RWTH Aachen **WS 2009**
Lehrgebiet Theoretische Informatik Aufgabenblatt 5
Kneis—Langer—Rossmanith 1. Dezember 2009

Softwareprojektpraktikum Graphalgorithmen

Vervollständigen Sie zunächst Ihre Graphbibliothek und die zugehörige Dokumentation. Alle Programme aus den vorhergehenden Aufgabenblättern sollen lauffähig und testbar sein. Vervollständigen Sie auch Ihre Unit-Tests und dokumentieren Sie sie. Mindestens jede Methode soll getestet werden.

Aufgabe 19

Implementieren Sie den Algorithmus von Kruskal zur Berechnung minimaler Spannbäume von ungerichteten, gewichteten Graphen.

Aufgabe 20

Erstellen Sie ausführbare Programme, die ihre Eingabe von `stdin` lesen und ihre Ausgabe nach `stdout` schreiben. Diese einzelnen Programme sollen die Namen `acyclicity`, `strong-components`, `topsort`, `shortest-path` und `mst` heißen.
Die Programme sollen jeweils einen Graphen (ggf. mit Kantengewichten) einlesen und folgende Ausgaben liefern:

`acyclicity`
`yes` oder `no`, je nachdem ob der Graph azyklisch ist.

`strong-components`
Jede starke Zusammenhangskomponente soll eine Nummer erhalten und die Nummern der starken Zusammenhangskomponente jedes Knotens soll in der Reihenfolge $0, \ldots, n-1$ ausgegeben werden.

`topsort`
Die Knotennummern in einer topologisch sortierten Reihenfolge oder `no`, falls es eine solche nicht gibt.

`shortest-path`
Die Distanzen aller Knoten vom Knoten mit der Nummer 0 in der Reihenfolge 0..n-1. Falls ein Knoten nicht erreichbar ist, soll `infty` für seine Distanz geschrieben werden. Falls es irgendeinen Knoten gibt, der zwar von 0 erreichbar ist, aber es keinen kürzesten Weg zu ihm gibt, dann soll das Programm `no` ausgeben.

`mst`
Der Graph, der durch die Kanten eines minimalen Spannbaums induziert wird oder `no`, falls es keinen Spannbaum gibt.

Aufgabe 21

Auf der Webseite gibt es verschiedene Dateien, die gerichtete Graphen im LuFGTI-Format enthalten. Verwenden Sie Ihre Programme, um festzustellen, welche der Graphen azyklisch sind und wieviele starke Zusammenhangskomponenten sie haben.

Weiter gibt es Dateien mit Graphen, die Kantengewichte haben. Stellen Sie für diese Graphen fest, wie lang der kürzeste Weg vom Knoten 0 zum Knoten 1 ist und ob ein solcher existiert. Stellen Sie weiterhin fest, ob diese Graphen einen Spannbaum besitzen und gegebenenfalls welches Gewicht ein minimaler Spannbaum besitzt.