Kris Marchevka

Professor Gudivada

CS-499 Computer Science Capstone

9/28/2025

Milestone 3 Narrative

The artifact I chose was Project 2 from CS-300 Data Structure & Algorithms – Analysis & Design. It works by loading a CSV file of courses and allowing the search and printing of them through a binary search tree given each courses prerequisite. The original itself worked just fine but was limited in its efficiency and complexity.

This artifact shows my skills with algorithms and data structures in a very practical manner and now demonstrates my ability to implement a few data structures at a time like BST, AVL tree, hash tables, and vectors and then compare their runtime through benchmarking. Additionally, I implemented a directed graph to model the course prerequisites with cycle detection and topological sorting as well as CSV validation for bad rows or missing prerequisites. With these enhancements considered, I showcase how I can think about efficiency, data integrity, and algorithm as a whole.

The enhancement I showcase for Milestone 3 are part and parcel with my planned enhancements that I planned for back in Module 1. More specifically, I showcased my ability to apply algorithmic principles to solve problems, evaluate the trade-offs in choice of data structures, and actually built them in a way that is functional, effective, and maintainable. I do not have any updates on the outcome-coverage plan as I have hit the mark in my opinion, for what I’ve set out to do.

When reflecting on the process of enhancing my artifact, it showed me the importance of clean input handling, the impact of balancing in trees, and how graph algorithms can help you discover issues like prerequisite cycles that aren’t readily apparent in the data (even though we had a small dataset). The hardest part was integrating all the data structures in a way that kept everything maintainable and readable. Learning about these data structures was a lot of fun and required lots of reading/video demonstrations. The benchmarking was also an excellent way to show how each time complexity showcases itself in the final execution time as seen below. A computer screen with white text

AI-generated content may be incorrect.