# *Project 1: CSC 430*

## *Summary and Table of the Test Results*

Erii Sugimoto

InsertionSort is at worst $O(n^2)$ which happens when the list is inversely ordered. When the list is almost sorted, insertion sort is very efficient as both the number of comparison and swap are significantly smaller than other data types.

SelectionSort's performance stays the same $O(n^2)$ regardless of different datasets. The number of comparison is always $\frac{n(n-1)}{2}$ and the number of swap is always $n-1$.
MergeSort is always $O(nlog_2n)$ regardless of the order the data is stored. It does not swap. Although the number of comparison appears the smallest in the result below, it is important to remember that MergeSort is space-inefficient in array implementation.
QuickSort is at worst $O(n^2)$ when the data is inversely ordered (or ordered) because it splits the list extremely unbalanced; on the other hand, when the data is sorted in such a way that it divides the list evenly in half every recursive call, it is $O(nlog_2n)$. For this reason, Quick Sort does much more efficient on random data than sorted data.
HeapSort's number of comparison differs slightly depending on how the data is sorted; specifically, the smaller the item is in the root, the more comparison it requires; however, it is $O(nlog_2n)$ and does not vary much amongst different data and is not space-inefficient.
BubbleSort is always $O(n^2)$ unless the data is sorted. The number of swapping varies greatly depending how the data is sorted.
From this testing, I have gained a better understanding of which sorting algorithms should be used depending on how the original data is sorted and the size of the data.

| | Inverse 100 | | Inverse 1000 | | Random 100 | | Random 1000 | | Almost 100 | | Almost 1000 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | compare | swap | compare | swap | compare | swap | compare | swap | compare | swap | compare | swap |
| InsertionSort | 4950 | 4950 | 499500 | 499500 | 2409 | 2312 | 253879 | 252886 | 342 | 243 | 6235 | 5736 |
| SelectionSort | 4950 | 99 | 499500 | 999 | 4950 | 99 | 499500 | 999 | 4950 | 99 | 499500 | 999 |
| MergeSort | 672 | 0 | 9976 | 0 | 672 | 0 | 9976 | 0 | 672 | 0 | 9976 | 0 |
| QuickSort | 4950 | 2599 | 499500 | 250999 | 596 | 354 | 11197 | 5585 | 2348 | 202 | 116104 | 2802 |
| HeapSort | 962 | 518 | 15948 | 8304 | 1054 | 593 | 16868 | 9097 | 1088 | 630 | 17276 | 9507 |
| BubbleSort | 4851 | 4851 | 498501 | 498501 | 4851 | 2271 | 498501 | 252845 | 4851 | 243 | 498501 | 5736 |