Paul Kenaga

Professor Ostrowski

CS 300

5 Nov. 2021

Reflection and Flowchart

The code for this assignment demonstrates the difference in time it takes for quick sort and selection sort to execute through a vector generated from a comma separated value file that contains a list of government bids. Using nested loops, Selection Sort iterates through unsorted elements of a list and continuously adds the next smallest element to a sorted side of the list, until all are in order. Though the implementation of this code is easier than others, its execution time is not the greatest at O(N^2). The Quick Sort process typically has a significantly faster execution time at O(N log N). Quick Sort uses a pivot point to divide elements into low and high partitions. Values in the low partition are less than or equal to the pivot point and values in the high partition are greater than or equal to the pivot point. Quick sort continuously sorts the low and high partitions until only one or zero elements are left which means they have been sorted. Though faster, the implementation of this algorithm is more complex. To call a Quick Sort method, we also need to call a Partition method. Quick Sort recursively calls the Partition method until all elements of a list are sorted.

For CSV files with fewer elements, both the quick sort and selection sort executed at the same rate. However, for larger quantities of elements, quick sort greatly outperformed selection sort.

While developing the code for this assignment, I experienced an issue with accessing the CSV file. An exception was continuously being thrown whenever I tried to run my code. At first I thought there I had made a coding error somewhere and began to troubleshoot using Stack Overflow, and by reaching out to the professor. Eventually, I realized that I needed to place the CSV files I wanted to open in the same file as my CPP code files. After making this minor adjustment, my code executed properly. The course material served as valuable reference for completing the assignment as well as online quick sort and selection sort articles. Articulation of processes/algorithms can be very difficult to comprehend and so I prefer to watch the execution of code, like in zyBooks, to better understand how different components of code work together. FlowchartDiagram

Description automatically generated