Datenstrukturen und Algorithmen
Bonus-Übung 12, Frühling 2011

19. Mai 2011

Diese Übung muss zu Beginn der Übungsstunde bis spätestens um 16 Uhr 15 am 26. Mai abgegeben werden. Die Abgabe der DA Übungen erfolgt immer in schriftlicher Form auf Papier. Programme müssen zusammen mit der von ihnen erzeugten Ausgabe abgegeben werden. Drucken Sie wenn möglich platzsparend 2 Seiten auf eine A4-Seite aus. Falls Sie mehrere Blätter abgeben heften Sie diese bitte zusammen (Büroklammer, Bostitch, Mäppchen). Alle Teilnehmenden dieser Vorlesung müssen Ihre eigene Übung abgeben (einzeln oder in Zweiergruppen). Vergessen Sie nicht, Ihren Namen und Ihre Matrikelnummer auf Ihrer Abgabe zu vermerken.

Theoretische Aufgaben

1. Erklären Sie, ob folgende Variation des Algorithmus für starke Zusammenhangskomponenten korrekt ist oder nicht: In der zweiten Tiefensuche wird der ursprüngliche Graph (statt der transponierte Graph) verwendet, und die Knoten werden gemäss aufsteigender (statt absteigender) Endzeiten bearbeitet. **2 Punkte**

2. Sei (u, v) eine Kante mit minimalem Gewicht in einem Graphen G. Zeigen Sie, dass (u, v) zu einem minimalen Spannbaum von G gehört, ein Mal mit Hilfe des Kruskal-Algorithmus und ein Mal mit Hilfe des Prim-Algorithmus. **2 Punkte**

3. Sei (u, v) eine Kante in einem MST eines Graphen. Zeigen Sie, dass es einen Schnitt von G gibt, so dass (u, v) eine leichte Kante ist. **1 Punkt**

4. Betrachten Sie den gewichteten Graphen in der Abbildung 1. Geben Sie eine Sequenz der Kanten an, die mit Kruskals Algorithmus zu einem minimalen Spannbaum hinzugefügt werden. **1 Punkt**

5. Betrachten Sie den gewichteten Graphen in der Abbildung 1. Geben Sie eine Sequenz der Kanten an, die mit Prims Algorithmus zu einem minimalen Spannbaum hinzugefügt werden. **1 Punkt**

6. Zeigen Sie ähnlich wie in Abbildung 24.6. wie der Dijkstra-Algorithmus auf dem in Abbildung 2 gezeigten Graphen arbeitet. Der Startknoten sei dabei a. **1 Punkt**

7. Sei $G = (V, E)$ ein gewichteter, gerichteter Graph ohne Zyklen mit negativem Gewicht und m das Maximum der minimalen Anzahl von Kanten in einem kürzesten Pfad von

u nach v über alle Knotenpaare $u, v \in V$. (Gemeint ist hier der "kürzeste" Pfad dem Gewicht nach, nicht nach der Anzahl der Kanten.) Schlagen Sie eine einfache Änderung des Bellman-Ford-Algorithmus vor, durch die er nach $m + 1$ Durchläufen terminieren kann, auch wenn m im Voraus nicht bekannt ist. **1 Punkt**

8. Angenommen, wir ändern Zeile 4 des Dijkstra-Algorithmus in **while** $|Q| > 1$. Diese Änderung bewirkt, dass die **while**-Schleife nur $|V| - 1$-mal statt $|V|$-mal durchgeführt wird. Arbeitet der Algorithmus mit der vorgeschlagenen Änderung korrekt? **1 Punkt**

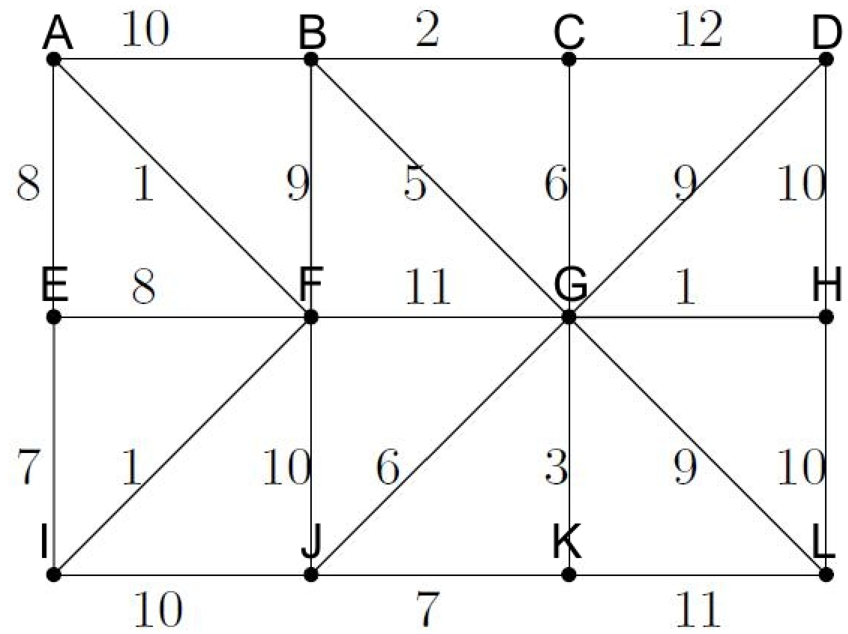

Abbildung 1: Graph für Aufgaben 4 und 5.

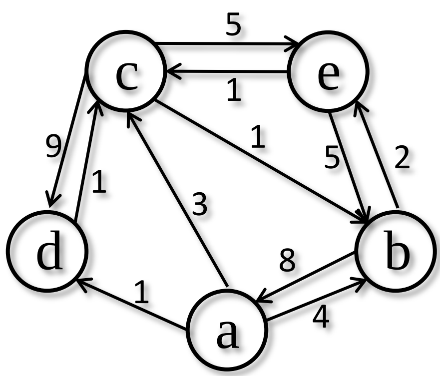

Abbildung 2: Graph für Aufgabe 6