

# Algorithmen und Datenstrukturen

Sommersemester 2009

## 11. Übungsblatt

### Aufgabe 1: Minimale Spannende Bäume

Ein Spediteur muss zwischen verschiedenen Städten Waren mittels Lkws transportieren. Um in die gewünschten Städte zu gelangen, stehen dem Spediteur verschiedene Verbindungen bestehend aus z.B. Autobahnen zur Verfügung. Es gibt mehrere mögliche Verbindungen, die allerdings unterschiedlich viel Mautgebühr kosten. Der Unternehmer ist nun daran interessiert, alle gegebenen Städte möglichst günstig hinsichtlich Maut zu erreichen. Da sein Städte- und Verbindungsmodell mit Maut einen zusammenhängenden, ungerichteten, bewerteten Graphen  $G$  beschreibt, wird dazu lediglich der minimale spannende Baum benötigt. Auf der Übungsseite zur Vorlesung können Sie das Projekt *AUD\_Autobahn* herunterladen. Im Prinzip ist es analog zum *Strassenplan*-Projekt der letzten Zettel aufgebaut. Diesmal ist der Graph, der aus einer Datei eingelesen wird, allerdings immer ungerichtet und das Gewicht repräsentiert die Maut in Euro. Im Folgenden soll lediglich die Klasse *Verbindungsplan* um einige Methoden erweitert werden.

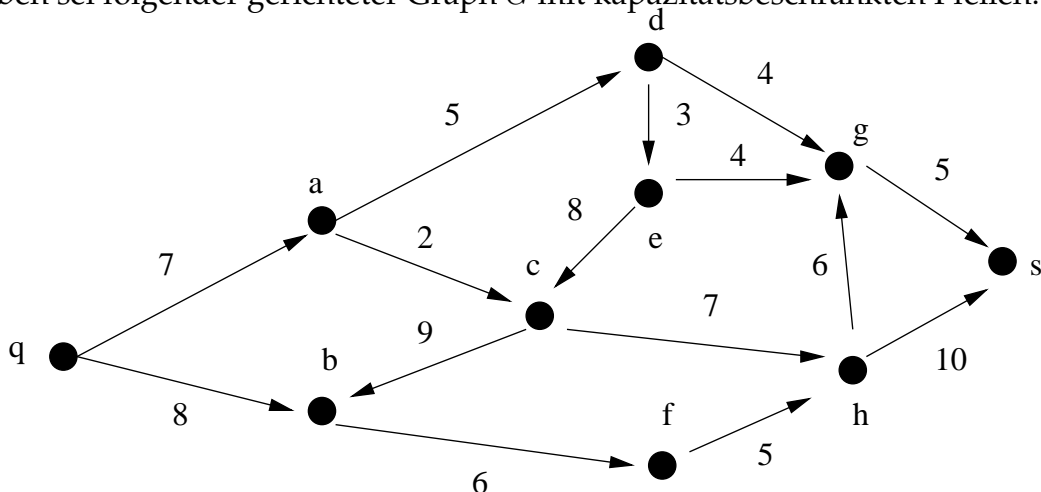
- a) Für den Algorithmus von Kruskal werden die Kanten des Graphen in aufsteigender Gewichtung abgearbeitet. Warum müssen Sie in der folgenden Implementierung diese Sortierung der Kanten nicht mehr vornehmen? (1 Punkt)
- b) Die (gewählten) Bäume im Algorithmus sollen hier als (gewählte) Knotenmengen mittels *LinkedLists* modelliert werden. Implementieren Sie *find*, *union* und *makeSet* als private Hilfsmethoden. Die Methode *find* bekommt einen Knoten (Stadt) sowie eine Menge (*LinkedList*) von aktuell gewählten Knotenmengen (*LinkedList*) übergeben und liefert die zugehörige gewählte Knotenmenge als

*LinkedList*. Die Methode *union* bekommt zwei gewählte Knotenmengen übergeben und liefert eine gewählte Knotenmenge zurück, die alle Knoten der Eingangslisten beinhaltet. Die Methode *makeSet* generiert aus einem übergebenen Knoten (Stadt) eine einelementige gewählte Knotenmenge, die zurückgeliefert wird. (6 Punkte)

- c) Implementieren Sie nun den Algorithmus von *Kruskal* in einer öffentlichen Methode *kruskal*, die den minimalen spannenden Baum des eingelesenen Strassenplans mittels der Hilfsmethoden aus b) berechnet und die gewählten Kanten als *LinkedList* zurückliefert. (5 Punkte)
- d) Schreiben Sie eine Methode *gewahlteKantenAusgeben*, die das Ergebnis der Methode aus c) benutzerfreundlich ausgibt. Wie lauten die gewählten Autobahnverbindungen des Graphens, der aus der Datei *Autobahnnetz\_gross.txt* generiert werden kann? (2 Punkte)

## Aufgabe 2: Flüsse in Netzwerken

Gegeben sei folgender gerichteter Graph  $G$  mit kapazitätsbeschränkten Pfeilen:



Bestimmen Sie mit dem in der Vorlesung vorgestellten Algorithmus den maximalen Fluss in  $G$ . Geben Sie dabei für jeden gewählten zunehmenden Weg den Restgraphen, die Flussserhöhung und den Weg an. Gibt es mehrere mögliche zunehmende Wege, so soll derjenige, mit der geringsten Pfeilanzahl und der maximal möglichen Flussserhöhung gewählt werden. (6 Punkte)

---

**Abgabetermin:** Donnerstag, 25. Juni 2009, 11 Uhr Abgabekasten IFIS und Code per Mail