

## Einführung in die Diskrete Mathematik

### 2. Übung

1. Zeigen Sie, dass ein ungerichteter Graph  $G$  genau dann eulersch ist, wenn jede Kante von  $G$  auf einer ungeraden Anzahl von Kreisen liegt. (6 Punkte)  
Hinweis: Um zu zeigen, dass jede Kante auf einer ungeraden Anzahl von Kreisen liegt, wenn der Graph eulersch ist, zeige man zunächst für jede Kante  $e = \{v, w\}$ , dass es in  $G - e$  eine ungerade Anzahl von Spaziergängen von  $v$  nach  $w$  gibt.
2. Sei  $G$  ein stark zusammenhängender gerichteter Graph mit  $n$  Knoten. Zeigen Sie, dass  $G$  dann einen stark zusammenhängenden aufspannenden Teilgraphen mit höchstens  $2n - 2$  Kanten enthält. (2 Punkte)
3. Kann man in linearer Zeit entscheiden, ob ein gegebener gerichteter Graph eine aufspannende Arboreszenz enthält? (4 Punkte)

Abgabe der Theorieaufgaben: Donnerstag, den 23.10.2014, vor der Vorlesung.

#### 4. Programmieraufgabe

Schreiben Sie ein Programm, das zu einem gegebenen gerichteten Graphen die starken Zusammenhangskomponenten bestimmt. Details zur Aufgabenstellung finden sich auf der Rückseite dieses Zettels. (16 Punkte)

Abgabe der Programmieraufgabe: Donnerstag, den 30.10.2014, vor der Vorlesung.

### Zur Programmieraufgabe:

Das Programm muss in C oder C++ geschrieben sein. Es wird empfohlen, C++ zu verwenden. In diesem Fall kann man zum Einlesen und Speichern der Graphen die Klasse `Graph` aus der Vorlesung "Algorithmische Mathematik I" aus dem Wintersemester 2012/2013 verwenden.

Alle Datenstrukturen, die in dieser Vorlesung vorgestellt wurden, finden Sie hier:

<http://www.or.uni-bonn.de/~vygen/lectures/alma1ws12.html>

Sie können alle diese Programme und Datenstrukturen verwenden.

**Eingabeformat:** In der ersten Zeile steht eine einzelne natürliche Zahl (größer als 0 und kleiner als  $2^{31}$ ), welche die Anzahl der Knoten 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 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 vorkommen. Der Graph muss nicht zusammenhängend sein. Die Sortierung der Kanten in der Eingabedatei kann beliebig sein.

**Ausgabeformat:** Die erste Zeile der Ausgabe muss die Zahl der starken Zusammenhangskomponenten enthalten. Jede weitere Zeile soll die Nummern der Knoten von einer starken Zusammenhangskomponente enthalten. Aufeinanderfolgende Einträge sollen durch einzelne Leerzeichen getrennt werden. In den einzelnen Zeilen sollen die Knotennummern aufsteigend sortiert sein.

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

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

Die Ausgabe der Programms kann dann so aussehen (die zweite und die dritte Zeile können auch vertauscht sein):

```
2
0 1 3
2
```

Das Programm 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 30. Oktober, 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 auf der Seite

[http://www.or.uni-bonn.de/lectures/ws14/edm\\_14\\_uebung.html](http://www.or.uni-bonn.de/lectures/ws14/edm_14_uebung.html)

Sollten weitere Hinweise zu der Programmierübung notwendig sein, werden diese ebenfalls auf dieser Homepage bekanntgegeben.