COMPSCI 2XC3 Lab Report 1

Version 1.0

Prepared by

Group 64

Luca Mawyin

Anderson Ray

Theo Pham

COMPSCI 2ME3

McMaster University

January 30, 2026

Experiment 1

In our experiment comparing Bubble Sort, Insertion Sort, and Selection Sort. We ran 100 tests for each sorting algorithm going from a list length of 0, to a list length of 1000 each list length being 10 elements longer then the last.

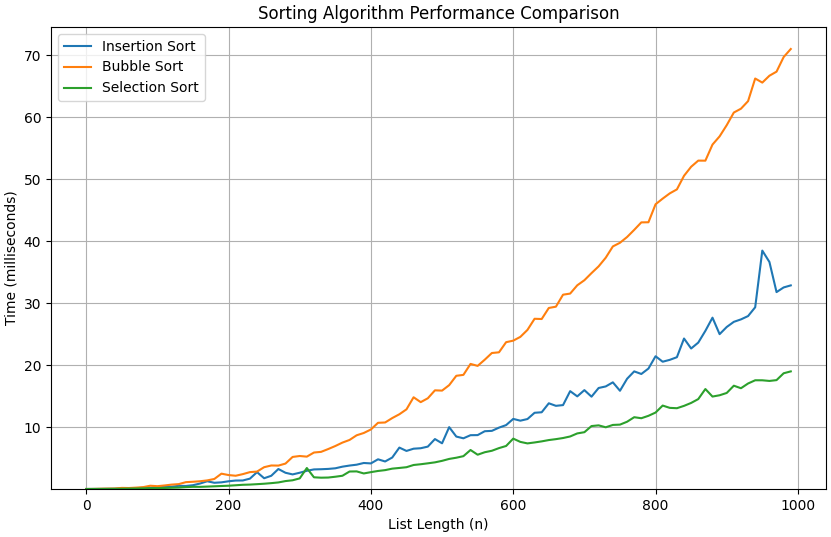

Figure 1: List Length vs. Time (ms) for Bubble, Selection, and Insertion Sort.

Looking at the slopes, each algorithm looks to have a parabolic shape which makes sense as we know that the algorithms used are $O(n^2)$.

Bubble sort is the slowest as the inner loop, loops through the entire list every iteration of the outer loop. Selection sort is faster then Insertion sort as swaps elements in the list smarter using less memory reads and writes making it faster.

Experiment 2

Experiment 3

In our experiment comparing Bubble Sort, Insertion Sort, and Selection Sort on sorted lists with varying numbers of swaps made. We ran 301 tests for each sorting algorithm on lists of size 2000 with the number of swaps ranging from 0 - 30179. Each test would increase the number of swaps made by 100.

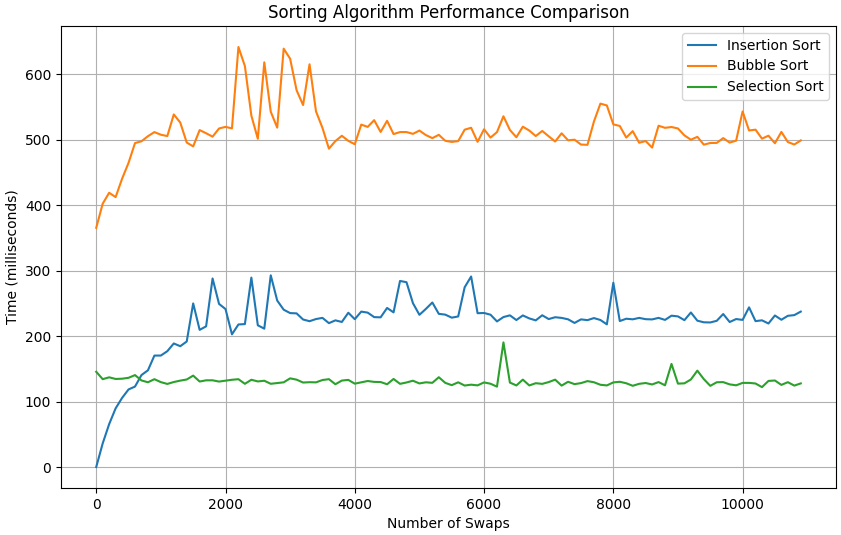

Figure 2: List Length vs. Number of Swaps on a Sorted List for Bubble, Selection, and Insertion Sort.

Looking at the graph, Bubble Sort and Insertion Sort preform better with less swaps. Bubble Sort preforms better with less swaps, as its doing much less memory reads and writes in the innerloop. Insertion swap preforms better with less swaps as it exits out the second loop if it runs into sorted pairs of elements which are more common with less swaps.

Selection Sort doesn't perform better with less swaps as the number of checks and swaps are independent of whether the list is sorted or not. Even if the innerloop can't find a min index past L[i] it still swaps L[i] with itself.

Experiment 4

Experiment 5

Experiment 6

Experiment 7

Experiment 8