

# Übungen zu Algorithmische Bioinformatik: Netzwerke, Graphen und Systeme

## Blatt 5

**Abgabetermin:** Freitag, 31.05.2019, 9 Uhr

Persönlich oder per Upload-Formular unter

[www.bio.ifi.lmu.de/studium/ss2019/vlg\\_ngs/uebungsabgabe](http://www.bio.ifi.lmu.de/studium/ss2019/vlg_ngs/uebungsabgabe)

### **Aufgabe 1 (Matroid):**

Gegeben sei ein ungerichteter verbundener Graph  $G = (V, E)$ . Sei  $F$  die Menge aller zyklensfreien Subgraphen von  $G$ .  $F$  ist also eine Teilmenge der Potenzmenge von  $E$ .

Zeigen Sie:  $IS = (E, F)$  ist ein Matroid.

### **Aufgabe 2 (Pfadsuche, Bonus-Aufgabe):**

Geben Sie einen rekursiven Algorithmus in Pseudocode an, der in einem gerichteten Graphen alle einfachen Pfade zwischen einem gegebenen Startknoten und einem gegebenen Zielknoten berechnet. Was ist die Laufzeitkomplexität dieses Algorithmus (mit Begründung)?

### **Aufgabe 3 (SCC Algorithmus, Bonus-Aufgabe):**

Implementieren Sie den SCC Algorithmus aus der Vorlesung in *Java*. Die Eingabe ist dabei eine Adjazenzliste wie in dem unten angegebenen Beispiel. Geben Sie alle Zusammenhangskomponenten als Knotenlisten (durch Komma separiert, eine Komponente pro Zeile) in eine Datei aus. Dabei sind die Komponenten in topologischer Ordnung sortiert anzugeben und die Knoten innerhalb der Komponenten lexikographisch sortiert. Verwenden Sie den folgenden gerichteten Beispielgraphen.

```
a -> g
b -> a
c -> d, g
d -> e
e -> d
f -> a, e
g -> b, d, e, f
```

Zu verwendende Commandlineflags: *-f Datei-mit-Adjazenzliste -o Ausgabedatei*

Abzugeben ist dabei das Programm als jar-File, der Source-Code sowie die Ausgabe auf dem Beispielgraphen. Falls zusätzliche Parameter verwendet werden, sind diese in einer README Datei zu spezifizieren.