Augustus Mendy

CS-300-11373-M01

Southern New Hampshire University

May 18, 2024

2-3 Assignment: Vector Sorting

The primary goal of this work was to implement the quicksort, partition, and selection Sort functions found in VectorSorting.cpp. I had to first have some comprehension of the duties that I was expected to perform in addition to the ones that the algorithm was supposed to sort. Items up for auction were being sorted in preparation for sale. They had to be sorted by title using the selection Sort and quicksort. I was given extremely clear directions on how to finish these processes and develop the algorithms, so I studied the comments within the methods that I would need to complete or make changes to. Although the quicksort technique was a little more complicated, I had no trouble designing the selection Sort method because it is a straightforward algorithm.

I did run into some issues with the code for the quicksort algorithm. It is a straightforward procedure, although I had some troubles with the partition method. I encountered issues with the pivot point. Overall, the challenge was rather simple, and I was able to get the code to run and function properly.

Pseudocode:

“START PROGRAM

WHILE choice IS NOT equal to 9 PRINT

user menu:

1.Load Bids

2.Display All Bids

3.Selection Sort All Bids

4.Quick Sort All Bids

9 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

oPARSE cvs 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 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 "Goodby"”

RETURN 0

END PROGRAM