Chair for Algorithms
and Data Structures
Prof. Dr. Hannah Bast
M. Brodesser/E. Sawin

Efficient Route Planning
SS 2012

http://ad-wiki.informatik.uni-freiburg.de/teaching

UNI FREIBURG

Exercise Sheet 8

Submit until Wednesday, July 4 at 4:00pm

Exercise 1 (12 points)

Implement *parts of* the precomputation and query algorithm for transit node routing, based on contraction hierarchies, as explained in the lecture.

For the sake of simplicity, there is no need that you precompute and store the set of access nodes and the distances to them for *each* node in the graph. Similarly, there is no need that you precompute and store the distance between *each* pair of transit nodes. Just compute these on the fly for a given query from s to t, but take care that in the query times reported on the Wiki, you *only* measure the time for computing $\min\{\text{dist}(s,x)+\text{dist}(x,y)+\text{dist}(y,t) : x \in X(s), y \in X(t)\}$, and not for computing the sets $X(s)$ and $X(t)$ and the $\text{dist}(s,x)$, $\text{dist}(x,y)$, and $\text{dist}(y,t)$.

For the query algorithm, you can always assume that $\text{Far}(s,t) = \textit{true}$. That is, you don't have to implement the "far-away" criterion. That way you will not necessarily compute the shortest path when s and t are close together, but the experimental results will still be meaningful and interesting.

As usual, consider the implementation advice given in the lecture and the code design suggestions linked from the Wiki. In particular, take care that the Dijkstra searches per contraction take only very little time, as explained in the lecture.

Exercise 2 (6 points)

Use your code from Exercise 1 to compute a set of 1000 transit nodes and then execute 1000 random queries.

As usual, report your results in a row on the table linked from the Wiki. In particular, report the average number of access nodes, the average time to comput them, the average query time, and the average shortest path cost.

Exercise 3 (2 points)

As usual, commit your code to our SVN and check that everything works on Jenkins, and also commit a text file *feedback-exercise-sheet-8.txt* where you briefly describe your experiences with this exercise sheet and the corresponding lecture.