Category Two Artifact Narrative – Data Structures and Algorithms

Miguel Little

CS-499 Computer Science Capstone

Professor Brooke

04/20/2024

Briefly describe the artifact. What is it? When was it created?

My artifact is part of my CS-260 course (Data Structures and Algorithms). I worked on it about a year ago, that takes bids from a CSV file and allows the user to sort the bids using quicksort algorithms that sort the data by using while loops and comparing the highest and lowest bids per loop.

Justify the inclusion of the artifact in your ePortfolio. Why did you select this item? What specific components of the artifact showcase your skills and abilities in software development? How was the artifact improved?

The reason I chose this artifact for my ePortfolio is that it will be a great way to demonstrate an understanding algorithms and data structures, this project also left a lot of room for improvement. It allowed me to build sort functions that sorted by parameters, as well as include "Add a bid" functionality.

I showcased my ability to create structures and functions in Python that will add to the program. The user now can create a new bid and then sort all the bids in the CSV file by date, title, or amount.

I also included time complexity for sort functions and a bubble sort algorithm title to show my expertise in data structures and algorithms.

Did you meet the course objectives you planned to meet with this enhancement in Module One? Do you have any updates to your outcome-coverage plans?

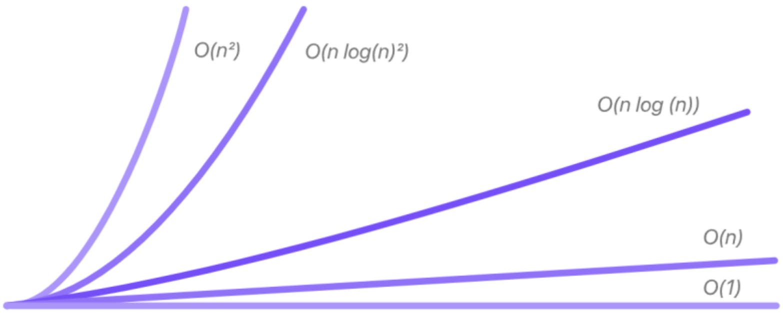
Through the completed work, I met these course outcomes:

I believe I met the outcome: *"Design, develop, and deliver professional-quality oral, written, and visual communications that are coherent, technically sound, and appropriately adapted to specific audiences and contexts."* By properly implementing sort by parameter functions and a bubble sort algorithm. My code is well commented out and I will be adding a ReadMe to the repository for additional notes.

I believe I have met the outcome: *"Demonstrate an ability to use well-founded and innovative techniques, skills, and tools in computing practices for the purpose of implementing computer solutions that deliver value and accomplish industry-specific goals (software engineering/design/database)."* By adding sort functions per parameter. I have also implemented this by adding the bubble sort algorithm for comparable algorithms and implementing time complexity for the sort functions.

I have met: *"Employ strategies for building collaborative environments that enable diverse audiences to support organizational decision making in the field of computer science."* once I finish all the pushed changes in the git repository. I have also commented out the code very concisely so that other programmers can jump in and make changes easily.

I have met *"Design and evaluate computing solutions that solve a given problem using algorithmic principles and computer science practices and standards appropriate to its solution,* *while managing the trade-offs involved in design choices (data structures and algorithms),"* by implementing a bubble sort algorithm for a title sort function, as well as by implementing time complexity into the code for the functions.



(Big-O best case, middle case, and worse case scenarios for bubble sort where n represents the number of items in the array list.)

Reflect on the process of enhancing and modifying the artifact. What did you learn as you were creating it and improving it? What challenges did you face?

The main thing I learned in enhancing this code is that it was possible. When I started working on this enhancement, I genuinely felt lost and as though I would not be able to make any substantial changes, and now I am looking forward to making more changes.

The sort by functions were returning "nan" for date and amount for a very long time. Initially it was because I was mixing up float and int for the values. Another difficulty was converting the dates into date time objects to be able to sort by date. Also, another issue that I was running into was calculating what was the actual runtime of the functions because they were running very quick so it would return an output of 0 seconds. Because of this it was difficult to verify whether the logic was correct and if time complexity was properly implemented.

References

Home - Big-O. (2020). Big-O; Big-O. https://big-o.io/