CS 344 Project 1:

Damon Gwinn & Eric Sognefest

**Introduction and overview**

The project was to implement Selection Sort, Insertion Sort, Merge Sort, Quick sort where the fist element is the pivot, A randomized quick sort, and a Randomized quick sort using the Partition in place algorithm, and compare the results to each other and the STL sort algorithm, using the number of operations as a baseline for comparison, we also decided to time the algorithm using the system time from the start and end of the function call and converting into milli seconds.

**Comparison Table**

Arrays are of size N = 10,000 and the random values in the array are of Range 1-500.

The data in the table are based on a single run of all the algorithms. Arrays are created as per instructions. for each sorting algorithm the number of operations is displaced followed by the running time in milli seconds. All of the data was taken from a single apple computer.

|  | SS | IS | MS | DQS | RQS | RQSIP | STL |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Sorted Array | 50044996 | 39997 | 341840 | 880876 | 740020 | 409332 | 70006 |
| Time (ms) | 270.052 | 0.211886 | 3.40566 | 5.55894 | 5.15449 | 2.11615 | 0.320356 |
| Reversed Array | 50044996 | 100019998 | 346240 | 776182 | 783169 | 466900 | 145093 |
| Time (ms) | 263.638 | 441.422 | 3.43302 | 5.11717 | 6.12773 | 2.41741 | 0.6801 |
| Random Array | 50044996 | 50023168 | 397701 | 905107 | 904045 | 458468 | 441208 |
| Time (ms) | 264.08 | 214.334 | 4.41032 | 6.71306 | 6.7094 | 2.78934 | 2.62921 |

**Analysis**

1. The smallest number of operation was 39,997 form the sorted array on insertion sort, as expected. It also had the shortest running time at 0.211886 ms.
2. taking T = 39,997 the expected operations on the reverse array is T^2/2 = 799,880,004.5 compared to our data 100,019,998 which yields a large percentage of error. However the operations is on the same order as the expected result. running time was at 441.422 ms.

C) On random arrays the expected operations was to be T^2/4 = 399,940,002.3 half the expected number of operations from part B , compared to our actual results of 50,023,168 which is roughly half that of our actually results from part B, this showed that there may be an error in the counting operations, but it is exhibition the trends that are expected from insertion sort. Or our analysis of the expected results was over pessimistic, however out data is off by a constant value, which is to be expected.

D) The expected output of selection sort is T^2/2 = 799,880,004.5. compared to out results 50,044,996 operations for every array, which is on the order of T^2/4. Which is faster than expected. All the running Times this algorithm were approximately 264 ms.

E) our analysis of merge sort yielded approximately 346,240 operations for every array and a running time of around 3.5 ms. which according to the expecting number of operations was to be 159,988 which is again off by some constant factor.

F) Deterministic quick sort for all arrays was around 850,000 operating and about 5.5 ms in running time for all arrays.

G) the randomized quick sort roughly halved the running time at aprox: 460,000 operations one all arrays at a running time of around 2.5 ms.

H) as expected the randomized version of quick sort in place rand with only 10,00 operations difference and a running time difference of 0.1 ms.

Our data follows most of the trends expected from out analysis, however our data was off by some constant factor. the exception of selection sort which exhibited T^2/4 number of operations instead of T^2/2.