**Comparison of Insertion Sort, Shell Sort, and Quick Sort**

In this comparison Insertion Sort, Shell Sort, and Quick Sort (median of three) algorithms was ran with unsorted array and almost sorted array. There were five array sizes: 50000, 100000, 200000, 400000, and 800000. Each algorithm ran three times per array size. In each scenario, Shell Sort performed the fasted for both unsorted and almost sorted. For both array types and all the array sizes test, Insertion Sort and Quick Sort performed near exactly the same. Shell Sort is a variation of Insertion Sort. Which, in this case, greatly improved the time to sort arrays of these sizes and types.

Time complexities for each sorting algorithm used:

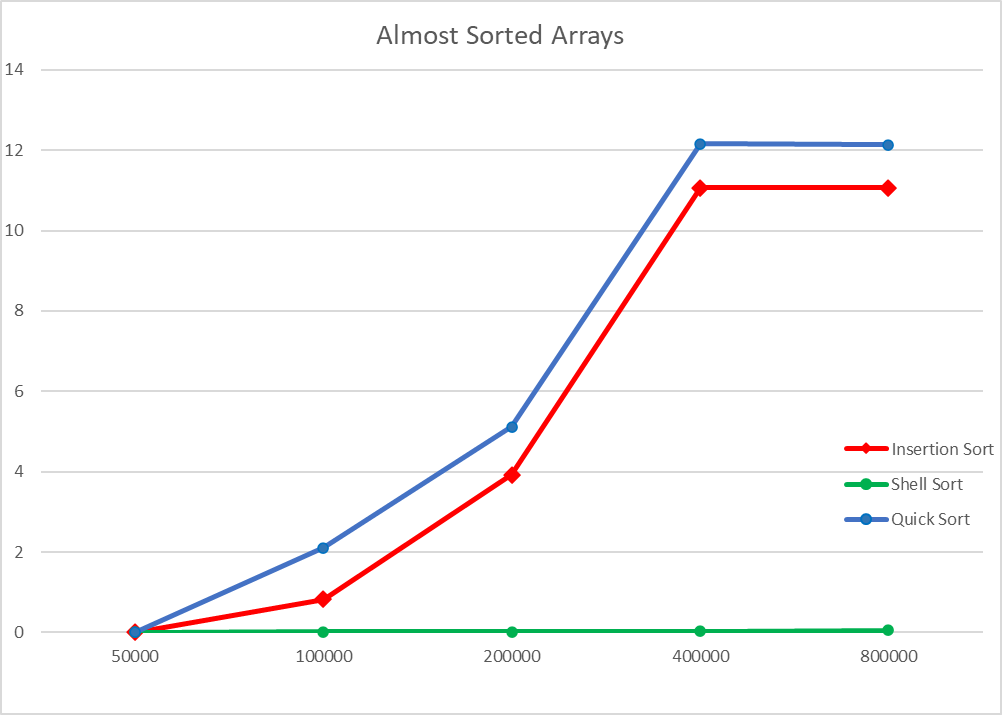
|  |  |  |  |
| --- | --- | --- | --- |
|  | Best | Average | Worst |
| Insertion Sort | Ω(n) | θ(n2) | O(n2) |
| Quick Sort | Ω(n log(n)) | θ(n log(n)) | O(n2) |

|  |  |  |
| --- | --- | --- |
|  | Average | Worst |
| Shell | O(n log(n)) | θ(n2) |
| Hibbard | O(n5/4) | θ(n3/2) |
| Knuth | O(n5/3) | θ(n3/2) |
| Sedgewick | O(n7/6) | O(n4/3) |

Graph representations of the average running times from this project:

Chart, line chart

Description automatically generated



Data Tables for the run times and calculated averages for this project:



Screenshot from one run is shown below.

Text

Description automatically generated