**Algorithm Analysis Questions**

1. How do O(n2 ) and O(n log n) sorts compare in actual run times? You will need to implement an effective and correct way to compare the run times of various sorting algorithms

a. Populate a large array with random numbers to effectively compare sorting algorithms. Your array must have at least 1000 elements to adequately test the differences between the fast and slow algorithms. A larger array may give you better results.

b. Capture times for unsorted array data. Please verify that the data you are using for each sort is unsorted by printing the first 5 elements in the array before each sort.

2. How does the order of the data prior to sorting affect the run time of any given sorting algorithm? You will need to measure run time of all sorts when the data is already sorted.

a. Use a large, sorted array to effectively compare sorting algorithms. Your array must have at least 1000 elements to adequately test the differences between the fast and slow algorithms. A larger array may give you better results.

b. Capture times for an already sorted data. Please verify that the data you are using for each sort is sorted by printing the first 5 elements in the array before each sort.

3. You may try different array sizes and multiple trials to obtain a better comparison. Use your creativity and curiosity in implementing your experiment.

Places i found my code for reference

<https://www.programiz.com/dsa/quick-sort>

<https://www.programiz.com/dsa/radix-sort>