Prof. Dr. V. Linnemann
Nils Hoeller
Jana Neumann
Universität zu Lübeck
Institut für Informationssysteme

Lübeck, den 28. Mai 2009

Algorithmen und Datenstrukturen

Sommersemester 2009

8. Übungsblatt

Aufgabe 1: Graph

Gegeben sei die folgende Adjazenzmatrix:

	1	2	3	4	5	6	7
1	0	1	0	0	0	0	0
2	1	0	1	0	0	1	1
3	0	0	0	0	0	0	1
4	1	0	0	0	0	1	0
5	0	1	0	1	0	0	0
6	0	0	0	0	0	1	0
7	0	0	0	0	0	0	0

a) Zeichnen Sie den durch die Matrix definierten Graph G an. (1 Punkt)

b) Geben Sie einen einfachen Weg in G vom Knoten 4 zum Knoten 7 an. (1 Punkt)

c) Zeichnen Sie einen spannenden Baum B des Graphen G. Markieren Sie den Wurzelknoten. (2 Punkte)

d) Geben Sie eine topologische Sortierung von B an. Gibt es eine topologische Sortierung von G ? Begründen Sie Ihre Antwort. (2 Punkte)

e) Beschreiben Sie G durch eine Adjazenzliste. (1 Punkt)

f) Zeichnen Sie die reflexive, transitive Hülle G^* von G. (1 Punkt)

Aufgabe 2: Straßenplan

Die für dieses Übungsblatt erstellten Klassen sollen beim kommenden Übungsblatt weiter verwendet werden. Arbeiten Sie also im eigenen Interesse sorgfältig. Kommentieren Sie alle Klassen!!

Ein Stadtplan soll durch einen Graph repräsentiert werden: ein Stadtplan besteht aus Orten (Knoten), die durch Straßen (Kanten) miteinander verbunden sein können. Orte und Straßen besitzen einen Namen. Straßen können in beide (ungerichtet) oder nur eine Richtung (gerichtet) befahrbar sein. Zusätzlich soll für eine Straße die Länge als gewichtender Faktor berücksichtigt werden.
Der Stadtplan wird folglich durch einen gerichteten und gewichteten Graph beschrieben.

a) Implementieren Sie eine Klasse *Ort*, die einen Knoten des Graphen repräsentiert. Ein Ort hat einen Namen und eine Nummer. (1 Punkt)

b) Implementieren Sie eine Klasse *Strasse*, die eine Kante des Graphen repräsentiert. Eine Straße hat eine Nummer, einen Namen, einen Startort, einen Zielort, ein Gewicht (die Länge der Straße) und eine Komponente, die angibt ob die Straße gerichtet oder ungerichtet ist. (2 Punkte)

c) Implementieren Sie eine Klasse *Stadtplan*, die den Graphen repräsentiert. Die Klasse beinhaltet eine Menge von Orten und Strassen. Verwenden Sie die LinkedList aus der Java API, um eine Menge von Objekten zu realisieren. (2 Punkte)

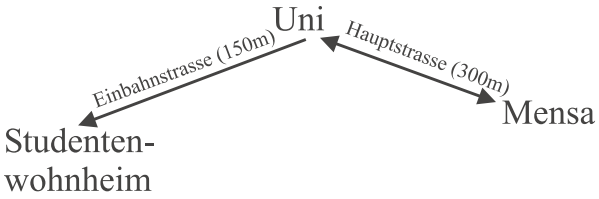
d) Die Definition eines Stadtplans soll jetzt aus einer Datei gelesen werden. Die Datei hat folgenden Aufbau:

```
n
m
1;Knotenname_1
2;Knotenname_2
..
n;Knotenname_n
1;Kantenname_1;StartNr_1;ZielNr_1;Richtung_1;Gewicht_1
2;Kantenname_2;StartNr_2;ZielNr_2;Richtung_2;Gewicht_2
..
m;Kantenname_m;StartNr_m;ZielNr_m;Richtung_m;Gewicht_m
```

wobei n die Anzahl der Orte (Knoten) und m die Anzahl der Strassen (Kanten)

ist. Die Richtung wird durch die Schlüsselwörter `gerichtet` und `ungerichtet` definiert. Das Gewicht ist eine ganze Zahl. Die StartNr und ZielNr bezieht sich auf die vorher definierten Knotennummern. Die einzelnen Datenfelder sind immer durch Semikola (;) getrennt.

Als Beispiel wird folgender Stadtplan durch eine Datei beschrieben:




```
3
2
1;Uni
2;Studentenwohnheim
3;Mensa
1;Hauptstraße;1;3;ungerichtet;300
2;Einbahnstraße;1;2;gerichtet;150
```

Implementieren Sie in der Klasse *Stadtplan* eine Methode, die eine solche Datei einliest und die Knoten und Kantenmengen korrekt füllt. Testen Sie Ihre Methode anhand des Stadtplanes *Studentendorf.txt*, die Sie von der Übungsseite herunterladen können. Um die Datenfelder aus einer Zeile zu extrahieren, können Sie die Methoden *indexOf()* und *substring* der Klasse *String* verwenden (siehe *java.lang.String* in der API). (6 Punkte)

e) Zeichnen Sie den Graph, der durch die Datei *Studentendorf.txt* definiert ist. Straßen sollen sich nicht kreuzen; die Entfernungen brauchen nicht berücksichtigt zu werden. (1 Punkt)

Abgabetermin: Donnerstag, den 04. Juni bis 11 Uhr Abgabekasten IFIS