CS 499 Milestone Three: Enhancement Two

Jeffrey Tung

Southern New Hampshire University

The artifact included for enhancement two is the vector sorting program from the CS 260 course. This artifact was originally created on February 15, 2020, during week four of the CS 260 course. It is a C++ sorting program which provides the user with a menu of functionalities to choose from. The program is designed to load in an excel csv file full of bids and translate each bid entry into my created vector structure within the program. Once the bids are loaded in, user can choose to display the bids, selection sort the bids, or quicksort the bids.

This artifact showcases my understanding of data structures and algorithm through my implementation of the two different sorting methods. Selection sort is a sequential sort, usually good for sorting smaller databases. Quicksort uses partitions and is harder to implement but has much better performance especially for larger databases. Having both sorting methods implemented along with the performance of each method displayed, users can see how much more effective quicksort is for sorting larger databases than selection sort. For comparison, using selection sort for a total of 17937 bids takes a total of 5.785 seconds to sort while quicksort takes only 0.013 seconds to sort the same number of bids.

My improvements for this artifact further enhance the menu options for my two sorting methods as well as giving users an option to limit the number of bids display outputs. For the menu options, I was able to successfully provide users embedded menus of sorting by ID, title, fund, or amount within each sorting methods. For the bids display output limit, I created a case 5 to my menu system and a new variable of user inputted displayLimit. I also created an error handling while loop so the user inputted displayLimit must be less than the number of bids loaded in. Overall, I was able to successfully implement both of my planned enhancements from module one.

My process of enhancing my artifact went pretty smooth as the most difficult part was re-familiarizing myself with the program 9 months after it was first programmed. Since I had initially coded this program with modularity, with each function taking care of an individual function request from the menu, I was able to quickly rekindle myself with what each function did within the program as well as apply the appropriate enhancements to obtain my desired results. I had to create multiple instances within my selection sort and my quicksort function to change between the different user inputted variables to sort by.