Megh Dholakia
Project #1
ECE368

Running the Code: These are the results of running both the Shell Insertion Sort and the Shell Selection Sort. The Insertion sort run a lot faster than the Selection sort and the selection sort does not work for 1000000.txt. Both the number of comparisons and the number of moves grow rapidly as you increase the number of inputs.

Shell Insertion Sort

1000.txt
Number of Comparisons: 2.9700000e+04
Number of moves: 5.460000e04
I/O Time: 0.00000e+00
Sort Time: 0.0000e+00

10000.txt
Number of Comparisons: 2.9700000e+04
Number of moves: 5.460000e04
I/O Time: 0.00000e+00
Sort Time: 0.0000e+00

100000.txt
Number of Comparisons: 7.310693e+06
Number of moves: 1.364907e+07
I/O Time: 0.00000e+00
Sort Time: 1.000000e+00

100000.txt
Number of Comparisons: 7.3100000e+06
Number of moves: 1.3640000e07
I/O Time: 0.00000e+00
Sort Time: 0.0000e+00

Shell Selection Sort

1000.txt
Number of Comparisons: 1.463488e+04
Number of moves: 1.422900e+04
I/O Time: 0.00000e+00
Sort Time: 0.0000e+00

10000.txt
Number of Comparisons: 1.491996e+08
Number of moves: 2.107080e+05
I/O Time: 0.00000e+00
Sort Time: 1.0000e+01

100000.txt
Number of Comparisons: 1.4982216e+10
Number of moves: 2.911654e+06
I/O Time: 0.00000e+00
Sort Time: 5.100000e+01

1000000.txt
Number of Comparisons: N/A
Number of moves: N/A
I/O Time: N/A
Sort Time: N/A

Time Complexity: The time complexity of the insertion sort was close to n^2(long(n)) after comparing the results across the various inputs. The time complexity of selection sort was far larger since it compares more indexes than the insertion sort.

Sorting Space Complexity: Since my sort function only allocates memory for one array my sort function has a space complexity of O(n).

Sequence Time Complexity: The time complexity for my sequence generate is nlog(n)^2 based on the method of deriving the sequence from 2^p*3^q.

Sequence Space Complexity: The space complexity for my sequence generation is O(n^2) because it uses nested for loops and an allocated array.