Lince Rumainum

Student ID: 112722337

CS 2413 Data Structures

Project 1

For Project 1, to read in the input, I use Big O (n) algorithm. The program prompts the user for input file(s) and read through each line of data, while only storing the data without invalid value. After storing the data, then it will do linear search of the duplicates (record ID) within the file and then the previous record ID data from previous file (if any). It will store the newest record ID if they are from two different files while reject any duplicate after the one that already stored when it is from the same file. The search time use in the program is *2n* since it's doing a linear search of each data that is in the current file and then it does it again on the data from previous file(s). For the insertion time of the program, it does it for *n* time complexity because it will read the file one line at time for however many lines that are in the file and insert the data if there is no invalid value. The total time of this approach ended being a *n + 2n* which is *3n* or Big O runtime complexity of *n*.

An alternative approach for this will be having a sorted list. When having one would be able to use the binary search of that sorted list to search and insert data. Because of that, the search time will become a *log n* time complexity while the insertion time will be a *n log n*. So, the total time of the alternative approach will be *log n + n log n* which is Big O runtime complexity of *n log n*.

In this program the merging of data across files is done with a time complexity of n^2 since it is reading each line of data from each file, storing it (the one without invalid data) and then going through each line of data to find it's duplicate. The sorting of data use for this program is a quicksort (where the midpoint is the pivot point of the array) which have an *n log n* time complexity. The space complexity used to merge the data is approximately *3n* since it double in size every time the array is full

The binary search for this program uses the same regular approach with the modification of when the item is found, then the program will go backward to previous index one at a time to see if it is the first index that the data value is in the array. The total time used to use for the binary search at Big O time complexity would be *n/2* which is technically *n* runtime.