2-3 Assignment: Vector Sorting

David Vega

Southern New Hampshire University

CS-300 DSA: Analysis and Design

Professor David Ostrowski

July 9, 2022

The purpose of this program is to sort a collection of bids stored in a vector.

There are two unique algorithms that allow us to sort the bids using min, mid and max values and they are known as selection sort and quick sort algorithms. The first sorting algorithm used is selection sort. This sorting method uses a min value to compare with the next value in the list. If the next value is less than the current min value then the next value becomes the NEW min value and the two values swap positions in the list. This process continues until all of the items in the list have been compared leaving the list sorted in ascending order. The second sorting algorithm used is the quick sort. This sorting algorithm divides the list into two separate partitions, low and high. A pivot value is chosen and compared between the remaining elements in the two partitions. This program utilizes the middle element in the collection as the pivot value. If the element is less than the pivot value, then the element remains to the left of the pivot. If the element is greater than the pivot value then the element gets swapped to the right side of the pivot value. A call to the quick sort method within the quick sort method recursively sorts through the items to the left of the pivot and subsequently on the items to the right of the pivot respectively until the sort is complete and all items are in ascending order.

The challenges encountered during the development of these sorting algorithms were mostly syntactical and IDE setup errors. The visual studio debugger was critical to helping me identify these issues. At first, I found myself recreating the project over because I thought that my project was setup incorrectly in the IDE. This was not obvious at first but my program was still running in the background as I was trying to recompile and was throwing a linker error. All I had to do was exit the program, build and compile and it worked. These are silly but common beginner mistakes.

PSUEDOCODE:

START PROGRAM

WHILE choice IS NOT equal to 9

EXECUTE

PRINT user menu:

1. Load Bids
2. Display All Bids
3. Selection Sort All Bids
4. Quick Sort All Bids
5. Exit

READ IN user input and store to variable choice for menu selection.

CHOICE 1 - (Load Bids)

* SET variable ticks equal to clock() method. Stores starting clock tick.
* INITIALIZE csv file
  + PARSE csv file and stores data into bid object
  + DEFINE vector to store collection of bids
  + RETURN bids
* PRINT size of bids
* CALCULATE elapsed time
  + Clock() – ticks
* PRINT elapsed time

CHOICE 2 (Display All Bids)

* LOOP through bids vector
* PRINT bids
* INSERT break

CHOICE 3 - (Selection Sort All Bids)

* SET variable ticks equal to clock() method. Stores starting clock tick.
* INVOKE selection sort method and pass in vector bids as a parameter.
* CALCULATE elapsed time
  + Clock() – ticks
* PRINT elapsed time
* INSERT break

CHOICE 4 - (Quick Sort All Bids)

* SET variable ticks equal to clock() method. Stores starting clock tick.
* INVOKE quick sort method and pass in vector bids, starting and ending (bid size -1) position
* PRINT bid size
* CALCULATE elapsed time
  + Clock() – ticks
* PRINT elapsed time
* INSERT break

ELSE

PRINT “Goodbye”

RETURN 0

END PROGRAM