

Einführung in die Diskrete Mathematik

1. Programmierübung

Schreiben Sie ein Programm, das zu einem gegebenen gerichteten Graphen die starken Zusammenhangskomponenten bestimmt. Das Programm soll lineare Laufzeit haben.

Insbesondere sollen Sie dazu eine Datenstruktur entwickeln, mit der sich gerichtete Graphen effizient verwalten lassen. In späteren Programmieraufgaben werden ebenfalls Algorithmen auf Graphen zu implementieren sein, daher sollte Ihre Datenstruktur so gestaltet sein, daß Sie sie leicht in anderen Algorithmen einsetzen können.

Für diese Programmierübung gibt es 12 Punkte.

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  
Kantenzahl  
Knoten0a Knoten0b  
Knoten1a Knoten1b  
...
```

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 (größer als 0), welche in der ersten Zeile die Anzahl der Knoten und in der zweiten die Anzahl der Kanten angibt. Wir nehmen an, daß, wenn wir n Knoten haben, die Knoten von 0 bis $n - 1$ durchnummeriert sind. 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. Die Kanten sind jeweils vom ersten angegebenen Knoten zum zweiten gerichtet. Es können parallele Kanten, aber keine Schleifen vorkommen. Der Graph muß nicht zusammenhängend sein. Die Sortierung der Kanten in der Eingabedatei kann beliebig sein.

Ausgabeformat: Die erste Zeile der Ausgabe muß genau eine Zahl enthalten, die die Zahl der gefundenen starken Zusammenhangskomponenten angibt. Jede weitere Zeile soll die Nummern der Knoten von einer starken Zusammenhangskomponente enthalten. Aufeinanderfolgende Einträge sollen durch einzelne Leerzeichen getrennt werden. Sortieren Sie die starken Zusammenhangskomponenten aufsteigend nach der kleinsten vorkommenden Knotennummer, und in den einzelnen Zeilen sollen die Knotennummerns ebenfalls aufsteigend sortiert sein.

Beispiel: Eine Eingabedatei für einen Graphen mit 4 Knoten und 5 Kanten kann so aussehen:

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

Die Ausgabe der Programms muß dann so aussehen:

```
2
0 1 3
2
```

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 der Ausgabe dürfen Sie `qsort` oder `std::sort` verwenden. Die Laufzeit dieser Sortier Routinen soll hier nicht betrachtet werden. Andere Algorithmen aus externen Bibliotheken dürfen nicht verwendet werden.

Abgabe: Der Quelltext der Programms muß bis 22. 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 8.11.2011 auf der Seite

http://www.or.uni-bonn.de/lectures/ws11/edm_11_uebung.html