Departement of Computer Science
Markus Püschel, David Steurer
Karel Kubicek, Johannes Lengler

6 December 2019

Algorithms & Data Structures Programming task 13 HS 19

The solutions for this sheet need to be submitted to the judge by **Friday, 20 December, 23:59:59**. Any minor changes to the template will be announced at Moodle: `https://moodle-app2.let.ethz.ch/mod/forum/discuss.php?d=41683`.

Exercise 13.1 *Custom graph* **(2 bonus point)**.

Your task is to implement a directed graph representation, such that it supports the following methods of graph G with given complexity.

$addEdge(u, v)$ adds an edge from u to v to G in $\mathcal{O}(1)$.

$removeAllOut(u)$ removes all outgoing edges from u in G in $\mathcal{O}(\deg_{\text{out}}(u))$.

$removeAllIn(v)$ removes all incoming edges to v in G in $\mathcal{O}(\deg_{\text{in}}(v))$.

$isPath(u, v)$ returns *true* if there exists a path from u to v in G, in time $\mathcal{O}(|V| + |E|)$.

The input is a sequence of these operations in an arbitrary order.

You get one point for each passing test set. The test set `small` does not test time complexity. That is done by the test set `large`.

Submission: Submit your `Main.java` at `https://judge.inf.ethz.ch/team/websubmit.php?cid=28784&problem=AD19H13P`. The enrollment password is “`asymptotic`”.

Exercise 13.2 *Dijkstra's algorithm* **(2 bonus point)**.

You are given a directed weighted graph, given by a distance matrix. All distances (edge weights) are positive. Your task is to find the shortest path from u to all vertices of the graph using Dijkstra's algorithm.

You are allowed to use PriorityQueue and comparators (if necessary). If you want to practice, consider implementing your own priority queue, for example using heap, it is a nice example of applying a data structure to construct an advanced algorithm.

You get one point for each passing test set. The test set `small` does not test time complexity. That is done by the test set `large`. JUnitTest `testSingle` is meant for debugging, it can run a single instance that you struggle with, and visualize the respective greph.

Submission: Submit your `Main.java` at `https://judge.inf.ethz.ch/team/websubmit.php?cid=28784&problem=AD19H13Q`. The enrollment password is "`asymptotic`".