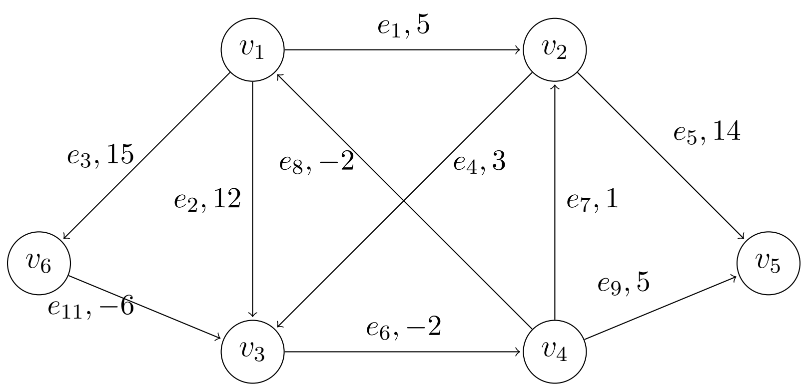
Abteilung Algorithmik
Institut für Betriebssysteme und Rechnerverbund **SoSe 15**
TU Braunschweig

Prof. Dr. Sándor P. Fekete
Frank Quedenfeld

Netzwerkalgorithmen
Übung 4 vom 08.06.2015

Abgabe der Lösungen bis Montag,
den 22.06.15, 13:00 Uhr im Hausaufgabenrückgabeschrank.
Bitte die Blätter zusammenheften und vorne deutlich mit eigenem Namen,
Matrikel- und Gruppennummer, sowie Studiengang versehen!

Aufgabe 1 (Algorithmus von Moore, Bellman und Ford):



Bestimme mit dem Moore-Bellman-Ford-Algorithmus einen kürzesten Pfad von v_1 nach v_5. Gib jeweils an, wenn sich Längenwerte oder Vorgänger ändern.

(15 Punkte)

Aufgabe 2 (Wege und Schnitte):
Sei G ein ungerichteter Graph mit Gewichten $c : E(G) \to \mathbb{Z}_+$ und zwei Knoten $s, t \in V(G)$, wobei t von s aus erreichbar ist. Für eine Knotenmenge $X \subseteq V(G)$ nennt man die Kantenmenge $\delta(X) = \{\{x, y\} \in E(G), x \in X, y \in V(G) \setminus X\}$ einen Schnitt in G. Falls $s \in X$ und $t \notin X$ gilt, trennt $\delta(X)$ die beiden Knoten s und t.
Zeige, dass die minimale Länge eines s-t Weges gleich der maximalen Anzahl von Schnitten ist, die s und t trennen, so dass jede Kante e in höchstens $c(e)$ solcher Schnitte enthalten ist.
(Tipp: Warum reicht es, einen Graphen mit Einheitsgewichten zu betrachten? Zeige, dass die maximale Anzahl von solchen Schnitten sowohl eine obere als auch eine untere Schranke für die minimale Länge ist. Für den Beweis der unteren Schranke gib eine Menge von Schnitten an, unter der Verwendung eines BFS-Baumes.)

(15 Punkte)

Aufgabe 3 (Knapsack mit Wiederholung):
In der großen Übung am 08.06. haben wir das Knapsackproblem und eine Formulierung als dynamische Programm für den Fall, das jeweils nur ein Gegenstand einer Sorte da ist, betrachtet.
Stelle das generelle Knapsackproblem mit positiven Gewichten und nichtnegativer Anzahl von Gegenständen als dynamisches Program dar.

(15 Punkte)

Aufgabe 4 (Kürzeste Wege zwischen allen Knoten):
Betrachte das folgende Problem:
ALLE KÜRZESTEN WEGE
Gegeben: Gerichteter Graph G, konservative Gewichte $c : E(G) \to \mathbb{R}$.
Gesucht: Kürzeste Wege für alle $s, t \in V(G)$, d.h.

l_{st}: Länge des kürzesten $s - t-$Pfades
p_{st}: Vorgänger von t in einem kürzesten $s - t-$Pfad.

(a) Welche Laufzeit liefert Lösung des Problems mittels wiederholter Anwendung von Moore-Bellman-Ford?

(b) Betrachte den Algorithmus von Floyd und Warshall, Algorithmus 1 auf Seite 3 dieses Blattes. Welche Laufzeit hat der Algorithmus?

(c) Beweise die Korrektheit des Algorithmus von Floyd und Warshall. Hinweis: Zeige die folgende Behauptung:
Nach Durchlaufen der äußeren Schleife mit $j = 1, \ldots, j_0$ enthält die Variable l_{ik} die Länge eines kürzesten $i - k-$Pfades, der nur die Zwischenknoten $1, \ldots, j_0$ benutzt; (p_{ik}, k) ist die letzte Kante eines solchen Pfades.

(2+3+10 Punkte)

Algorithm 1: Floyd, Warshall (1962)

Input : Digraph G mit konservativen Gewichten, $c : E(G) \to \mathbb{R}$

Output: Für jedes Knotenpaar $i, j \in V(G)$ die Angaben: l_{ij}: Länge des kürzesten $i - j-$Pfades, p_{ij}: Vorgänger von j in einem kürzesten Pfad.

Betrachte $V(G) = \{1, \ldots, n\}$

1.

$l_{ij} := c((i, j))$ für alle $(i, j) \in E(G)$

$l_{ij} := \infty$ für alle $(i, j) \in V(G)^2 \backslash E(G), i \neq j$

$l_{ii} := 0$ für alle $i \in V(G)$

$p_{ij} := i$ für alle $(i, j) \in E(G)$

2.

for $j = 1$ **to** n **do**

  for $i = 1$ **to** n **do**

    if $i \neq j$ **then**

      for $k = 1$ **to** n **do**

        if $k \neq j$ **then**

          if $l_{ik} > l_{ij} + l_{jk}$ **then**

            $l_{ik} := l_{ij} + l_{jk}$;

            $p_{ik} := p_{jk}$