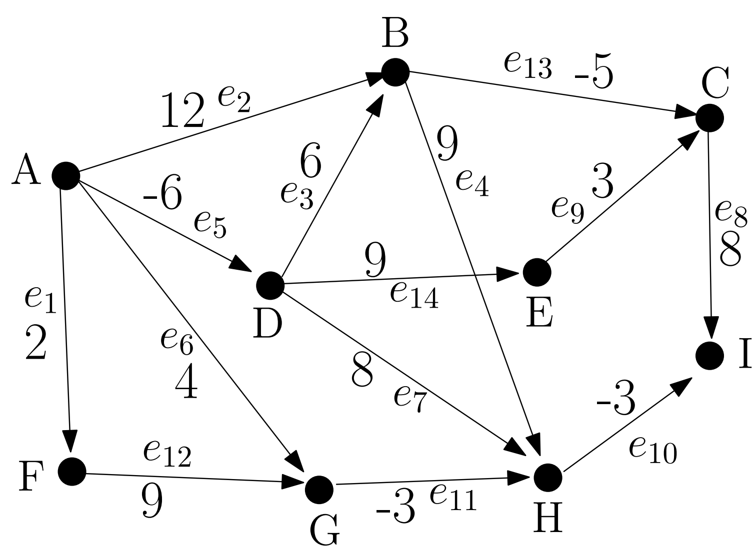
Abteilung Algorithmik
Institut für Betriebssysteme und Rechnerverbund **SoSe 11**
TU Braunschweig

Prof. Dr. Sándor P. Fekete
Christiane Schmidt

Netzwerkalgorithmen
Übung 3 vom 23.05.2011

Abgabe der Lösungen bis Montag, den 06.06.11, bis 13:00 Uhr in der
Abteilung *Algorithmik* (gegenüber IZ 252).
Bitte die Blätter vorne deutlich mit eigenem Namen und Gruppennummer versehen!

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



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

(15 Punkte)

Aufgabe 2 (Zweitkürzester Pfad):
Gegeben sei ein Digraph G mit Gewichten $c : E(G) \to \mathbb{R}_+$, und zwei Knoten $s, t \in V(G)$. Der kürzeste Pfad P von s nach t sei eindeutig. Kann man den kürzesten, von P verschiedenen Pfad von s nach t in polynomieller Zeit bestimmen? Begründe die Korrektheit und die Laufzeit des von Dir angegebenen Verfahrens.

(15 Punkte)

Aufgabe 3 (Kürzeste Wege zwischen allen Knoten):

Betrachte das folgende Problem:
ALLE KÜRZESTE 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) Lösung des Problems mittles wiederholter Anwendung von Moore-Bellman-Ford liefert welche Laufzeit?

(b) Betrachte den Algorithmus von Floyd und Warshall, Algorithmus 1. 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.

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 $(i, j) \in V(G)^2 \setminus E(G), i \neq j$
$l_{ii} := 0$ 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**
            $l_{ik} := l_{ij} + l_{jk}$;
            $p_{ik} := p_{jk}$

(2+3+10 Punkte)

Aufgabe 4 (Pfade und Zuverlässigkeit):
Gegeben sei ein Digraph G mit $s, t \in V(G)$. Jeder Kante $e \in E(G)$ wird eine Zahl $r(e) \in [0, 1]$, ihre Zuverlässigkeit, zugewiesen. Die Zuverlässigkeit eines Pfades ist das Produkt der Zuverlässigkeiten seiner Kanten. Gesucht ist der Pfad von s nach t mit maximaler Zuverlässigkeit.

a) Zeige, dass man dieses Problem unter Anwendung von Logarithmen auf das Kürzeste-Wege-Problem reduzieren kann.

b) Gib einen Algorithmus an, der das Problem in polynomieller Zeit ohne die Anwendung von Logarithmen löst.

(8+7 Punkte)