Programming Problem

Purpose:

To understand efficiency analysis including benchmarking and big O.

Description:

This homework is designed to help you begin comparing the efficiencies of different sorting algorithms. Specifically, the purpose of this program is to benchmark and compare bubble sort, insertion sort, mergesort, and quicksort. You will test each sort on an array with (1) randomly generated values (2) in order values and (3) reverse order values. Benchmark the following cases for each:

A) Sorting an array with 1,000 values
B) Sorting an array with 10,000 values
C) Sorting an array with 100,000 values
D) Sorting an array with 1,000,000 values

You will need to use millisecond measurements in order to compare accurately for a small number of values in the array. You will get better results if you run your program on at least two machines. Include information about the machines you use for timings in your write-up.

This homework will require you to research how to perform timings, as well as to produce random values.

From your results, you should create a chart that shows the big O of each algorithm alongside the benchmark timings (use the average timings for all machines for this chart). You should also submit a graphical comparison of the benchmarking results. Finally, you should present an analysis of your results, explaining how the benchmarked timings correspond to the big O analyses.

For this homework, submit everything listed below both electronically and in hardcopy using the same parameters as for the previous assignments.

- Your code
- A table and graphical comparison of your results
- A written analysis of the results including a discussion of when you would choose each sort
- A summary of the stats of the systems on which you run the benchmarks.