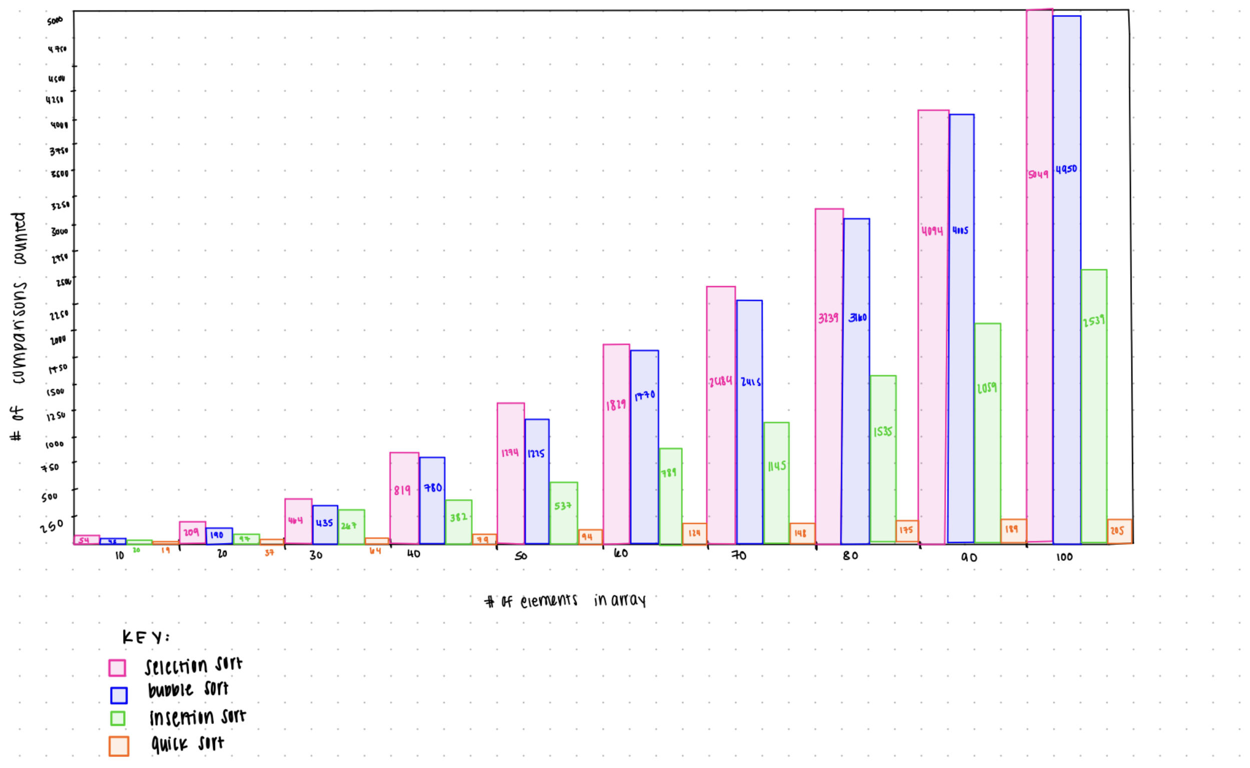
In-Class Exercise 3

The program first assumes that the smallest element is at index 0, and then checks each element after 0 to see if it is smaller. 51 is originally selected as the smallest element, and as the loop traverses the array, the smallest element 24 is selected. 51 and 24 swap places with each other because 24 is the smallest element. The for loop then searches again for the next smallest element, 32, which is located in the last index of the array. 51 (which is now at index 1) is swapped with 32 (which is at the final index), which places 51 at the last position in the array.

In-Class Exercise 5-6



In-Class Exercise 7

Selection sort takes 36 milliseconds, and bubble sort takes around 35 milliseconds, so the two methods take up roughly the same amount of time. Quick sort takes 3 milliseconds, so it is the fastest choice by far. When I ran the SortPerformance program, insertion sort was taking between 0-3 milliseconds, so I think that my program may have bugs, because it doesn't make sense for insertion sort to perform more quickly than quick sort, especially on an array of such a large size.