

Project 1: Sorting Report

The structure of my program takes the sequences based on the size and puts both sequences in their own array. I made two functions `int *seq_1` and `int *seq_2` in order to do this. The improved bubble sort would call `*seq_2` and the shell sort would call `int *seq_2` and have both sequences put in a array. For loading and saving I used `fscanf` and `fprintf` as my main file commands to parse and output long integers. Within my shell sort and bubble sort I implemented algorithms from the lecture notes and then added the gap sequence to it in an outer loop. I did not perform any additional optimizations.

The time complexity of both sequences is $O(n)$ this is because each sequence only has one loop each. The time complexity of the shell and bubble sort for average and worst cases is $O(n^2)$ and the space complexity is $O(1)$ since both sorting algorithms do not re-size the array.

Insertion Sort

	1000.txt	10000.txt	100000.txt	1000000.txt
Time (s)	0	0.01	0.09	1.12
Comparisons	35266	615529	9484124	135697411
Moves	12933	194454	2636139	35130780

Bubble Sort

	1000.txt	10000.txt	100000.txt	1000000.txt
Time (s)	0	0.01	0.08	0.98
Comparisons	54903	949233	14113444	191772106
Moves	13455	189528	2448540	30237996

The running time grows somewhat linearly with size of the array. The main factors that determine the running time however are the comparisons and moves. This will affect the time linearly.

The additional memory required by my routines was done in my sequence generator functions. My functions malloced an array of size n to make sure it was big enough to contain the entire sequence. I also reversed my array in shell sort for simplicity which definitely added more time to the total running time.