Code Reflection

The purpose of this code is to manage bids effectively. It provides functionality to load bid data from a CSV file, display the bids, and sort them by title. Users can interact with the data through a straightforward menu interface, which offers options to load, display, and sort bids using various sorting algorithms. The code also includes the capability to convert string data (such as bid amounts) into numerical formats for further processing. It utilizes basic input/output operations to capture user input for manually creating new bids and employs a CSV parser to read bid data from a file. Additionally, the program tracks execution time to evaluate the performance of the sorting algorithms, offering insights into their efficiency.

Two sorting techniques are implemented: selection sort and quick sort. Selection sort is a simple comparison-based algorithm with a time complexity of O(n²). It repeatedly selects the minimum element from the unsorted portion and moves it to the sorted portion. While this method is straightforward, its quadratic time complexity renders it inefficient for large datasets. In contrast, quick sort is a more efficient sorting algorithm with an average time complexity of O(n log n). It operates by selecting a pivot element and partitioning the data into smaller sub-arrays, which are then sorted recursively. One challenge faced during the implementation of the quick sort was ensuring proper index manipulation within the partition function. Accurate comparisons between bid titles and compelling swapping logic were crucial for maintaining both the efficiency and correctness of the algorithm. Both algorithms were timed to measure their performance, with quick sort anticipated to outperform selection sort on larger datasets due to its divide-and-conquer approach.

Pseudocode

- Initiation of the Program:

- Present a menu comprising the following options:

- Load Bids

- Display All Bids

- Selection Sort All Bids

- Quick Sort All Bids

- Exit

- Loading Bids (Option 1):

- Request the user to provide the path to the CSV file for loading (default set to "eBid\_Monthly\_Sales.csv").

- Initialize a CSV parser to facilitate reading the file.

- For each row within the CSV:

- Extract the relevant bid data, including ID, title, fund, and amount.

- Organize the extracted data within a `Bid` structure.

- Accumulate all `Bid` structures within a list (vector).

- Communicate the total number of bids successfully loaded and the time expended during this process.

- Displaying All Bids (Option 2):

- For each bid contained in the list:

- Present the bid information, which includes ID, title, amount, and fund.

- Selection Sort (Option 3):

- Iterate through the list of bids.

- For each position within the list:

- Identify the smallest bid (based on title) in the unsorted segment of the list.

- Exchange the identified smallest bid with the current position.

- Report the total number of bids sorted and the time taken for the sorting operation.

- Quick Sort (Option 4):

- Select a pivot element from the list (preferably the middle element).

- Partition the list by rearranging elements such that those smaller than the pivot are located on the left, while those greater than the pivot are on the right.

- Recursively apply the quick sort algorithm to both the left and right partitions.

- Inform the user of the total number of bids sorted and the time taken for this sorting process.