

## Einführung in die Diskrete Mathematik

## 10. Übung

1. Sei  $(G, u, b, c)$  eine Instanz des MINIMUM-COST-FLOW-PROBLEMS. Sei  $e_0 \in E(G)$  eine Kante mit  $c(e_0) > (|V(G)| - 1) \max_{e \in E(G) \setminus \{e_0\}} |c(e)|$ . Beweisen Sie die folgende Aussage: Wenn es einen  $b$ -Fluss  $g$  in  $(G, u)$  mit  $g(e_0) = 0$  gibt, dann gilt  $f(e_0) = 0$  für jede Optimallösung  $f$ . (4 Punkte)
2. Es sei  $(G, u, b, c)$  eine Instanz des MINIMUM-COST-FLOW-PROBLEMS mit ganzzahligen Kosten  $c$ . Außerdem sei  $f$  eine optimale Lösung und  $e_0 \in E(G)$ . Die Kostenfunktion  $c' : E(G) \rightarrow \mathbb{Z}$  sei wie folgt definiert:  $c'(e_0) = c(e_0) + 1$  und  $c'(e) = c(e)$  für  $e \in E(G) \setminus \{e_0\}$ . Zeigen Sie, wie man zu gegebenem  $(G, u, b, c')$ ,  $e_0$  und  $f$  in Zeit  $O(|V(G)|^3)$  einen kostenminimalen Fluss in  $(G, u, b, c')$  finden kann. (4 Punkte)
3. Der Weihnachtsmann soll in einer Stadt Geschenke austragen, die nur aus Einbahnstraßen besteht. Die Stadt kann als stark zusammenhängender gerichteter Graph mit positiven Kantengewichten beschrieben werden, wobei die Kanten den Straßen und die Gewichte ihren Längen entsprechen. Der Weihnachtsmann muss jede Straße mindestens einmal durchlaufen und zum Ausgangspunkt zurückkehren. Verständlicherweise möchte er die Gesamtlänge seiner Tour minimieren. Können Sie ihm helfen? (5 Punkte)

**Abgabe der Theorieübungen:** Donnerstag, den 18.12.2014, vor der Vorlesung.

4. Implementieren Sie einen Algorithmus, um das ZUORDNUNGSPROBLEM zu lösen. Das Programm soll Laufzeit  $O(n^3)$  haben, wobei  $n = |V(G)|$  sei. (14 Punkte)

**Abgabe der Programmierübung:** Donnerstag, den 8.1.2015, vor der Vorlesung.

## Hinweise zur Programmierübung:

Das Programm muss in C oder C++ geschrieben sein. Es wird empfohlen, C++ zu verwenden.

Es können wieder alle Programme und Datenstrukturen, die in der Vorlesung "Algorithmische Mathematik I" aus dem Wintersemester 2012/2013 vorgestellt wurden, benutzt oder geeignet modifiziert werden. Alle Programme aus dieser Vorlesung finden Sie hier:

<http://www.or.uni-bonn.de/~vygen/lectures/alma1ws12.html>

Dem Programm muss beim Aufruf der Name einer Datei übergeben werden. Ein Aufruf hat also die Form

`<programmname> <dateiname>`

Eine gültige Datei, die einen Graphen beschreibt, hat das folgende Format:

```
Knotenanzahl
Knoten0a Knoten0b Kosten0
Knoten1a Knoten1b Kosten1
...
```

Die Einträge der Datei sind ausschließlich ganze Zahlen. Sie können voraussetzen, dass die Summe der Absolutbeträge aller Zahlen in der Eingabe kleiner als  $2^{31}$  ist. In der ersten Zeile steht eine einzelne positive gerade Zahl  $n$ , welche die Anzahl der Knoten angibt. Die Knoten werden von 0 bis  $n - 1$  durchnummeriert. Jede folgende Zeile kodiert genau eine Kante. Die ersten beiden Einträge einer Zeile sind die Nummern der Endknoten der Kante (wobei die Kante vom jeweils ersten angegebenen Knoten zum zweiten gerichtet sei). Die erste Nummer liegt dabei in  $\{0, \dots, \frac{n}{2} - 1\}$  und die zweite in  $\{\frac{n}{2}, \dots, n - 1\}$ . Der dritte Eintrag in der Zeile gibt die Kosten der Kante an.

Parallele Kanten kommen in den Instanzen nicht vor.

**Ausgabeformat:** Wenn das Zuordnungsproblem lösbar ist, muss das Programm in der ersten Zeile der Ausgabe die Kosten der berechneten Lösung ausgeben. Danach folgen  $\frac{n}{2}$  weitere Zeilen, die jeweils genau zwei Zahlen enthalten und eine Kante der berechneten Lösung kodieren. Die Nummern in der jeweiligen Zeile geben dabei die Nummern der Endknoten an (wobei die kleinere Nummer zuerst stehen soll). Wenn keine zulässige Lösung existiert, muss das Programm dies melden.

**Beispiel:** Eine Eingabedatei für einen Graphen mit sechs Knoten und sechs Kanten kann so aussehen:

```
6
2 5 37
0 3 12
0 4 -14
1 4 8
2 3 4
1 5 2
```

Die Ausgabe der Programms kann dann so aussehen (die letzten drei Zeilen können anders sortiert sein):

```
-8
0 4
2 3
1 5
```

Das Programm muss korrekt arbeiten und ohne Fehlermeldung kompiliert werden können. Der Code muss auf einem gängigen Linuxsystem funktionieren. Algorithmen aus externen Bibliotheken dürfen nicht verwendet werden.

**Abgabe:** Der Quelltext des Programms muss bis 8. Januar 2015, 16:15 Uhr per E-Mail beim jeweiligen Tutor eingegangen sein. Außerdem ist bis zu diesem Zeitpunkt ein Ausdruck des Quelltextes zusammen mit den Theorieaufgaben abzugeben.

Testinstanzen befinden sich auf der Seite

[http://www.or.uni-bonn.de/lectures/ws14/edm\\_14\\_uebung.html](http://www.or.uni-bonn.de/lectures/ws14/edm_14_uebung.html)