Jiacheng Yuan

ECE 368

Sep 29. 2015

Project 1

Milestone 1

The goal of the project is to work with Shell sorting and to improve the performance of the program. The shell sorting and bubble sorting needs to be improved by using "gap" strategy. The project require the sequence $\{2^{p'}3^{q'}, ..., 2^{p}3^{q}, ..., 16, 12, 9, 8, 6, 4, 3, 2, 1\}$ for implementing Shell sort and using the sequence $\{N/1.3, N/(1.3)^2, N/(1.3)^3, \ldots, 1\}$ to improve bubble sort.

There are few functions that needs to be write. The first function that needs to be write is the *long *Load_File (char *Filename, int *Size)*. It loads the file with a given filename and number of integers that needs to be sorted. The function will return a long integer of the subsequent lines which will become the array that needs to be sorted in later use. The second function that needs to be write is the *int Save_File (char *Filename, long *Array, int Size)*. It saves the *Array* to a file specified by the given *Filename*. The function will return the count of the integers that has been saved. Those two functions could simply be manipulated by using *fopen* and pointers.

The two function that mainly needs to be wrote are *void Shell_Insertion_Sort (long *Array, int Size, double *NComp, double *NMove)* and *void Imporved_Bubble_Sort (long *Array, int Size, double *NComp, double *NMove).* These two functions sort the given array in ascending order with the given size of the array. It also given the number of comparisons and the number of moves. And two functions needs to use their given sequence.

The two sequence are also need to be generated by writing functions. The function void Save_Seq1 (char *Filename, int N) and the function void Save_Seq2 (char *Filename, int N) are the two functions that to generate the sequence for shell sorting and bubble sorting. The sequence can be generated by using *for loop* and simply manipulating numbers and save in the array.

After finishing all the functions, there needs to debug the program and organize the structure to optimize the program in order to improve the performance.