1. What is the time complexity of each algorithm? (Include in your answer why the Excel trend-line equation you get from your empirical analysis does not match the textbook’s answer.)

Quick sort: O(x), y=2E-07x-0.0451

Merge sort: O(x2), y = 8E-16x2 + 2E-07x + 0.0005

These answers do not match the text book answers because of the degree of difficulty of getting the same empirical analysis results. The results rely on the data set and size, the computer specifications, code optimizations and many other factors. These differences in the way testing occurred are also what caused the differences in answers.

1. Using the mathematical analysis we performed in class (and that is discussed in the textbook), explain why one algorithm is faster than the other one. Be as specific as you can.

The quick sort is twice as fast as the merge sort. This is because in the mathematical analysis the merge sort ends up with a 2x term while the quick sort has an x term. The reason for this 2x term in the merge sort is because in that sort it has to touch every element twice. Once to sort a smaller array of the set and a second to put it back into the original array.

1. Can you place each algorithm in a big-O classification based on your empirical analysis? Why or why not?

Based on my mathematical analysis I can place each algorithm into a big-O classification. It is strictly an empirical big-O classification and it is also an estimate. Since random data does not put each algorithm through its worst case scenario it is really not as accurate as it could be.

1. What is the largest array that your computer could perform a quick sort on in less than 10 seconds?

n = 67108864 in a time of 9.797155089 seconds.

1. What is the largest array that your computer could perform a merge sort on in less than 10 seconds?

n = 36000000 in a time of 9.824304331 seconds.

1. For you answer to question 5, how much extra memory would you need for the merge sort?

An integer takes up 4 bytes so 36000000 \* 4 = 144000000 bytes or 137 megabytes extra.