Algorithmen und
Berechnungskomplexität I
Wintersemester 2016/17

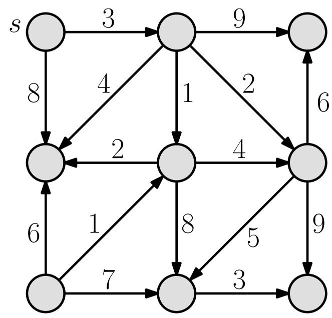
Abgabe: 24.01.2017, 12.30 Uhr

Prof. Dr. Heiko Röglin
Clemens Rösner
Institut für Informatik

Übungsblatt 12

Aufgabe 12.1: (4+2 Punkte)

(a) Bestimmen Sie mithilfe des Algorithmus von Dijkstra für den folgenden Graphen die Werte $\delta(s, v)$ für alle Knoten v und geben Sie einen Kürzeste-Wege-Baum mit Wurzel s an.



(b) Zeigen Sie, dass der Algorithmus von Dijkstra die Abstände $\delta(s, v)$ für Graphen mit negativen Kantengewichten im Allgemeinen nicht korrekt berechnet, auch wenn diese keinen negativen Kreis enthalten.

Aufgabe 12.2: (6 Punkte)

Sei $G = (V, E)$ ein gerichteter Graph mit einer nichtnegativen Kantengewichtung w. Wir führen den Algorithmus von Dijkstra für einen Knoten $s \in V$ aus. Ohne Einschränkung sei jeder Knoten des Graphen von s aus erreichbar. Mit v_i bezeichnen wir den Knoten von G, der als i-ter Knoten zur Menge S hinzugefügt wird. Zeigen Sie induktiv, dass $\delta(s, v_i) \leq \delta(s, v_{i+1})$ für alle $i = 1, \ldots, n-1$ gilt, d. h. zu keinem Zeitpunkt liegt ein Knoten außerhalb von S näher an s als irgendein Knoten in S.

Hinweis: Nutzen Sie die Invariante aus dem Beweis von Theorem 5.23.

Aufgabe 12.3: (6 Punkte)

Implementieren Sie den Algorithmus von Dijkstra und den Floyd-Warshall-Algorithmus.

(a) Wenden Sie den Algorithmus von Dijkstra mit Startknoten $s = 0$ auf die Instanz `Ue12a.txt` an. Geben Sie die 10 Knoten mit dem größten Abstand zu s und deren Abstände zu s an.

(b) Wenden Sie den Floyd-Warshall-Algorithmus auf die Instanz `Ue12b.txt` an. Geben Sie die 10 Paare von Knoten mit den größten Abständen zueinander und die entsprechenden Abstände an.

(c) Bestimmen Sie einen einfachen negativen Kreis in der Instanz `Ue12c.txt`.

Die Dateien `Ue12a.txt`, `Ue12b.txt` und `Ue12c.txt` sowie die Erklärung des Dateiformates können auf der Homepage heruntergeladen werden.

Aufgabe 12.4: (6 Punkte)

Wir betrachten das Single-Source Shortest Path Problem auf Graphen mit beliebigen Kantengewichten, aber ohne negative Kreise. Wie wir in Aufgabe 12.1 (b) gezeigt haben, können wir den Algorithmus von Dijkstra nicht anwenden. Der Floyd-Warshall-Algorithmus hingegen besitzt mit $\Theta(|V|^3)$ eine sehr hohe Laufzeit.

Geben Sie einen Algorithmus an, der obiges Problem in Zeit $\Theta\big(|V| \cdot (|V| + |E|)\big)$ löst, wenn der Graph in Adjazenzlisten- oder Adjazenzmatrixdarstellung gegeben ist.

Hinweis: Verwenden Sie dynamische Programmierung.

Aufgabe 12.5: (6 Zusatzpunkte)

Arbitrage bezeichnet das Ausnutzen von Preisunterschieden für gleiche Waren auf verschiedenen Märkten. Ein einfaches Beispiel ist das Ausnutzen von Wechselkursen, also das Handeln mit Währungen. Angenommen, ein Händler startet mit 1\$ (Dollar). Damit kauft er zunächst 95.739¥ (Yen). Pro Yen erhält er 0.0063£ (britische Pfund). Schließlich kauft er zum Kurs 1.6583\$/£ wieder Dollar und hat jetzt $95.739 \cdot 0.0063 \cdot 1.6583 = 1.0002131$ Dollar (Wechselkurse vom 02.06.2009). Insgesamt ergab sich also ein Gewinn von 0.021%.

Gegeben seien n Währungen $c_1, \ldots, c_n$ und eine $(n \times n)$-Matrix R, die die aktuellen Wechselkurse enthält. Die Einträge von R sind positiv und auf der Hauptdiagonalen stehen Einsen. Geben Sie einen Algorithmus an, der für eine Wechselkursmatrix R in Zeit $O(n^k)$ bestimmt, ob ein Arbitrage-Geschäft möglich ist. Dabei soll k eine geeignete Konstante sein.