Abteilung Algorithmik
Institut für Betriebssysteme und Rechnerverbund **SoSe 12**
TU Braunschweig

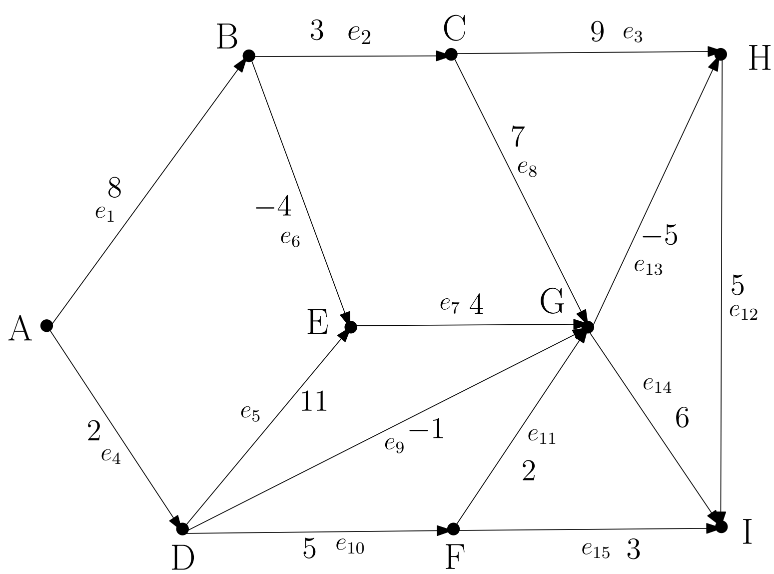
Prof. Dr. Sándor P. Fekete
Dr. Christiane Schmidt

Netzwerkalgorithmen
Übung 4 vom 11.06.2012

Abgabe der Lösungen bis Montag, den 25.06.12, bis 13:00 Uhr.
Bitte die Blätter vorne deutlich mit eigenem Namen und Gruppennummer versehen!

ACHTUNG: auch dieses Übungsblatt besteht aus 4 Seiten!

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

(b) Betrachte den Algorithmus von Floyd und Warshall, Algorithmus 1, Seite 3. 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)

Aufgabe 3 (Minmax-$s - t-$Pfade):
Modifiziere den Algorithmus von Dijkstra, so dass er das Problem löst, einen Minmax-$s - t-$Pfad zu finden: Gegeben sind ein gerichteter Graph G, $c : E(G) \to \mathbb{R}$ und zwei ausgezeichnete Knoten $s, t \in V(G)$. Finde einen $s - t-$Pfad, dessen längste Kante so kurz wie möglich ist.

(15 Punkte)

Aufgabe 4 (Knapsack ohne Wiederholung):
In der großen Übung am 04.06. haben wir uns das Rucksackproblem und eine Formulierung als dynamische Programm für den Fall der unbeschränkten Anzahl der einzelnen Objekte angeguckt.
Betrachte das Rucksackproblem ohne Wiederholung, das heißt, jedes Objekt darf nur genau einmal (maximal einmal) gepackt werden.
Die Kenntnis von $K(w - w_i)$, wie sie für den Fall unbeschränkter Anzahl verwendet wurde, hilft hier nicht, da wir nicht wissen, ob Objekt i schon verwendet wurde, um diesen Wert zu erreichen.

- Welcher Wert sollte stattdessen verwendet werden?
- Wie kann man diesen aus kleineren Teilproblemen ableiten?

Algorithm 1: Floyd, Warshall (1962)

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

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 ≠ j then
            for k = 1 to n do
                if k ≠ j then
                    if l_ik > l_ij + l_jk then
                        l_ik := l_ij + l_jk;
                        p_ik := p_jk
```

- Gib einen Dynamisches Programm an, für das Rucksackproblem ohne Wiederholung.

(15 Punkte)