

## Assignment Review

In this review, I will describe the overall process, specific task implementations, challenges encountered, and key lessons learned from this project. My approach to the assignment was systematic, dealing with each task in the order they were presented, except for the tests that I tackled in the end. With this approach, I wanted to maintain clarity and organization throughout the project.

I began with constructing the Graph class. Firstly, I implemented methods like *readFromFile* to process the input files, *addEdge* to add new nodes, *getPlaces* to ensure valid places, and *shortestPath* to use Dijkstra's algorithm. The last method I completed is *saveToReport*, which was difficult for me to implement and I believe the crucial part of this assignment. After dealing with all of the difficulties, I tried to make the output files look clearer and more readable.

The second task was relatively straightforward. Using priority queues in Dijkstra's algorithm, the *shortestPath* method calculates the shortest path between two nodes. It provides insights into the time required to travel from source to destination and returns -1 for all irregularities. When I was done with the coding part, I covered up this class with tests where I passed different graph configurations with weights and tested correct outputs.

Task three was about determining optimal time between specified locations, for any given input file. The corresponding path weights are written into the output files. If there is no path between two nodes or one of the nodes is not included in the places.txt file, -1 is written instead of the weight.

The coding journey witnessed critical adaptations for the final task. Introducing the feature, where the code takes into account obstacles with different chances of happening, added an interesting twist. It computes the probability of encountering obstacles on specific routes. This task required a lot of changes to be made in the *saveToReport* method. If there is a constraint on a specific route, it should be written at the top of the output file, and excluded when calculating the shortest path between other nodes. To make this easier, I made *allConstraints* Map which stores all source-destination pairs between which there is an obstacle.

The most challenging part of the assignment was deciding how to start with the project and create its structure, as well as writing this report. The part I thought would be relatively simple is the last task because it looked like the code needed just a few modifications to support it.

If I was starting the assignment again, I would probably pay more attention to the details. Making the code cleaner and easier to read and finding ways to make it work faster are areas I would like to focus on in the future. Overall, this assignment was very creative and interesting, as it required using knowledge from Statistics, and Data Structures and Algorithms courses. The project structure let us apply what we learned in classes to solve problems in a practical way, and thus it transformed our knowledge from theoretical constructs to real-world solutions.

Ajša Hercegljia