**Danny Forte**

**CS-300 Analysis and Design**

**March 16, 2025**

Our assignment was to complete the code within the VectorSorting.cpp file to allow it to sort through a collection of bids stored in a vector. We will be using two algorithms that will allow us to sort these bids using min, mid, and max values. One method we will use is called selection sort, this method compares a min value with the next value in the list. If the next value is lower than the current min value, it will assign a new min value as they will swap places in the list. This process repeats itself until all items within the list have been compared and we are left with a list that is now in ascending order. The second method we will be using is a quick sort method. This method breaks our list into two halves, low and high. It uses a pivot value which is the middle element. It then compares each item to the pivot value and if its lower in value then it goes onto the left side of the pivot, if its higher it moves to the right side of the pivot. This continues until all items have been compared and sorted. This is sometimes called a divide and conquer approach as we are breaking our list it into smaller parts and then recombining our list back together after we have sorted it. This when completed also leaves us with a sorted list of ascending values.

I had many challenges with this assignment. My first one was with understanding the quick sort concept and its pivot. What made it hard for me during the zybooks activities was my rounding up mindset when determining the pivot values when faced with an odd number of elements. Once I worked past that I really like this model of sorting and can think about how many times I have used this approach myself when doing things such as sorting a deck of cards. My other challenge is still with my IDE's. I always tend to run into issues with either compiling or debugging my code. This has prompted me to use many different IDE's such as VS, Eclipse, IntelliJ and PyCharm. For this assignment after cleaning up some syntax errors and typos, I was not getting any other errors and seeing build successful, but when I go to run the file I was seeing exe could not be found. This then leads me to issues with directories issues, I have 7 local drives, and to an issue with my Antivirus program interfering with the creation of those needed files.

**Pseudocode**

While choice IS NOT equal to 9

Main

Display user menu:

1. Load Bids

2. Display All Bids

3. Selection Sort All Bids

4. Quick sort All Bids

9. Exit

Ask for User input and store choice for menu selection

If Choice = 1. Load Bids

Start Clock and store ticks

Call loadBids to open and store CSV data

Output number of records found in CSV file

Stop the clock

Display time needed to read CSV file

Break

If Choice = 2. Display All Bids

Loop through bids vector

Display Bids

Break

If Choice = 3. Selection Sort All Bids

Start Clock and store ticks

Call selectionSort to pass bids vector

Stop the clock

Display time needed to sort the vector

Break

If Choice = 4. Quick sort All Bids

Start clock and store ticks

Call quickSort to pass bids vector

Stock the clock

Display time needed to sort the vector

Break

If Choice = 9. Exit

Exit application

Display Goodbye

End

selectionSort

Get vector to sort

set min to 0

set current index to smallest index

Loop from current index through vector

Compare current index value to value of smallest index

if value is less than value of smallest index

make smallest index equal to current index

Swap the vector item at smallest index with item at the outer loop position

End

quickSort

Get vector to sort, lowest index of vector and highest index of vector

If lowest index is greater than or equal to highest index return nothing

Call partition function

Set low Index equal to begin

Recursively call quickSort to pass the vector, lowest

Recursively call quickSort to pass the vector, low Index (from above) plus one, and highest index

End

partition

Have the vector partition, the lowest index and the highest index

Determine the midpoint between the lowest and highest index

Set pivot equal to this vector element

Loop until the lowest index is greater than or equal to the highest index

Loop through the vector from lowest index until an item larger than the pivot is found

Overwrite lowest index with this element’s position

Loop through the vector from lowest index until an element smaller than the pivot is found

Overwrite highest index with this element’s position

Swap the elements at the new highest and lowest index

Overwrite the lowest index by increasing it by one

Overwrite the highest index by decreasing it by one

Return the highest index

End