Übungen zu Algorithmen | **Sommersemester 2018**
WSI für Informatik | **Universität Tübingen**
Kaufmann/Bekos/Schneck | **29.06.2018**

Übungsblatt 10

Abgabe bis 05.07.2018

Besprechung: 09.07.2018 – 12.07.2018

Aufgabe 1: Grapheigenschaften (2 + 2 + 1 + 1 Punkte)

Gegeben sei ein einfacher, zusammenhängender Graph $G = (V, E)$ mit $|V| = n$ und $|E| = m$.

(a) Zeigen Sie, dass es in G zwei Knoten mit demselben Grad gibt.

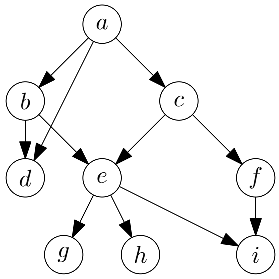
(b) Zeigen Sie, dass G eine gerade Anzahl an Knoten mit ungeradem Grad hat.

(c) G heiß *vollständig*, wenn jeder Knoten durch Kanten mit jedem anderen Knoten verbunden ist. Berechnen Sie die Anzahl der Kanten in einem vollständigen Graphen.

(d) G heißt *d-regulär*, wenn jeder Knoten aus V einen Grad von d hat. Zeigen Sie, dass $m = \frac{d \cdot n}{2}$, falls G d-regulär ist.

Aufgabe 2: Adjazenzmatrix vs. Adjazenzliste (2 + 1 + 4 + 4 Punkte)

Sei $G = (V, E)$ der folgende gerichtete Graph:



(a) Berechnen Sie die Adjazenzmatrix und die Adjazenzliste für G.

(b) Wenden Sie den Algorithmus aus der Vorlesung auf Ihre Adjazenzliste aus (a) an, um eine topologische Sortierung von G zu berechnen.

(c) Der *transponierte Graph* eines Graphen $G = (V, E)$ ist der Graph $G' = (V, E')$, wobei

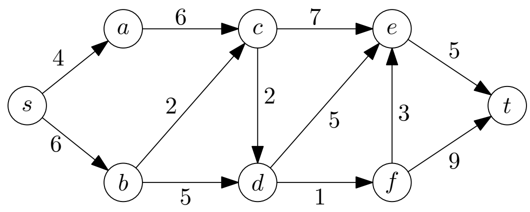
$$E' := \{(v, u) \mid (u, v) \in E\}.$$

Nehmen Sie an, dass G als Adjazenzmatrix vorliegt. Beschreiben Sie einen Algorithmus, der den transponierten Graph G' in einer Laufzeit von $\mathcal{O}(|V|^2)$ berechnet.

(d) Nehmen Sie nun an, dass G als Adjazenzliste vorliegt. Beschreiben Sie einen Algorithmus, der den transponierten Graph G' in einer Laufzeit von $\mathcal{O}(|V| + |E|)$ berechnet.

Aufgabe 3: Dijkstra-Algorithmus (5 Punkte)

Wenden Sie den Dijkstra-Algorithmus auf das folgende Beispiel an. Verwenden Sie dabei s als Startknoten und geben Sie alle Zwischenschritte an!



Aufgabe 4: Anwendung von Dijkstra (1 + 2 + 5 Punkte)

Betrachten Sie ein gerichtetes Netzwerk $G = (V, E)$ zusammen mit einer Wahrscheinlichkeitsfunktion $p : E \to (0, 1]$. Dabei ist $0 < p(v, w) \leq 1$ die *Zuverlässigkeit* der Kante (v, w). Seien $s, t \in V$ Knoten, zwischen denen ein Packet gesendet werden soll. Ob das Packet erfolgreich gesendet wird, hängt von der Zuverlässigkeit des verwendeten Pfades ab. Dabei beschreibt die *Zuverlässigkeit eines Pfades* die Wahrscheinlichkeit, dass ein Packet erfolgreich über einen Pfad gesendet werden kann.

(a) Wie lässt sich die Zuverlässigkeit eines Pfades $s = v_0 \to v_1 \to \cdots \to v_k = t$ berechnen?

(b) Geben Sie ein Beispiel an, bei dem der Dijkstra-Algorithmus nicht den Pfad von s nach t mit der höchsten Zuverlässigkeit findet, wenn man für jede Kante (v, w) das Kantengewicht $1 - p(v, w)$ (d. h. die Ausfallswahrscheinlichkeit) verwendet.

(c) Beschreiben Sie einen Algorithmus, der mit Hilfe des Dijkstra-Algorithmus, den Pfad zwischen s und t mit der höchsten Zuverlässigkeit findet. Begründen Sie die Laufzeit und Korrektheit Ihres Algorithmus.