
Exercise Sheet 2

Submit until Wednesday, May 9 at 4:00pm

Exercise 1 (8 points)

Implements Dijkstra's algorithm, considering the explanations and implementation advice given in the lecture. Don't forget to write a good unit test. All of the more sophisticated algorithms to come will build on this implementation, so it is absolutely essential that your implementation is correct.

Exercise 2 (4 points)

Add a method *reduceToLargestConnectedComponent* to your *RoadNetwork* class that reduces the graph (already read from an OSM file) to its largest connected component. Consider the implementation advice given in the lecture.

Exercise 3 (4 points)

Run Dijkstra's algorithm for 100 random queries for both of our OSM graphs from Exercise Sheet 1 (separately). Use the version of the graph reduced to its largest connected component (see Exercise 2). In the result table linked from the Wiki report the average running time and the average shortest path length per query, and whatever other information is asked for there.

Exercise 4 (2 points)

Submit your code to our SVN and check that *compile*, *test*, and *checkstyle* run through without errors on our build system *Jenkins*. The link to that is provided on your Daphne page.

Exercise 5 (2 points)

Add and commit a text file *feedback-exercise-sheet-2.txt* where you briefly describe your experiences with this exercise sheet and the corresponding lecture. In particular, say how much time you invested and where you had problems.