

## Einführung in die Diskrete Mathematik

### 6. Übung

1. Man nenne einen  $s$ - $t$ -Präfluss  $f$  maximal, wenn  $ex_f(t)$  maximal ist.
  - (a) Man zeige, dass es für jeden maximalen Präfluss  $f$  einen maximalen Fluss  $f'$  mit  $f'(e) \leq f(e)$  für alle  $e \in E(G)$  gibt.
  - (b) Man zeige, wie man in  $O(nm)$  Zeit einen maximalen Präfluss in einen maximalen Fluss umwandeln kann. (2+2 Punkte)
2. Zeigen Sie, dass der Wert eines blockierenden  $s$ - $t$ -Flusses in einem Netzwerk  $(G, u, s, t)$  mit azyklischem gerichtetem Graphen  $G$  um höchstens den Faktor  $|V(G)|$  kleiner ist als der Wert eines maximalen Flusses. Zeigen Sie außerdem, dass diese Schranke bis auf einen konstanten Faktor bestmöglich ist. (5 Punkte)
3. Implementieren Sie den PUSH-RELABEL-ALGORITHMUS zur Berechnung eines maximalen  $s$ - $t$ -Flusses. Ihr Programm sollte höchstens  $O(n^2m)$  Laufzeit haben, idealerweise nur  $O(n^2\sqrt{m})$ . (11 Punkte)

Abgabe: Dienstag, den 26.11.2013, **vor** der Vorlesung.

#### Hinweise zur Programmierübung:

Zum Einlesen und Speichern der Graphen ist die Klasse `Weighted_Graph` aus der Vorlesung "Algorithmische Mathematik I" aus dem Wintersemester 2012/2013 zu verwenden.

**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 eine Instanz beschreibt, hat das folgende Format:

Knotenanzahl

Knoten0a Knoten0b Gewicht0

Knoten1a Knoten1b Gewicht1

...

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 natürliche Zahl (größer als 1), welche die Anzahl der Knoten angibt. Wir nehmen an, dass, wenn wir  $n$  Knoten haben, die Knoten von 0 bis  $n - 1$  durchnummeriert sind. Der Knoten  $s$  hat Nummer 0 und der Knoten  $t$  Nummer 1.

Jede weitere Zeile spezifiziert genau eine Kante. Die ersten beiden Einträge einer Zeile sind zwei verschiedene nichtnegative ganze Zahlen, welche die Nummern der Endknoten der Kante sind. Der dritte Eintrag in der Zeile ist eine positive ganze Zahl, die die Kapazität der Kante bezeichnet. Der Index einer jeden Kante ist durch ihre Zeilennummer in der Eingabedatei gegeben: Zeile  $i$  kodiert die Kante mit Index  $i - 2$  (für  $i = 2, \dots, m + 1$ , wobei  $m$  die Zahl der Kanten sei). Die Kanten sind dadurch auch von 0 bis  $m - 1$  durchnummeriert. Parallele Kanten können vorkommen.

**Ausgabeformat:** Das Programm muss in der ersten Zeile der Ausgabe den Wert eines maximalen  $s$ - $t$ -Flusses ausgeben. Die weiteren Zeilen müssen jeweils genau einen Index einer Kante und den zugehörigen Flusswert enthalten. Es werden dabei nur die Kanten mit positivem Fluss ausgegeben, und die Zeilen sollen nach dem Kantenindex aufsteigend sortiert sein.

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

```
5
0 1 2
2 1 2
2 3 2
3 1 1
0 2 3
3 4 2
```

Die Ausgabe des Programms muss dann so aussehen:

```
5
0 2
1 2
2 1
3 1
4 3
```

**Das Programm muss in C++ geschrieben sein.** Es muss korrekt arbeiten und ohne Fehlermeldung kompiliert werden können. Der Code muss auf einem gängigen Linuxsystem funktionieren. Für die Sortierung der Ausgabe dürfen Sie `qsort` oder `std::sort` verwenden. Andere Algorithmen aus externen Bibliotheken dürfen nicht verwendet werden.

**Abgabe:** Der Quelltext der Programms muss bis 26. November, 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 ab dem 20.11.2013 auf der Seite

[http://www.or.uni-bonn.de/lectures/ws13/edm\\_13\\_uebung.html](http://www.or.uni-bonn.de/lectures/ws13/edm_13_uebung.html)