

Theoretische Informatik I

2. Übung

Geben Sie die Lösung der Aufgabe 1 bitte bis zum 23.10.2017 9:15 Uhr ab. (Briefkasten vorm Raum 1/266 oder per eMail an fal@informatik.tu-chemnitz.de, *Betreff: TI1 Hausaufgaben*)

1. Aufgabe: 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 $\text{BFS}(G, s)$ 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 $\text{BFS}(G, s)$ 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 $\text{BFS}(G, s)$ 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 $\text{Dist}_B(s, v) = \text{Dist}_G(s, v)$, 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 $\text{BFS}(G, s)$ als Breitensuchbaum entstehen.

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

4. **Aufgabe:** Wir betrachten den Algorithmus *Verbessertes TopSort* 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 $EGrad_i[u]$ den Wert von $EGrad[u]$ nach dem i -ten Durchlauf der Hauptschleife. Zeigen Sie die Korrektheit der folgenden Invariante:

$$EGrad_i[u] = EGrad_0[u] - \sum_{j=1}^i A[v[j], u]$$

Wobei $v[j]$ 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 $(v[1], v[2], \dots, v[n])$ 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?