

# Assignment Review

The proposed assignment tackles the perennial issue of urban transportation in Sarajevo by introducing a unique solution involving quantum mechanics-inspired glass capsules. This review reflects on the implementation process, challenges faced, and the effectiveness of the designed program for finding optimal paths between various city points. The tasks were interconnected which made coding easier, since every additional line of code was just an “upgrade” on the already existing lines.

The first task was to create a weighted graph, which I decided to do using a map in Graph class which contained nodes from the Node class. This way I found to be the most logical implementation and I did not experience any massive issues while writing the code. Apart from that, I believe that it makes my code look clean and readable. Inside the Graph class, I have ensured that in the creation of the edge, i.e. nodes, the probability of the constraint that might exist between the two nodes processed at the time is handled properly, that is if the constraint happens, that path is ignored. This is exactly what the fourth task is about.

The second task was to create a class for solving the shortest path algorithm between any two nodes in the graph, with proper exception handling. I decided to implement Dijkstra algorithm and that is something that took the most time to do. Luckily the problems were not major and relatively easy to solve, with the help of the Internet and YouTube. Some of the exceptions that I needed to handle were situations where there is no path between the two nodes, situations where there are negative distances and situations with non-existent nodes. All these situations are also covered in tests, and I have also tested the algorithm for self-loops and whether all nodes in the graph are connected amongst themselves, which third task noted to be done in the main program as well. There are also two additional tests which are primarily focused on seeing how the program handles the constraints, with one test being the 100% probability and the other being 0% probability.

To conclude, I would say that I am satisfied with the overall implementation and solution of the given assignment. It had an interesting storyline to it and made the “dull” task of Dijkstra implementation on the graph fun. The assignment seemed easy at first, and overall, it was not that hard, but there were some issues with me reading what needs to be done (functioning of probabilities in constraints.txt file). However, the changes were easy to implement, which made life much easier. As far as the future improvements are concerned, there could always be some additional instructions given for the creation of the graph/nodes or something else, which could ultimately lead to the program requiring some additional logic to be added. Other than that, there is always room for improvement, so I am sure there is some here as well.