1. The running time for Merge Sort was .032, whereas the runtime for Quick Sort was .112. The best case for both Merge Sort and Quick sort is N(Log)N, whereas the worst case for Merge Sort is N(Log)N and the worst case for Quick Sort is (N^2). Merge Sort is faster in this case due to the data set. Quick Sort is slower for this case due to a poor pivot point, while Merge Sort is more consistent. Quick Sort is better in memory, due to Merge Sort not being in place, and having to create temporary arrays over and over.

Merge Sort:

[16 5 11 2 1 15 8 14 7 12 3 9 4 6 13 10]

[16 5 11 2 1 15 8 14] [7 12 3 9 4 6 13 10]

[16 5 11 2] [1 15 8 14] [7 12 3 9] [4 6 13 10]

[16 5] [11 2] [1 15] [8 14] [7 12] [3 9] [4 6] [13 10]

[16] [5] [11] [2] [1] [15] [8] [14] [7] [12] [3] [9] [4] [6] [13] [10]

[5 16] [2 11] [1 15] [8 14] [7 12] [3 9] [4 6] [10 13]

[2 5 11 16] [1 8 14 15] [3 7 9 12] [4 6 10 13]

[1 2 5 8 11 14 15 16] [3 4 6 7 9 10 13]

[1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16]

Quick Sort:

[16 5 11 2 1 15 8 14 7 12 3 9 4 6 13 10]

[5 2 1 8 7 3 9 4 6] [10] [16 11 15 14 12 13]

[5 2 1 3 4] [6] [8 7 9] [10] [11 12] [13] [16 15 14]

[2 1 3] [4] [5] [6] [8 7] [9] [10] [11] [12] [13] [14] [16 15]

[2 1] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [14] [15] [16]

[1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [14] [15] [16]

[1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16]