Explanation Assignment1

Files reading and information extraction

At first the problem file will be read. For each problem a new class object of the class Problem will be created containing all necessary data. Afterwards the schedule file of the first problem will be read. For every unique station code a new object of the class Station will be created containing information of the station. These are the vertexes of the graph. Connections between to stations will be saved as objects of the class Connection, which are the edges of the graph. The class Connection also contains the data for the cost functions, only the shortest distance will be tracked, but all possible drive off times. If the islno is greater than ten fake edges with the cost of ten, will be created to stations of the same train with a islno of at least ten less , which will be only used for the cost function “Price”. After the cost calculations the prices of the vertexes will be rested.

Calculation of the “shortest” path

The calculation of the “shortest” path between the start and the destination station will be done via an implementation of the Dijkstra algorithm according to the cost function. During the execution of the algorithm the cost of the vertexes, the predecessor and the used connection will be tracked. The working graph is a copy of the original graph, which shrinks during the execution of the algorithm. Because the Dijkstra algorithm is greedy, the cost of a vertex won’t change again after it has been the vertex which the lowest cost in the working graph and can be deleted from the working graph after checking all its connections. If the train of the predecessor has been reached by a train with a different train number of the currently checked connection, five minutes will be first added to respect the time to switch trains and then subtracted again. Also, all departure times must be checked to ensure the fastest connection.

Preparing the output

After the Dijkstra has finished the path will be extracted by going from the destination station backwards via the tracked predecessors and used connections, until the start station has been reached. Afterwards the path will be trimmed to the given output format.

Writing the solution to a new file

After all given problem have been solved a new file called “Solution.csv” will be created, where each line is a solution to a problem containing the problem number, the used train with islnos and the calculated cost.

Dijkstra Algorithm

Start vertex has “distance” 0, remaining vertexes “distance” infinity/ very big

Select vertex of working graph with minimal “distance”

> Delete vertex from working graph
>
> Calculate “distance” for all neighbours by adding the “distance” of the selected vertex and the “distance” of the connection to the neighbour
>
> If the calculated “distance” is smaller than the current “distance” of the neighbour, update the “distance” value and set as predecessor of the neighbour the selected vertex