illustrate the number of moves it takes for an array of n size to be sorted. Taking a look at the first graph, we can see that it parallels the time complexity graph for the respective sorting algorithms. According to the graph, quicksort is the fastest sorting algorithm tested followed by shell sort. Bubble sort is extremely inefficient compared to the others tested with an average time complexity of O(n^2). When analyzing the graph of sorting small arrays, arrays with elements of 10 or greater seem to quickly transform into what we would expect from those algorithms. Sorting arrays with less than 10 elements seem to have a very similar amount of moves except for heapsort, which has a very high move count compared to the rest.

Some experiments I ran when testing my different sorting algorithms were inputting large arrays, small arrays, and pre-sorted arrays. The outputs of large arrays and small arrays were expected but the output of an array already in order was interesting. None of the arrays had to move any elements except for heapsort as heapsort arranges the array from highest to lowest then swaps the elements. Bubble sort seemed the most efficient with a pre-sorted array as it had the least amount of compares followed by shell sort and quicksort.

After creating four sorting programs and analyzing their time complexity, moves, comparisons, best, and worst case scenarios, I have learned that there are many different ways to solve a problem, but we should always try to find the most efficient way possible. In terms of these four algorithms it seems like quicksort is the most efficient way to sort numbers in almost all cases.