Eidgenössische
Technische Hochschule
Zürich

Ecole polytechnique fédérale de Zurich
Politecnico federale di Zurigo
Federal Institute of Technology at Zurich

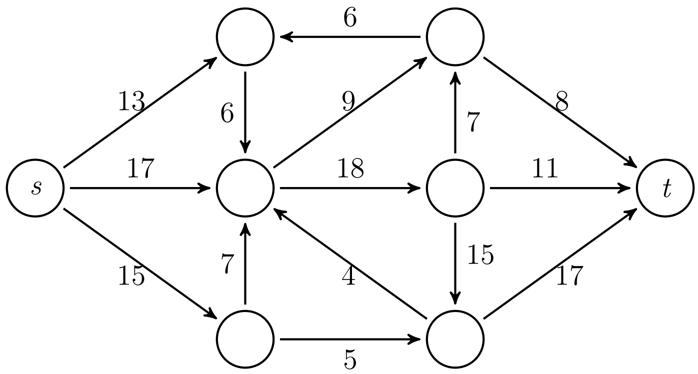
Institut für Theoretische Informatik
Peter Widmayer
Thomas Tschager
Antonis Thomas

18. Mai 2016

Datenstrukturen & Algorithmen Blatt 12 FS 16

Aufgabe 12.1 *Max-Flow von Hand.*

Benutzen Sie den in der Vorlesung vorgestellten Algorithmus von Ford und Fulkerson, um im folgenden Netz einen maximalen Fluss von s nach t zu finden. Die Kapazitäten sind neben den Kanten angegeben. Geben Sie den resultierenden maximalen Fluss, den minimalen Schnitt sowie den entsprechenden Restgraphen an.



Aufgabe 12.2 *Meisterschaftsproblem.*

In der Schweizer Fussball-Nationalliga A galt bis 1995 die sogenannte *2-Punkte-Regel*: Ein Sieg gibt 2 Punkte, ein Unentschieden 1 Punkt und eine Niederlage 0 Punkte. Bei Punktgleichheit wird die bessere Position dem Verein mit der höheren *Tordifferenz*, also der Anzahl erzielter Tore abzüglich der erhaltenen Gegentore, zugeteilt. Angenommen, es gäbe noch genau zwei Spiele zu spielen und die Tabelle sähe wie folgt aus:

	Verein	Punkte	Nächste Gegner
1)	FC St. Gallen (FCSG)	37	FCB, FCW
2)	BSC Young Boys (YB)	36	FCW, FCB
3)	FC Basel (FCB)	35	FCSG, YB
4)	FC Luzern (FCL)	33	FCZ, GCZ
5)	FC Winterthur (FCW)	31	YB, FCSG

Auf den ersten Blick scheint FC Luzern noch Meister werden zu können, wenn beide verbleibende Spiele haushoch gewonnen werden und FC St. Gallen beide verbleibenden Spiele verliert (in diesem Fall hätten beide 37 Punkte und FC Luzern wäre bei besserer Tordifferenz Meister).

Modellieren Sie die obige Situation als Flussproblem und überlegen Sie, welchen Wert ein maximaler Fluss haben muss, damit FC Luzern die Meisterschaft (bei besserer Tordifferenz) gewinnt. Nutzen Sie aus, dass in jedem Spiel genau zwei Punkte vergeben werden. Benutzen Sie das Max-Flow-Min-Cut-Theorem, um zu zeigen, dass FC Luzern in keinem Fall Meister werden kann.

Bemerkung: Eine vergleichbare reale Situation existierte in der deutschen Fussball-Bundesliga 1964/65.

Aufgabe 12.3 *Algorithmus von Edmonds und Karp (**Programmieraufgabe**).*

In dieser Aufgabe soll der Algorithmus von Edmonds und Karp zur Berechnung eines *maximalen Flusses* implementiert werden. Wir definieren einen Fluss wie Sie ihn in der Vorlesung gesehen haben: Sei $G = (V, E, c)$ ein gerichteter Graph mit der Kapazitätsfunktion $c : E \to \mathbb{Q}^+$, der Quelle $s \in V$ sowie der Senke $t \in V$. Ein *Fluss* ist eine Funktion $f : E \to \mathbb{Q}^+$, die folgende Eigenschaften erfüllt:

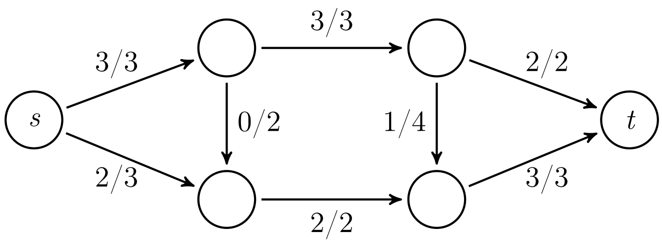
- Der Fluss respektiert die jeweiligen Kapazitäten, d.h. $\forall e \in E : f(e) \le c(e)$,
- Für jeden Knoten ausser der Quelle und der Senke gelten die *Kirchhoffschen Regeln*, d.h.

$$\forall v \in V \setminus \{s, t\} : \sum_{e=(\cdot,v)\in E} f(e) = \sum_{e=(v,\cdot)\in E} f(e).$$

Der Wert des Flusses f ist die Summe aller Flusseinheiten, die die Quelle s verlassen abzüglich der Einheiten, die in s einfliessen, also

$$\mathsf{val}(f) := \left(\sum_{e=(s,\cdot)\in E} f(e) - \sum_{e=(\cdot,s)\in E} f(e) \right).$$

Die Aufgabe besteht nun darin, einen Fluss mit maximalem Wert zu berechnen. Ein Beispiel für einen Graphen und einen maximalen Fluss ist in der untenstehenden Abbildung dargestellt. Neben jeder Kante sind zwei Zahlen dargestellt: Die erste Zahl gibt den Wert des Flusses auf dieser Kante an, das zweite die maximale Kapazität. Der maximale Flusswert beträgt 5.



Eingabe Die erste Zeile der Eingabe enthält lediglich die Anzahl t der Testinstanzen. Danach folgt genau eine Zeile pro Testinstanz. Sie enthält eines Beschreibung des Graphs $G = (V, E)$ im Format $n, m, u_1, v_1, c_1, ..., u_m, v_m, c_m$. Es sind $1 \le n, m \le 100$ mit $V = \{1, ..., n\}$ und $|E| = m$. Für alle i, $1 \le i \le m$, definieren $u_i, v_i \in \{1, ..., n\}$ die Kante $(u_i, v_i) \in E$ mit einer Kapazität von c_i, $1 \le c_i \le 10^7$.

Ausgabe Für jede Testinstanz soll lediglich eine Zeile ausgegeben werden. Sie enthält den Wert einen maximalen Flusses vom Knoten $s = 1$ zum Knoten $t = n$.

Beispiel

Eingabe:

```
2
2 1 1 2 1
6 8 1 2 3 1 3 3 2 3 2 2 4 3 3 5 2 4 5 4 4 6 2 5 6 3
```

Ausgabe:

```
1
5
```

Hinweise Wir stellen eine Code-Vorlage zur Verfügung, in der das Einlesen der Eingabe bereits implementiert wurde. Der Graph ist in Adjazenzlisten-Darstellung in `ArrayList<Integer>[] graph` repräsentiert, d.h. in einem Array der Länge n von ArrayLists mit Ganzzahlen. Jede ArrayList repräsentiert die Adjazenzliste des entsprechenden Knoten. Zusätzlich werden zwei Arrays der Grösse $n \times n$ erstellt: `capacity` enthält die gegebenen maximalen Kapazitäten der Kanten und `flow` die Werte des Flusses auf den entsprechenden Kanten. Implementieren Sie den Algorithmus von Edmonds und Karp. Für diese Aufgabe gibt es lediglich ein Testset.

Abgabe: Am Mittwoch, den 25. Mai 2016 in Ihrer Übungsgruppe.