Prof. Dr. A. Goerdt
Julian Pape-Lange

TU Chemnitz
Wintersemester 2019/2020
22.10.2019

Theoretische Informatik I

2. Übung

Abgabe: Lösen Sie Aufgabe **1**. Ihre Lösungen geben Sie bitte entweder

- vor oder nach der Vorlesung am 29.10.2019 oder
- bis zum 29.10.2019 um 13:00 Uhr per Mail
  an `julian.pape-lange@informatik.tu-chemnitz.de`
  mit *Betreff:* `TI1 Hausaufgaben`

ab.
Bitte geben Sie klar erkennbar an, an welcher Übungsgruppe Sie teilnehmen werden.

1. Aufgabe: ((3+3+4)P)
Sei $G = (V, E)$ ein *gerichteter Graph* in Adjazenzlistendarstellung. Welche der folgenden Aussagen sind wahr? Begründen Sie Ihre Aussage durch *Beispiel* (Adjazenzlisten und Ablauf der Breitensuche) oder *Gegenbeweis*.

(a) Es ist möglich, dass es eine Kante $(u, v) \in E$ gibt, so dass gilt: Bei der Expansion von u während $\texttt{BFS(}G\texttt{,}s\texttt{)}$ ist v – je nach Adjazenzlistendarstellung von G – *grau* oder *schwarz*.

(b) Es ist möglich, dass es eine Kante $(u, v) \in E$ gibt, so dass gilt: Bei der Expansion von u während $\texttt{BFS(}G\texttt{,}s\texttt{)}$ ist v – je nach Adjazenzlistendarstellung von G – *weiß* oder *grau*.

(c) Es ist möglich, dass es eine Kante $(u, v) \in E$ gibt, so dass gilt: Bei der Expansion von u während $\texttt{BFS(}G\texttt{,}s\texttt{)}$ ist v – je nach Adjazenzlistendarstellung von G – *weiß* oder *schwarz*.

2. Aufgabe: Geben Sie einen gerichteten Graphen $G = (V, E)$ mit einem Knoten $s \in V$ an, so dass folgende Bedingungen gelten.

(a) G enthält einen Baum B mit Wurzel s.

(b) Für jeden Knoten $v \in V$ ist $\texttt{Dist}_B(s, v) = \texttt{Dist}_G(s, v)$, d.h. der Weg von s nach v in B ist genauso lang wie der kürzeste Weg von s nach v in G.

(c) Trotzdem kann der Baum B niemals bei $\texttt{BFS(}G\texttt{,}s\texttt{)}$ als Breitensuchbaum entstehen.

3. Aufgabe: Sei $(v_1, \ldots, v_n)$ eine Folge von n Knoten. Wieviele gerichtete Graphen $G = (V, E)$ mit $V = \{v_1, \ldots, v_n\}$ haben diese Folge als topologische Sortierung?

4. Aufgabe: Wir betrachten den Algorithmus zur *topologischen Sortierung* aus der Vorlesung.

```
1. /** Initialisierung */
   EGrad[v] auf 0, Q = ();
   for each v ∈ V {
       Gehe Adj[v] durch,
       zähle für jedes gefundene u
       EGrad[u] = EGrad[u]+1;
   }

2. /** Einfügen in Schlange */
   for each u ∈ V {
       if EGrad[u] = 0
           Q = enqueue(Q,u)
   }

3. /** Array durchlaufen */
   for i = 1 to n {
       if Q = ()
           Ausgabe „Kreis“; return;

4. /** Knoten aus Schlange betrachten*/
       v[i] = Q[head];
       Q = dequeue(Q);

5. /** Adjazenzliste durchlaufen*/
       for each u ∈ Adj[v[i]] {
           EGrad[u] = EGrad[u]-1;
           if EGrad[u] = 0
               Q = enqueue(Q,u)
       }
   }
```

Sei $G = (V, E)$ der eingegebene gerichtete Graph in Adjazenzlistendarstellung.

(a) Wir bezeichnen mit $\texttt{EGrad}_i\texttt{[}u\texttt{]}$ den Wert von $\texttt{EGrad[}u\texttt{]}$ nach dem i-ten Durchlauf der Hauptschleife. Zeigen Sie die Korrektheit der folgenden Invariante:

$$\texttt{EGrad}_i\texttt{[}u\texttt{]} = \texttt{EGrad}_0\texttt{[}u\texttt{]} - \sum_{j=1}^{i} \texttt{A[v[}j\texttt{]},u\texttt{]}$$

Wobei $\texttt{v}[j]$ den Knoten bezeichnet, der im j-ten Durchlauf entfernt – also einsortiert – wird. `A` ist die Adjazenzmatrix, die jedoch nicht im Algorithmus benutzt wird.

Der `EGrad` eines Knotens u verringert sich also bei jedem Durchlauf der Schleife um die Anzahl der bereits einsortierten Knoten, die mit ihm verbunden sind.

(b) Folgern Sie aus (a): Wenn die Hauptschleife n-mal ohne die Ausgabe „Kreis“ durchlaufen wurde, ist $(\texttt{v[1]}, \texttt{v[2]}, \ldots, \texttt{v[}n\texttt{]})$ eine topologische Sortierung von G.

(c) Folgern Sie aus (a): Wenn der Algorithmus „Kreis“ ausgibt, so enthält G tatsächlich einen Kreis.

(d) Betrachten Sie die Modifikation des Algorithmus, dass jeweils ein Knoten mit Ausgangsgrad `AGrad` $= 0$ gelöscht und die topologische Sortierung in umgekehrter Reihenfolge konstruiert wird.

Welche Vor- und Nachteile ergeben sich daraus?