Kelly Perez

11/22/2024

CS 499

Milestone Three Narrative

**Intro to the Artifact**

The C++ implementation of a triangle is the artifact I have chosen for my ePortfolio. The enhancement of Dijkstra's Algorithm uses a greedy approach to determine the shortest pathways between a source node and every other node in a graph. This artifact was initially produced as an assignment from CS 330, where I created a triangle in C++.

**Justification for Including the Artifact**

Because it demonstrates my proficiency with algorithms and data structures, particularly graph traversal and optimization techniques, I selected this artifact for my ePortfolio. The inclusion of Dijkstra's Algorithm, a fundamental algorithm in computer science, demonstrates my capacity to apply difficult ideas in a useful and effective manner. Additionally, the improvements I made to the artifact show that I am capable of enhancing functionality, performance, and usability. By presenting this artifact, I hope to demonstrate my capacity to use theoretical understanding to address practical computing issues.

**Course Outcome Alignment**

The improvements are exactly in line with the goals I had set for the course. I demonstrated my grasp of data structures by enhancing the algorithm's performance, using algorithmic principles to maximize computational efficiency, and creating a graphic to convey difficult ideas clearly. These enhancements address the results pertaining to algorithm design, data structure impact, and technical communication.

**Reflection on the Enhancement Process**

Improving this artifact taught me more about algorithm design and optimization, which was a worthwhile experience. The significance of striking a balance between theoretical ideas and real-world application was one important lesson. For instance, careful consideration of graph traversal and distance updates was necessary while switching from a set-based approach to a priority queue, which eventually enhanced the algorithm's performance. Implementing path reconstruction also increased my understanding of how predecessor arrays and other data structures work in tandem with algorithms to produce more insightful results. I learned the value of strong error handling through debugging invalid input circumstances, and including graphical visualization forced me to learn new libraries and hone my technical communication abilities.

**Challenges Faced**

Understanding how the priority queue, also known as min-heap, works with the algorithm to effectively retrieve the node with the shortest distance was one of the major hurdles. Understanding how the priority queue and distance array cooperate to guarantee that every node is visited in the right order and without redundancy required some time.