EECS 211 – Spring Quarter, 2010
Program 9 (extra credit)

Due Thursday, June 10th, 2010 at 11:59PM

Background:

There are a variety if sorting algorithms that have been developed over the last several decades. You can find many of these sorting algorithms at http://en.wikipedia.org/wiki/Sorting_algorithm. You are to implement 5 different sorting algorithms and test their performance. The 5 algorithms you will investigate are:

1. Bubble sort: http://en.wikipedia.org/wiki/Bubble_sort
2. Merge sort: http://en.wikipedia.org/wiki/Merge_sort
3. Heapsort: http://en.wikipedia.org/wiki/Heapsort
4. Quicksort: http://en.wikipedia.org/wiki/Quicksort
5. Bucket sort: http://en.wikipedia.org/wiki/Bucket_sort

Assignment:

Implement these 5 sorting algorithms, to operate on an array of numbers, where the numbers could be integers, floats, or doubles. Your array should be dynamically allocated at run time, when the number of numbers to sort is determined. You should use templates to make your sort functions general enough so that the same sort function could sort an array of integers, or an array of floats, or an array of doubles. Your program should be general enough to work on an arbitrary size array, up to the amount of memory the particular system has. For each algorithm, you need to sort a variety of different array sizes:

- 1
- 10
- 100
- 1,000
- 10,000
- 100,000
- 1,000,000
- 10,000,000
- 100,000,000
- 1,000,000,000

Each algorithm has a different memory requirement, so some algorithms will run out of memory before others. You are also to measure the elapsed time for each sort operation with at least millisecond granularity. You are plot the results for these 5 algorithms at different scale, along with the expected run-time of each algorithm from the O(n/nlogn/n^2) notation you find at http://en.wikipedia.org/wiki/Sorting_algorithm. Write a brief report that discusses the pros and cons of each algorithm, as well as that presents and explains the timing results. Don't forget to include the hardware specifications (processor, speed, memory, etc) of the machine you used to run the experiments. Please submit both the report and source code via Blackboard.

Comments, suggestions, and hints:

See this link for more details about a similar assignment I found online.
http://www.cs.rpi.edu/~musser/gp/timing.html

Test data:

You are to generate the data randomly prior to each experiment, using uniform distribution. The data generation should not be included in the timing information.