

ECE 368 – Data Structures

Project 1 - Nausherwan Korai

Algorithm and Structure

2 Algorithms were written, and both utilized shell sort.

Shell_Insertion_Sort creates a gap sequence to the note of $2^p * 3^q \dots 16, 12, 9, 8, 6, 4, 3, 2, 1$. Once this gap sequence is created, an insertion sort is run within sub-arrays of the length of each gap sequence.

Improved_Bubble_Sort creates a gap sequence to the tune of $N/1.3, N/(1.3)^2, \dots, 1$. Once this gap sequence is created, a bubble sort is run within sub-arrays of the length of each gap sequence.

All the gap sequences are created using iteration and not recursion.

Time Complexity & Space Complexity of gap sequences

Seq1

Space Complexity: $O(N)$

Time Complexity: $O(1)$

Seq2

Space Complexity: $O(N)$

Time Complexity: $O(N)$

Tables and analysis

Shell_Insertion_Sort

Data set	1000	10000
Run-time	0 s	0 s
No. of Comparisons	4311	64818
No. of moves	30955	550711

Improved_Bubble_Sort

Data set	1000	10000
Run-time	0.07 s	5.55 s
No. of Comparisons	12560491	1706971854
No. of moves	747345	0

As data set goes higher, shell insertion sort stays constant or grows at a slower rate compared to shell bubble sort.

Space Complexity

Shell_Insertion_Sort $O(n)$

Improved_Bubble_Sort $O(n^2)$