Algorithmen und
Berechnungskomplexität I
Wintersemester 2016/17

Abgabe: 20.12.2016, 12.30 Uhr

Prof. Dr. Heiko Röglin
Clemens Rösner
Institut für Informatik

Übungsblatt 9

Aufgabe 9.1: (6 Punkte)

Gegeben sei ein ungerichteter Graph $G = (V, E)$ in Adjazenzlistendarstellung. Geben Sie einen Algorithmus an, der in Zeit $O(|V| + |E|)$ eine Bipartition (V_1, V_2) der Knotenmenge V ausgibt, sodass keine Kante $\{u, v\} \in E$ mit $u, v \in V_1$ oder mit $u, v \in V_2$ existiert, oder ausgibt, dass keine solche Bipartition existiert.

Anmerkung: Eine Bipartition (M_1, M_1) einer Menge M ist eine Zerlegung der Menge M in zwei nichtleere, disjunkte Teilmengen M_1 und M_2, d. h. es gilt $M_1, M_2 \neq \emptyset$, $M_1 \cap M_2 = \emptyset$ und $M_1 \cup M_2 = M$.

Aufgabe 9.2: (4+2 Punkte)

Eine Funktion $t\colon V \to \{1, \dots, |V|\}$ auf den Knoten eines gerichteten Graphen $G = (V, E)$ heißt *topologische Sortierung*, falls $t(v) < t(w)$ für alle Kanten (v, w) von G gilt. Veranschaulicht bedeutet das, dass alle Kanten von links nach rechts verlaufen, wenn wir jeden Knoten v an Position $t(v)$ auf der Zahlengerade eintragen.

(a) Betrachten Sie den folgenden Algorithmus zur Ermittlung einer topologischen Sortierung.

Topsort(G, t)
1. Setze count $= 0$.
2. **while** (G ist nicht leer) **do**
3. Finde einen Knoten v in G mit Ingrad 0.
4. **if** ($v =$ **null**) **then return false**
5. Erhöhe count um 1.
6. Setze $t(v) =$ count.
7. Entferne v und alle Kanten der Gestalt (v, w) aus G.
8. **end while**
9. **return true**

Beweisen Sie, dass Topsort eine *topologische Sortierung* von G erzeugt, falls G keinen Kreis besitzt.

Hinweis: Zeigen Sie zunächst, dass jeder azyklische Graph einen Knoten mit Ingrad 0 besitzt. Dazu können Sie die Kontraposition dieser Aussage zeigen: Wenn der Ingrad jedes Knotens mindestens 1 ist, dann besitzt der Graph einen Kreis. Überlegen Sie sich einen konstruktiven Beweis dieser Aussage.

(b) Beweisen Sie, dass G genau dann eine topologische Sortierung besitzt, wenn G keinen Kreis enthält.

Aufgabe 9.3: (3+3 Punkte)

Sei $G = (V, E)$ ein gerichteter Graph. Wir betrachten eine Tiefensuche auf G. Zeigen oder widerlegen Sie die folgenden Aussagen:

(a) Für alle Knoten $u, v \in V$ gilt: Gibt es einen Weg von u nach v in G, dann gilt $v.d \leq u.f$.

(b) Für jeden Knoten $u \in V$ mit In- und Outgrad mindestens 1 gibt es einen Knoten $v \in V$, sodass (u, v) oder (v, u) eine T-Kante ist.

Aufgabe 9.4:

(2+2+2 Punkte)

Implementieren Sie einen Algorithmus, der für einen gegebenen gerichteten Graphen $G = (V, E)$ einen einfachen Kreis ausgibt, falls einer existiert, und sonst ausgibt, dass kein solcher Kreis existiert. Wenden Sie Ihren Algorithmus auf die Instanzen

(a) `Ue9_small.txt` mit $|V| = 20$,

(b) `Ue9_medium.txt` mit $|V| = 100$ und

(c) `Ue9_large.txt` mit $|V| = 500$

an, die auf der Homepage heruntergeladen werden können. Das Dateiformat ist dort ebenfalls erklärt.

Geben Sie jeweils einen einfachen Kreis an. Enthält einer der Graphen keinen solchen Kreis, dann geben Sie das ebenfalls an.

Aufgabe 9.5:

(6 Zusatzpunkte)

Gegeben sei ein gerichteter Graph $G = (V, E)$ in Adjazenzlisten- oder Adjazenzmatrixdarstellung. Geben Sie einen Algorithmus an, der in Zeit $O\big(|V| \cdot (|V| + |E|)\big)$ zwei verschiedene einfache gerichtete Wege ausgibt, deren Anfangs- und Endknoten übereinstimmen, oder ausgibt, dass keine zwei solchen Wege existieren.