Aleesha Nageer
anageer@ucsc.edu
11/15/20

CSE 13S: Fall 2020

Assignment 5: Sorting
Writeup

In this assignment, we implemented four different sorting algorithms, Binary Insertion Sort, Quick Sort, Shell Sort, and Bubble Sort to put values in an array in ascending order. The purpose of this lab was to demonstrate that each of these sorting algorithms have different time complexities and efficiencies. In addition to understanding they have different complexities, we also learned how different ways of implementing these algorithms could make their time complexities vary. Each sorting function has a best, worst and average time complexity:

	Best Time Complexity	Worst Time Complexity	Average Time Complexity
Binary Insertion Sort	O(nlogn)	O(n^2)	O(n^2)
Quick Sort	O(nlogn)	O(n^2)	O(nlogn)
Shell Sort	O(nlogn)	O(n^2)	O(nlogn)
Bubble Sort	O(n)	O(n^2)	O(n^2)

What I learned from each of the sorting algorithms is that the divide and conquer technique will boost the efficiency of the function overall. Going through each index and comparing them one by one, like Bubble Sort does, takes a lot of computation and will be the most extensive way of sorting an algorithm. By implementing gaps, pivot points, and combining two sorting algorithms into one, the time complexity improved a substantial amount.

All four of my sorting algorithms didn't work initially so I had to play around with the pseudocode provided to get each of them to properly sort. This helped me improve my skills with programming in C because I personally struggle with the syntax and specifications needed for arrays from time to time. Python has dynamic arrays so working with the pseudocode to make the algorithms work in C was tricky. This assignment allowed me to mess with decrementing and incrementing the array size for certain conditional statements in my sorting functions, work with temporary variables for swapping, and understand how going through the different values at different indexes can make an immense difference for the overall efficiency of the program.