Diskrete und geometrische Algorithmen
Übung 9

11. Dezember 2023

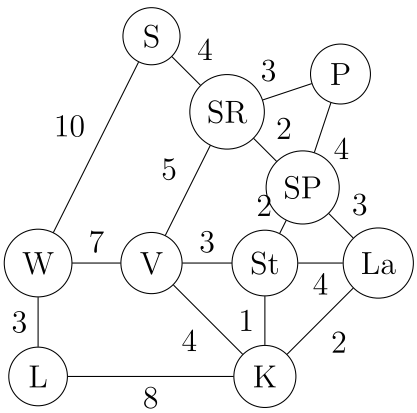
1. Sei G ein stark zusammenhängender gerichteter Graph. Ein eulerscher Kreis in G ist ein Kreis, der jede Kante genau einmal benutzt. Knoten dürfen mehrmals besucht werden.

   (a) Zeigen Sie, dass G genau dann einen Eulerschen Kreis besitzt, wenn in jedem Knoten der Eingangs- und der Ausgangsgrad gleich groß ist.

   (b) Beschreiben Sie einen Algorithmus mit Laufzeitverhalten $O(|E|)$, der einen Eulerschen Kreis findet, wenn einer existiert.

2. Eine Studentin fährt mit der U-Bahn (Graph des U-Bahn-Netzes siehe unten, die Gewichte der Kanten geben die Fahrzeit in Minuten an) von der TU am Karlsplatz (Knoten K im Graphen) nach Hause zu ihrer Wohnung in Spittelau (Knoten S im Graphen) und möchte dafür so wenig Zeit wie möglich brauchen.

   (a) Man bestimme mithilfe des Algorithmus von Dijkstra den kürzesten Weg unter der (zugegeben etwas unrealistischen) Annahme, dass zum Umsteigen keine Zeit benötigt wird.



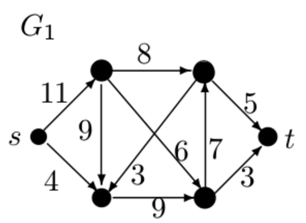
   (b) Wie könnte man die beim Umsteigen benötigte Zeit im Modell berücksichtigen? (Hinweis: Fügen Sie geeignete Knoten und Kanten hinzu.)

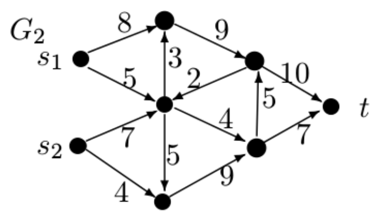
3. Geben Sie eine (nicht notwendigerweise effiziente) Variante des Bellman-Ford-Algorithmus an, die folgendes leistet:

   - kann der Knoten v von s aus durch einen Pfad (also eine Folge gerichteter Kanten) erreicht werden, der einen Zyklus mit negativem Gewicht enthält, dann sei $v.d = -\infty$,
   - ansonsten sei $v.d$ die minimale Distanz zu s.

4. Sei $G = (V, E)$ ein gerichteter Graph mit Gewichtsfunktion $w : E \to \mathbb{R}$ und $v_0 \in V$. Der Graph werde mit $\textsc{Init}(G, v_0)$ initialisiert und dann werde eine Folge von Relaxationsschritten ausgeführt, sodass danach $\pi(v_0) \neq 0$ gelte. Beweisen Sie, dass G einen Zyklus mit negativen Gewicht besitzt.

5. Bestimmen Sie im Netzwerk G_1 (siehe unten) mittels Algorithmus von Ford und Fulkerson einen maximalen Fluss. Geben Sie auch einen minimalen Schnitt an.





6. Bestimmen Sie im Netzwerk G_2 (siehe oben) mit zwei Quellen s_1 und s_2 mithilfe des Algorithmus von Ford und Fulkerson einen maximalen Fluss.
   Hinweis: Ergänzen Sie G_2 durch eine virtuelle Quelle s, die vor den Quellen s_1 und s_2 liegt.