**Code Reflection**

**Jaden Bryon Knutson**

**A brief explanation of the code and its purpose**

Description: Vector Sorting Algorithms – We are sorting a vector of property auction bids stored in a CSV file. We are creating and implementing the selectionSort and quicksort functions to sort the vector of bids.

This program enables the user to specify the path to a bid CSV file that can be loaded by selecting "1" in the menu (option 1). Once the file is loaded, the user can choose from various menu items, such as:

* display all bids (option 2)
* sort the bids using selection sort (option 3)
* sort the bids using quicksort (option 4)
* When the user wants to end the program, they must enter "9".
* By default, the initial CSV file is eBid\_Monthly\_Sales\_Dec\_2016.csv

I have added menu options 5 and 6 to enable the user to choose between two bid files:

* Option 5 switches the load directory to: eBid\_Monthly\_Sales.csv
* Option 6 switches the load directory to: eBid\_Monthly\_Sales\_Dec\_2016.csv

Once the new directory is switched, the user can reload bids by selecting "1" and can now sort/display the new file.

**Brief discussion of your experience in developing it**

"Overall, the process went smoothly, even though it was intimidating at first. As I worked through the assignment, the picture became clearer in my mind. While some aspects of the system made me feel uncomfortable at times, I was able to tackle most of them successfully. Nevertheless, the assignment provided me with a valuable reference to learn from and use in the future."

**Issues encountered during exercise and approaches to solve them.**

I encountered the issue of not being able to use the command in the terminal to specify the CSV file for the load designation. I saw that in the announcement that you wrote, we could hard-assign the larger CSV file to option 3. I hope it doesn’t mess things up on your part in any way, but I instead added two options to the user menu so that I could easily switch between the two files for testing and messing around with the code.

* + Option 5 is to designate the large CSV file to be loaded:

eBid\_Monthly\_Sales.csv

* + Option 6 is to designate the small CSV file to be loaded:

eBid\_Monthly\_Sales\_Dec\_2016.csv

* + I will continue to mess with the program to figure out the issue previously mentioned.

I believe I have identified a bug in my code, which appears to be more of a nuisance than a serious problem. Occasionally, all of the 'couts' in my program will suddenly be highlighted in red. Each time this happens, it seems to have a different cause, and then the issue disappears on its own, without any action on my part. The most recent error message indicated that one of my '<' symbols was causing a problem, but then it changed to say that all of my 'couts' were ambiguous. These error notifications typically only persist for a brief period before disappearing without any intervention on my part. I've looked up solutions on Google and other resources, but so far I haven't found a permanent fix.

**Pseudocode or Flowchart: A pseudocode or flowchart description of the code that is clear and understandable and captures accurate logic to translate to the programming language**

Start by defining the csvPath and other necessary variables for the program. Also, define a vector that will hold all the bids and a timer variable to keep track of elapsed time. Lastly, define a variable to store the user's choice.

While the user's choice is not equal to 9, do the following:

* Display the menu options to the user.
* Retrieve the user's choice.
* Use a switch statement to execute different actions based on the user's choice.

If the user selects option 1, load the bids by performing the following actions:

* Start the timer to measure elapsed time.
* Load the bids from the csvPath into the bids vector.
* Display the number of bids that were successfully read from the csv file.
* Calculate the elapsed time and display the result.

If the user selects option 2, display all the bids in the bids vector by iterating over each bid and outputting it to the console.

If the user selects option 3, sort all the bids in the bids vector using the selectionSort function. Perform the following actions:

* Start the timer to measure elapsed time.
* Sort the bids using the selectionSort function.
* Display the number of bids that were successfully sorted.
* Calculate the elapsed time and display the result.

If the user selects option 4, sort all the bids in the bids vector using the quickSort function. Perform the following actions:

* Start the timer to measure elapsed time.
* Sort the bids using the quickSort function.
* Display the number of bids that were successfully sorted.
* Calculate the elapsed time and display the result.

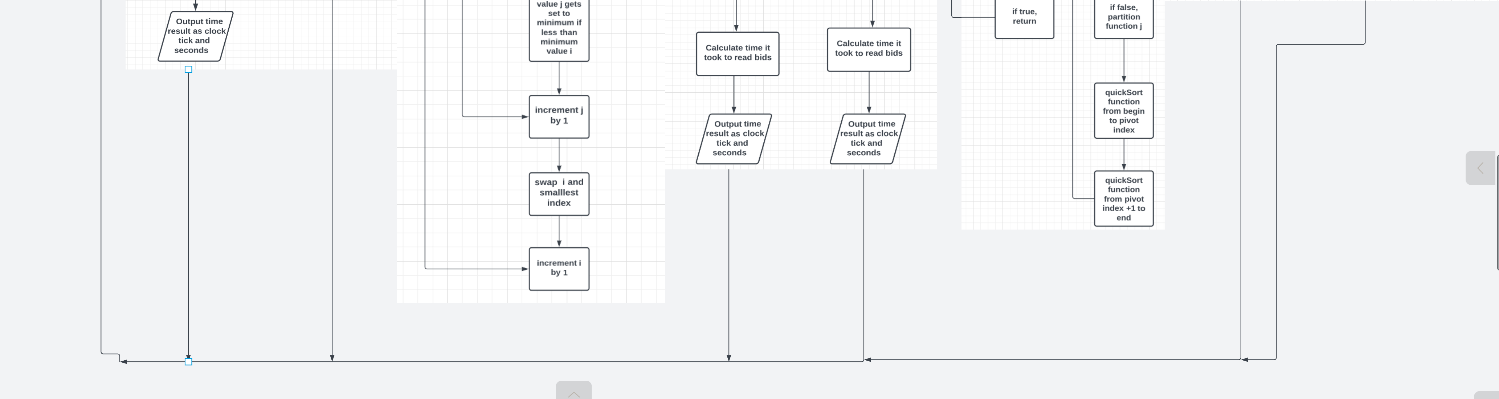
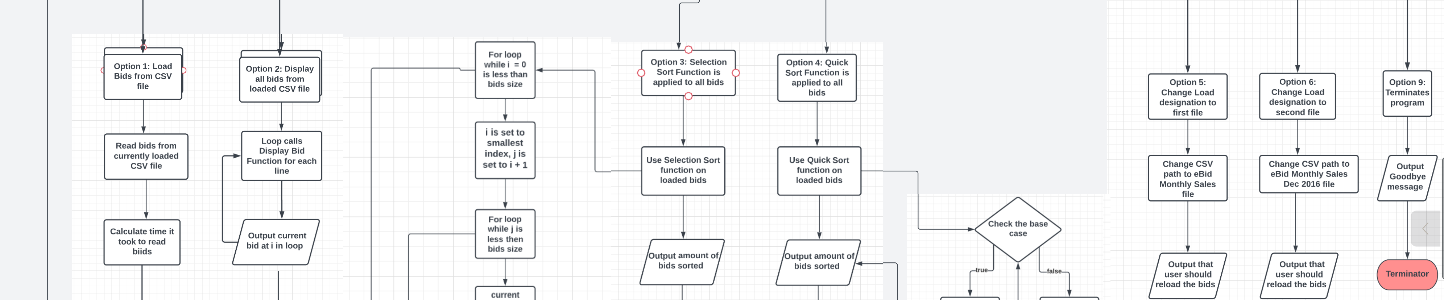
If the user selects option 5, switch the load designation to eBid\_Monthly\_Sales.csv by changing the value of csvPath to "eBid\_Monthly\_Sales.csv".

If the user selects option 6, switch the load designation to eBid\_Monthly\_Sales\_Dec\_2016.csv (default) by changing the value of csvPath to "eBid\_Monthly\_Sales\_Dec\_2016.csv".

If the user selects option 9, exit the program.

Once the loop ends, the program will reach its termination.

**A screenshot of a computer

Description automatically generated with medium confidence**