Max Chi
ECE 368: Project 1

Description of Algorithms

The Shell_Insertion_Sort (SHS) I have implemented into the project works by generating a sequence of gaps. The gaps are generated by running a nested loop and incrementing p and q by one each time through. Each time it is run through, the value 2^p * 3^q is evaluated for every p and q value until the total product exceeds the range of the integers required to be sorted. After the gap sequence is completely generated, the algorithm proceeds to run through the unsorted array with an offset of the gap, incrementing both the gap and the initial index until the gap exceeds that of the anticipated size. Then the following gap in the generated gap sequence replaces the current gap value, continuing to do so until the array is completely sorted.

The Improved Bubble Sort (IBS) works in a similar fashion as the above SHS, in that it first generates a gap sequence. The gaps are obtained by dividing the number of integers continually by 1.3 until reaching 1. In the case of a gap of 9 or 10, the values are converted to 11. These gaps the comb through the unsorted arrays until gap reaches 1, where the array will be fully sorted.

Analysis of Time- and Space- Complexity

SHS's space complexity is *O(log(N*N))*, since the size of the array is determined by the amount of times N is divided by 2 and summing the integers from 1 to the last value. The time complexity is *O(N*log(N*N))* due to the nested loops.

IBS's space complexity is *O(log(N))* since the size of the array is the number of times 1.3 is raised to a power while it is still less than N. IBS's time complexity is *O(log(N))* since the loop only iterates as many times as there are sequences.

Data Analysis

Shell Insertion Sort

Input Size	Run Time (s)	Comparisons	Moves
1000	0.00	30752	65786
10000	0.00	547099	1158590
100000	0.52	8548969	17970887
1000000	0.66	123987151	249974302

Improved Bubble Sort

Input Size	Run Time (s)	Comparisons	Moves
1000	0.00	21704	13197
10000	0.00	306727	186408
100000	0.00	3966742	2438928
1000000	0.31	49666762	30144183

Comparing the run times between the Shell Insertion Sort (SIS) and the Improved Bubble Sort (IBS), it can be concluded that the Improved Bubble Sort was much faster than the other sorting algorithm. However, in terms of moves and comparisons, IBS increased by a factor of 1 for each input. While the SHS actually carries out moves and comparisons at a faster rate than that of the IBS.