Wintersemester 2008/09
Prof. Dr. J. Vygen
Dr. U. Brenner

Forschungsinstitut für Diskrete Mathematik
Universität Bonn

Einführung in die Diskrete Mathematik
9. Übung

1. Gegeben seien ein gerichteter Graph G und nichtnegative ganze Zahlen $a(x)$, $b(x)$ für jedes $x \in V(G)$. Zeigen Sie: G besitzt genau dann einen aufspannenden Teilgraphen H mit $|\delta_H^+(x)| = a(x)$ und $|\delta_H^-(x)| = b(x)$ für alle $x \in V(G)$, wenn

$$\sum_{x \in V(G)} a(x) = \sum_{x \in V(G)} b(x) \quad \text{und}$$

$$\sum_{x \in X} a(x) \leq \sum_{y \in V(G)} \min\{b(y), |\delta_G^+(X \setminus \{y\}) \cap \delta_G^-(y)|\} \quad \text{für alle } X \subseteq V(G).$$

(4 Punkte)

Hinweis: Betrachten Sie den Graphen $(\{(v,i) : v \in V(G), i = 1,2\}, \{((v,1),(w,2)) : (v,w) \in E(G)\})$, und ergänzen Sie Knoten s und t.

2. Der Weihnachtsmann soll in einer Stadt Geschenke austragen, die nur aus Einbahnstraßen besteht. Diese Stadt kann als stark zusammenhängender gerichteter Graph mit positiven Kantengewichten beschrieben werden, wobei die Kanten den Straßen entsprechen. Jede Straße hat eine bestimmte Länge und kann nur in einer Richtung durchlaufen werden. Der Weihnachtsmann muss jede Straße mindestens einmal durchlaufen und zum Ausgangspunkt zurückkehren. Verständlicherweise möchte er die Gesamtlänge seiner Tour minimieren. Können Sie ihm helfen? (4 Punkte)
Hinweis: Der Weihnachtsmann weiß im Gegensatz zu Ihnen weder über Eulersche Spaziergänge noch über das Minimum-Cost-Flow-Problem Bescheid.

3. Aufgrund eines erst jetzt entdeckten Fehlers im Buchungssystem hat ein großes Hotel für 2009 viele Buchungen angenommen, ohne die Verfügbarkeit freier Zimmer zu prüfen. Jede Buchung betrifft einen bestimmten Zeitraum; es wird aber immer nur ein Zimmer benötigt. Alle Zimmer sind gleichwertig, dennoch wurden die Buchungen zu unterschiedlichen Preisen vorgenommen. Das Hotel möchte nun einigen Kunden

absagen, so dass die freien Zimmer ausreichen, und möglichst wenige Einnahmen verlorengehen. Wie würden Sie dieses Problem lösen? Kann man erreichen, dass kein Gast während seines Aufenthalts umziehen muss? (4 Punkte)
Tipp: Formulieren Sie dies als Minimum-Cost-Flow-Problem, wobei jeder Tag einem Knoten entspricht.

4. Ein Graph heißt *k-regulär*, wenn jeder Knoten Grad k hat. Man beweise, dass ein k-regulärer bipartiter Graph k paarweise disjunkte perfekte Matchings besitzt. Man folgere daraus, dass die Kantenmenge eines bipartiten Graphen mit maximalem Grad k in k Matchings partitioniert werden kann. (4 Punkte)

5. **Programmieraufgabe**
Implementieren Sie den Sukzessive-Kürzeste-Wege-Algorithmus zur Berechnung eines Flusses mit minimalen Kosten. Wir beschränken uns für diese Aufgabe auf **einfache Graphen** und betrachten nur **positive Kantenkosten**. Das Programm soll Laufzeit $O(Bn^2)$ haben, wobei $B = \frac{1}{2}\sum_{v \in V(G)} |b(v)|$ und $n = |V(G)|$ sei.

   Zur Berechnung der kürzesten Wege soll Dijkstras Algorithmus benutzt werden. Dabei genügt eine Implementierung mit Laufzeit $O(n^2)$.

   Einlesen der Daten: 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
   Supply0
   Supply1
   ...
   Kantenanzahl
   Knoten0a Knoten0b Kapazitaet0 Kosten0
   Knoten1a Knoten1b Kapazitaet1 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 Zahl n, welche die Anzahl der Knoten angibt.

Die nächsten n Zeilen spezifizieren die b-Werte der einzelnen Knoten. Jede dieser n Zeilen enthält genau eine Zahl. Wir nehmen an, dass die Knoten von 0 bis $n-1$ durchnummeriert sind. Die Zahl in Zeile i gibt den b-Wert von Knoten $i-2$ an (für $i = 2, \ldots, n+1$). Nach diesen insgesamt $n+1$ Zeilen folgt eine Zeile, die die Kantenzahl angibt. Jede weitere Zeile spezifiziert genau eine Kante. Die ersten beiden Einträge einer Zeile sind zwei verschiedene nichtnegative Zahlen, welche die Nummern der Endknoten der Kante sind (wobei die Kante vom jeweils ersten angegebenen Knoten zum zweiten gerichtet sei). Der dritte Eintrag in der Zeile ist eine positive Zahl, die die Kapazität der Kante bezeichnet. Der vierte und letzte Eintrag in jeder dieser Zeilen ist ebenfalls eine positive Zahl und gibt die Kosten der zugehörigen Kante an. Der Index einer jeden Kante ist durch ihre Zeilennummer in der Eingabedatei gegeben: Zeile i kodiert die Kante mit Index $i-3-n$ (für $i = n+3, \ldots, m+n+2$, wobei m die Zahl der Kanten sei).

Ausgabeformat: Das Programm muss in der ersten Zeile der Ausgabe die Kosten der berechneten Lösung ausgeben. Jede weitere Zeile enthält genau zwei Zahlen. Die erste Zahl ist der Index einer Kante und die zweite Zahl der Fluss-Wert auf dieser Kante. Die Zeilenindizes sollen dabei aufsteigend sortiert sein. Nur Zeilen mit positivem Fluss-Wert sind anzugeben.

Beispiel: Eine Eingabedatei für einen Graphen mit fünf Knoten und sieben Kanten kann so aussehen:

```
5
2
0
0
3
-5
7
0 1 1 1
1 2 1 1
0 2 5 3
0 3 7 5
3 2 3 4
2 4 2 1
3 4 3 2
```

Die Ausgabe der Programms muss dann so aussehen:

```
13
0 1
1 1
2 1
5 2
6 3
```

Das Programm muss in C oder C++ geschrieben sein. Es 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 6. Januar 2009, 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.

(16 Punkte)

6. Man betrachte das Minimum-Cost-Flow-Problem mit möglichen unendlichen Kapazitäten (d.h. $u(e) = \infty$ für manche Kanten e). Eine Instanz (G, u, b, c) heißt *unbeschränkt*, wenn es für jedes $\gamma \in \mathbb{R}$ einen b-Fluss f in (G, u) gibt mit $c(f) < \gamma$.

   (a) Man zeige, dass eine Instanz genau dann unbeschränkt ist, wenn es einen b-Fluss in (G, u) gibt und ein negativer Kreis existiert, dessen Kanten alle unendliche Kapazität haben.

   (b) Man zeige, wie man in $O(n^3 + m)$-Zeit entscheiden kann, ob eine Instanz unbeschränkt ist.

   (c) Man zeige, dass in einer nicht unbeschränkten Instanz jede unendliche Kapazität auf äquivalente Weise durch eine endliche Kapazität ersetzt werden kann.

(8 Zusatzpunkte)

Abgabe: Dienstag, den 6.1.2009, **vor** der Vorlesung.