Dasha Gore

November 18, 2024

CS 499

Milestone Three: Algorithms and Data Structures

The artifact I selected for my ePortfolio is a pathfinding algorithm project from CS 300: Data Structures and Algorithms. This project involves implementing Dijkstra’s Algorithm to find the shortest path in a weighted graph. It was originally created as part of coursework focused on teaching algorithm design, efficiency, and problem-solving using complex data structures. This artifact demonstrates my ability to apply theoretical knowledge to practical programming challenges, showcasing skills in graph traversal, efficient data storage, and computational problem-solving.

I chose this artifact because it highlights my ability to work with foundational computer science concepts that are essential for solving real-world problems. Dijkstra’s Algorithm is a widely recognized solution for determining the shortest path, and implementing it required a deep understanding of graph structures, priority queues, and algorithmic efficiency. This artifact also reflects my ability to design and analyze solutions while managing trade-offs between computational complexity and clarity of implementation. The project stands out because it not only illustrates my ability to build functional solutions but also my capacity to improve them through thoughtful enhancements.

For this milestone, I enhanced the original artifact by focusing on key areas such as code modularity, edge case handling, performance optimization, and user feedback. First, I refactored the code to make it more modular. Breaking the code into smaller functions, such as graph construction, shortest path calculation, and result display, improved its readability and maintainability. These modular functions can now be reused or adapted more easily for future projects. This change also aligns with best practices in software design, ensuring the artifact meets professional standards.

Next, I addressed edge case handling, which was limited in the original version. I added checks to validate inputs, ensuring that start and end nodes exist within the graph before proceeding. I also implemented logic to handle scenarios where no path exists between the given nodes. These enhancements made the program more robust and user-friendly, as it now provides clear error messages instead of failing silently or producing incorrect results.

In terms of performance, I improved the algorithm by optimizing its data structures. I refined the priority queue operations to reduce overhead during edge relaxation, which is a critical step in Dijkstra’s Algorithm. Additionally, I used an adjacency list representation for the graph, which is more efficient for both space and time compared to other representations like adjacency matrices. These changes ensured that the program could handle larger graphs more effectively while maintaining computational efficiency.

To improve usability, I incorporated enhanced user feedback. The program now includes clear prompts for users to input graph data and specify start and end nodes. It also provides meaningful progress updates during execution and displays results in a more readable format. The shortest path and its total cost are presented step-by-step, ensuring the user can easily follow the output. These improvements make the artifact more interactive and accessible to users who may not have a technical background.

Through these enhancements, I achieved key outcomes related to algorithm design and software development. I demonstrated my ability to analyze and refine an existing solution, incorporating principles of modularity, efficiency, and usability. The process also reinforced the importance of anticipating and addressing edge cases, which is a critical skill for creating reliable software.

Enhancing this artifact was a valuable learning experience. It deepened my understanding of Dijkstra’s Algorithm and graph-based problem-solving while teaching me how to balance algorithmic performance with user-centered design. One challenge I faced was ensuring that the enhancements, such as modularization and performance optimizations, did not inadvertently introduce bugs or reduce the program’s clarity. To overcome this, I conducted extensive testing with various graph configurations, including edge cases like disconnected nodes and invalid inputs. This testing process not only validated the enhancements but also strengthened my problem-solving and debugging skills.

Overall, this artifact reflects my growth as a computer science student and my ability to design effective solutions to complex problems. By including this project in my ePortfolio, I aim to showcase my technical skills, attention to detail, and commitment to creating software that is both efficient and user-friendly. This artifact is a testament to my ability to build on foundational knowledge, adapt to challenges, and deliver high-quality solutions.