

Einführung in die Diskrete Mathematik

Programmieraufgabe

In dieser Programmieraufgabe soll der Goldberg-Tarjan-Algorithmus zur Berechnung eines maximalen s - t -Flusses implementiert werden.

Diese Aufgabe wird mit 16 Punkten bewertet.

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  
Kantenzahl  
Knoten0a Knoten0b Kapazitaet0  
Knoten1a Knoten1b Kapazitaet1  
...
```

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 den ersten beiden Zeilen steht jeweils eine einzelne natürliche Zahl, welche in der ersten Zeile die Anzahl der Knoten und in der zweiten die Anzahl der Kanten angibt. 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. Dabei nehmen wir an, daß, wenn wir n Knoten haben, die Knoten von 0 bis $n - 1$ durchnumeriert 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 $i - 3$ (für $i = 3, \dots, m + 2$, wobei m die Zahl der Kanten sei). Die Kanten sind dadurch auch von 0 bis $m - 1$ durchnumeriert. Parallele Kanten können vorkommen.

Sie können voraussetzen, daß es mindestens 2 Knoten gibt. Der Knoten s hat dabei Nummer 0 und der Knoten t Nummer 1.

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
6
0 1 2
2 1 2
2 3 2
3 1 1
0 2 3
3 4 2
```

Die Ausgabe der Programms muß dann so aussehen:

```
5
0 2
1 2
2 1
3 1
4 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. Für die Sortierung dürfen Sie `qsort` oder `std::sort` verwenden. Andere Algorithmen aus externen Bibliotheken dürfen nicht verwendet werden.

Abgabe: Der Quelltext der Programms muß bis **19. Januar, 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.

Weitere Hinweise und Instanzen befinden sich auf der Seite

http://www.or.uni-bonn.de/lectures/ws09/edm_uebung_ws09.html