

Logan Kelsch

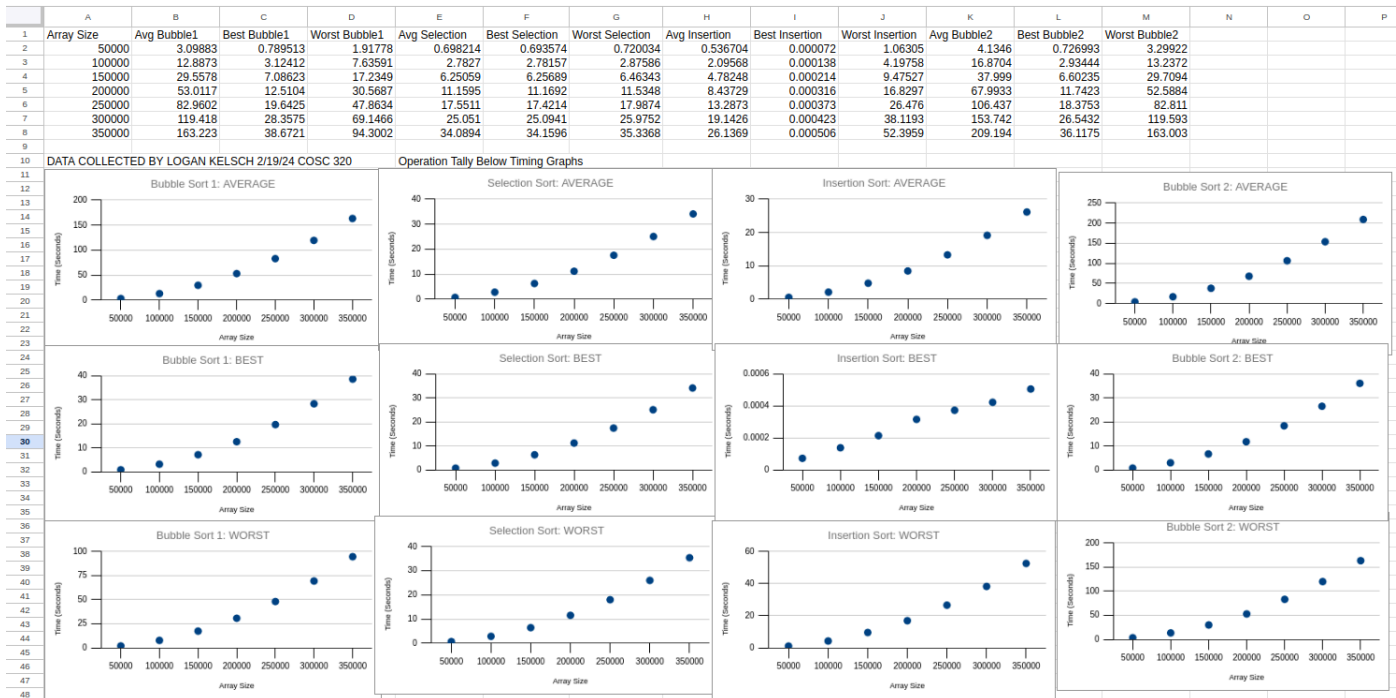
2/21/2024

COSC 320 MW 9:00am

Homework #2 - Report

ALL DATA COLLECTION:

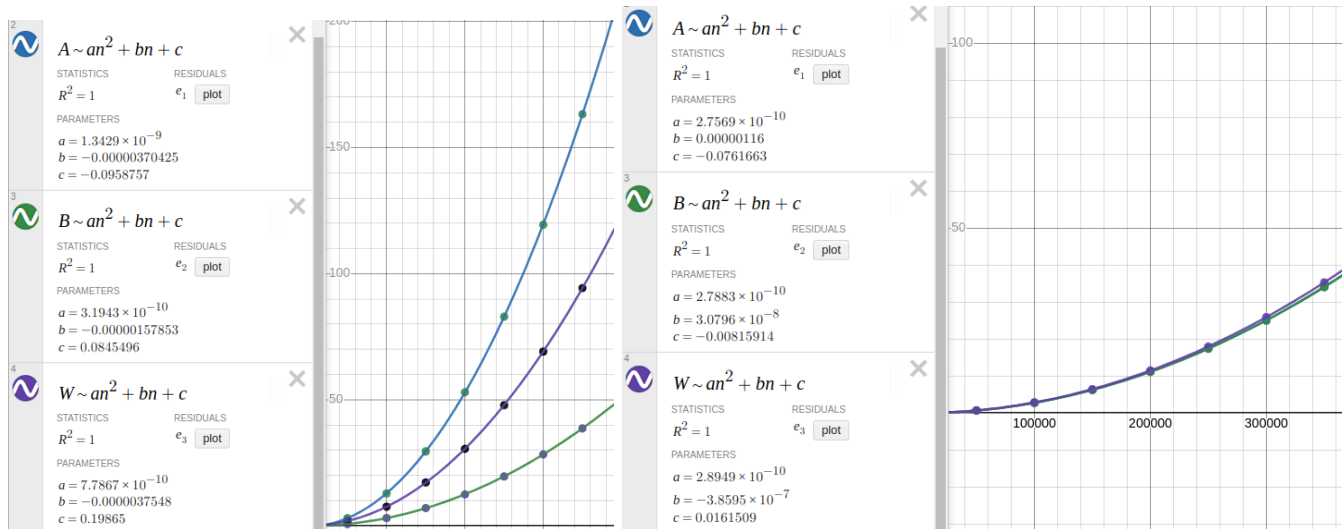
All Timing data and graphs:



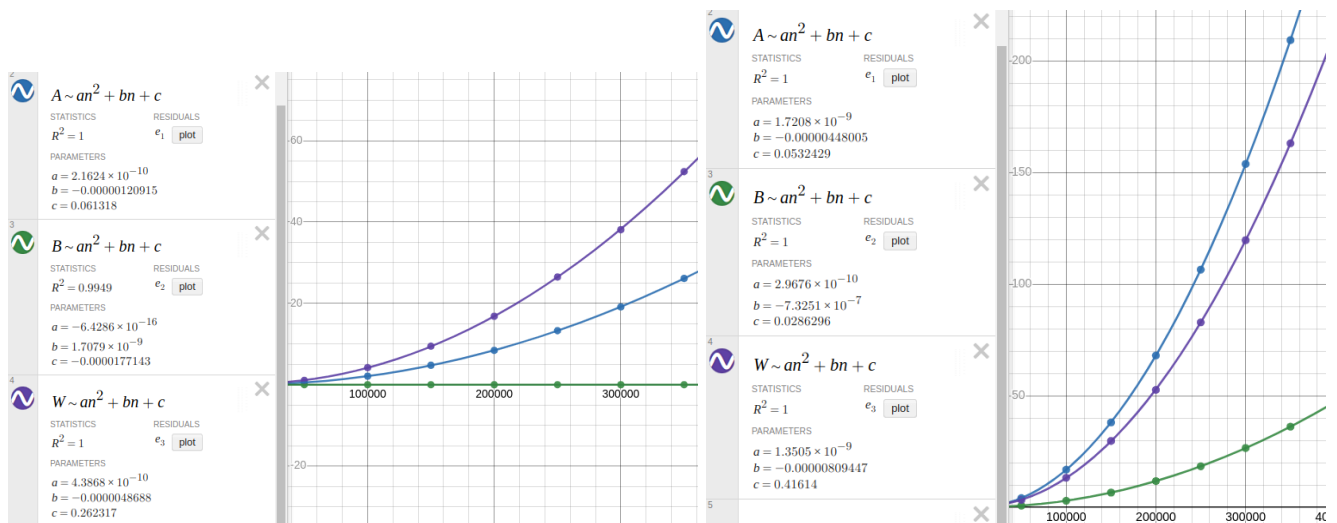
All best-fit lines (VIA DESMOS):

A-Average, B-Best, W-Worst

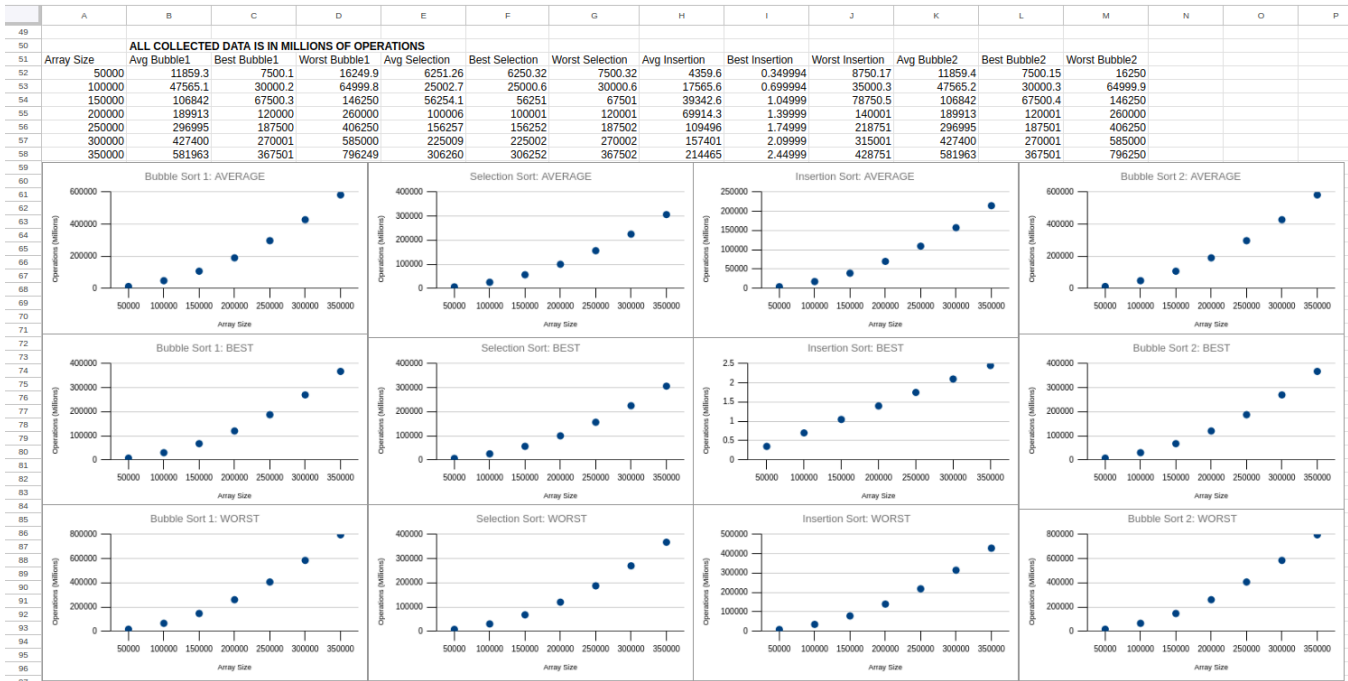
Bubble Sort(LEFT), Selection Sort(RIGHT):



Insertion Sort(LEFT), Bubble Sort 2 (RIGHT):



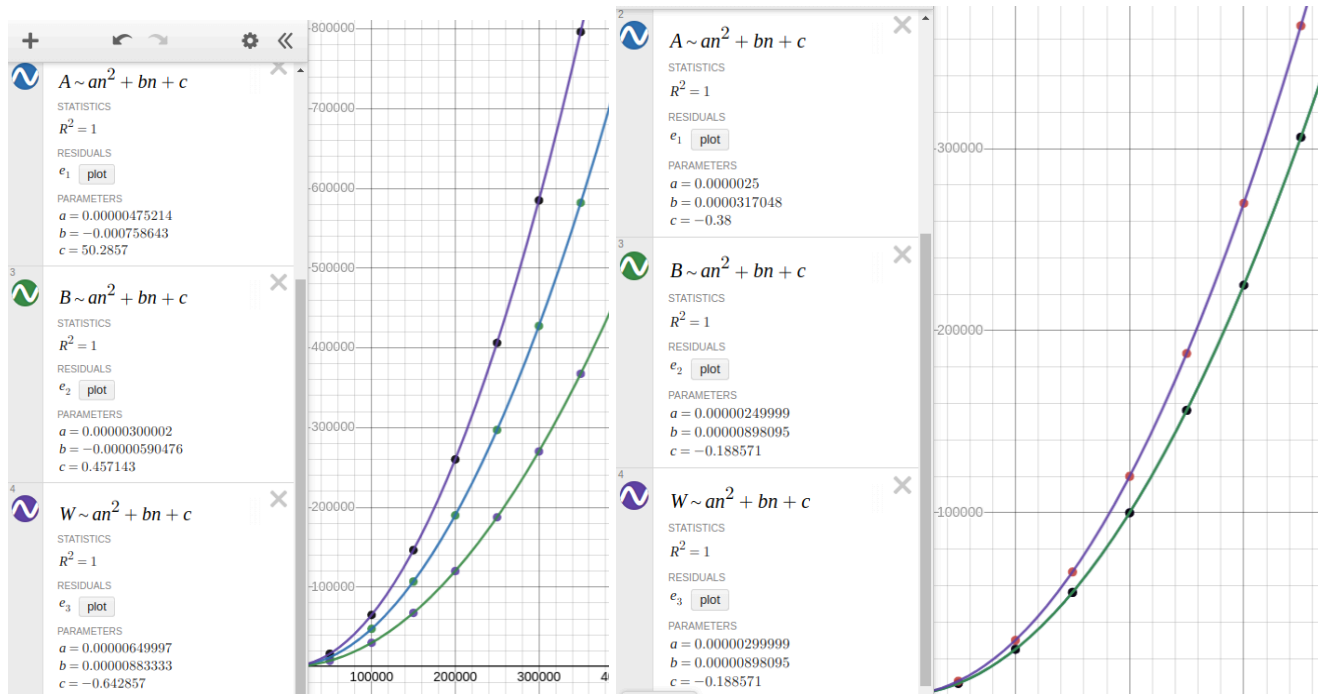
All operation count and data graphs:



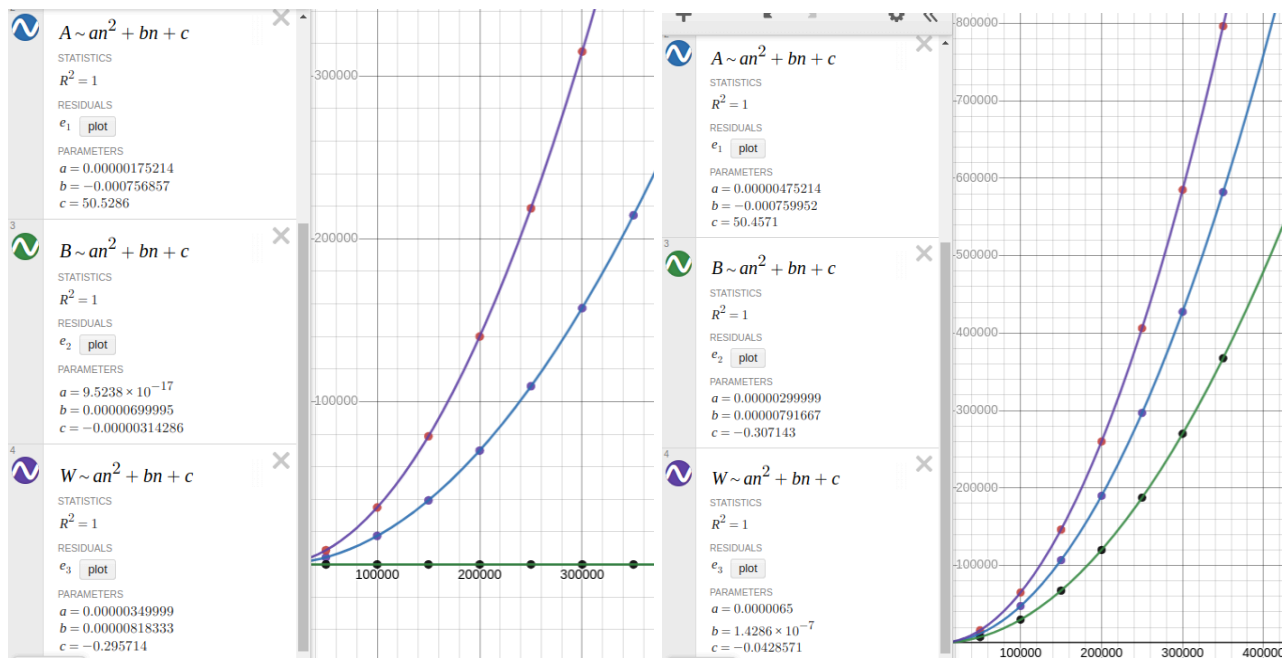
All best-fit lines (VIA DESMOS):

A-Average, B-Best, W-Worst

Bubble Sort(LEFT), Selection Sort(RIGHT):



Insertion Sort(LEFT), Bubble Sort 2(RIGHT):



1. According to part one, as well as what is clearly visually represented in the graphs, all scenarios and all sorts had the same complexity of  $O(n^2)$ , except for Insertion Sort's Best Case Scenario, which had a complexity of  $O(n)$ .

2. Fastest to slowest:

Best Case Scenarios:

Insertion, Selection, Bubble 2, Bubble sort

Worst Case Scenarios:

Selection, Insertion, Bubble, Bubble 2

Average Case Scenarios:

Insertion, Selection, Bubble, Bubble 2

3. The difference of the coefficients produced from Desmos line fitting equations and the ones derived from section one was around x100, meaning that they were not relatively close. This is most likely a product of the computer operations being understood and read differently than the way we were instructed to approach this concept in our analyses.
4. Unfortunately, I had turned in my written homework before recognizing that it was required in this section of the homework. Although I would like to confirm my analysis is beyond perfection and must certainly be parallel to the analysis in part two of this report.
5. After doing significant testing and analysis of a few sorting options, it is in the company's best interest to stick to the Insertion sort algorithm, as it consistently sorts faster than the other options explored. Under no circumstance should the bubble or bubble 2 sort algorithms be used within the company's programming. If we will be re-sorting partially sorted data sets, insertion sort remains the best option. In the situation where you are to be without the capacity to use Insertion sort, selection sort is the next best option, running nearly as fast. There are other potentially faster algorithms in which we can explore in the near future if required, but for the time being, insertion sort will be more than sufficient. Please inform me of any questions regarding the data collection or analysis, as I will be happy to clarify.