ECE 331
Homework 6

Objective:

1. Compare execution times of Quicksort and Randomized Quicksort algorithms when sorting arrays consisting of elements that are already sorted. The arrays to be sorted shall be the same for each algorithm, in order to make a valid comparison.
2. Compare execution times of the Randomized Quick Sort and Merge Sort algorithms when sorting randomly populated arrays of various sizes. The arrays to be sorted shall be the same for each algorithm, in order to make a valid comparison.
3.

Assignment:

1. Generate c functions to implement both the Quicksort and Randomized Quicksort algorithms using the pseudo-code given in the course textbook as the basis. Do not use library versions of these two algorithms, nor should you just copy this algorithm from an on-line source.
2. In your main program, include a provision to generate arrays of varying sizes. Populate the arrays with integers, using a random number generator function (a library function is fine). Sort the input matrix using any sorting algorithm of your choice and make a copy of the sorted input array so that each can be used as the input to your Quicksort and Randomized Quicksort routines.
3. Use a timing function to measure the execution time for each algorithm to sort the input arrays.
4. Test your sorting algorithms using arrays of size, such as 10, 100, and 1000 elements. Record the execution times and summarize the execution times in a table, ensuring that you provide time units for the execution times, i.e. seconds, milliseconds, clock cycles, etc. For better insight, you should plot the results so that you can compare the two algorithm's order of growth. I recommend you include the data from homework 2 to further enhance your learning about order of growth.
5. Using two similar arrays populated with random numbers, conduct similar sorting execution time performance tests using the Merge Sort and Randomized Quicksort algorithms.
6. Generate a short summary of your experimental observations and based on your results, provide a commentary on when you would use each algorithm. (You may want to conduct more experiments to refine your recommendations.)