6. Übung Optimierung B

Aufgabe 1.
Beweisen Sie die folgende Aussage: Der Dijkstra Algorithmus (siehe nächste Seite), durchgeführt am Graphen $D = (V, A)$ mit ausgezeichneter Ecke s, liefert zu jeder Ecke $v \in V$ die Länge $\text{dist}(v)$ eines kürzesten gerichteten $(s - v)-$Weges ($\text{dist}(v) = \infty$ bedeutet, dass kein solcher Weg existiert). Der Digraph $F := (V', A')$ mit

$$V' := \{v \in V \mid \text{dist}(v) < \infty\} \text{ und } A' := \{(\text{pred}(v), v) \mid v \in V' \setminus \{s\}\}$$

hat die folgenden Eigenschaften:
1. $A' \subseteq A$.
2. F ist ein Baum.
3. Für jede Ecke $v \in V' \setminus \{s\}$ gibt es genau einen gerichteten $(s - v)-$Weg in F. Dies ist gleichzeitig ein kürzester $(s - v)-$Weg in D.

Aufgabe 2.
a) Zeigen Sie, dass der Dijkstra Algorithmus auf einem Graphen mit n Ecken höchstens n^2 Operationen benötigt.
b) Wie kann man den Algorithmus vereinfachen, wenn man alle Kanten des Graphen mit 1 bewertet?

Aufgabe 3.
Ein schlichter und bewerteter Graph mit den 8 Ecken $x_1, ..., x_8$ sei durch folgende Bewertungsmatrix gegeben:

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
x_1	0	2	5	4	3	8	2	1
x_2	∞	0	1	6	7	∞	5	3
x_3	3	2	0	5	3	4	2	3
x_4	7	4	5	0	3	6	4	2 .
x_5	6	2	∞	6	0	∞	5	3
x_6	1	3	1	6	3	0	2	∞
x_7	5	2	4	3	3	∞	0	3
x_8	4	3	5	4	3	2	5	0

Dabei bedeutet ein Eintrag a an der Stelle (i, j) für $a < \infty$, dass ein Bogen mit Bewertung a vom Knoten x_i zum Knoten x_j führt. Ist $a = \infty$, so existiert dieser Bogen nicht. Führen Sie den Dijkstra Algorithmus für diesen Graphen durch. Wählen Sie dabei $s := x_1$.

Bitte wenden.

Aufgabe 4.
a) Führen Sie den Algorithmus von Moore, Bellman und Ford (siehe nächste Seite) an dem Graphen mit Bewertungsmatrix

$$\begin{pmatrix} 0 & 2 & 5 & 4 \\ \infty & 0 & -1 & 6 \\ 3 & 2 & 0 & -5 \\ 7 & 7 & 5 & 0 \end{pmatrix}$$

durch und wählen Sie $s := x_3$.
b) Warum ist ein Kürzeste-Wege Problem auf einem Graphen, der einen negativen Kreis enthält, nicht sinnvoll gestellt?

Dijkstra-Algorithmus:

Eingabe: Ein Digraph $D := (V, A)$ mit einer ausgezeichneten Ecke $s \in V$ und einer Längenfunktion $l : A \to \mathbb{R}^+$.
Schritt 1: Setze $\text{dist}(s) := 0$, $\text{dist}(v) = l(a)$ für $a = (s, v) \in A$ und $\text{dist}(v) = \infty$ für $(s, v) \notin A$. Außerdem definiere $T := V \setminus \{s\}$ und $\text{pred}(v) := s$ für $v \in T$.
Schritt 2: Sei $U := \{u \in T \mid \text{dist}(u) \leq \text{dist}(v) \text{ für alle } v \in T\}$, setze dann $T := T \setminus U$.
Schritt 3: Für jedes $u \in U$ und für alle Bögen $a = (u, v) \in A$ mit $v \in T$ überprüfe man, ob $\text{dist}(v) > \text{dist}(u) + l(a)$. Falls ja, so setze man $\text{dist}(v) := \text{dist}(u) + l(a)$ und $\text{pred}(v) := u$.
Schritt 4: Falls $T = \emptyset$: STOP. Anderenfalls gehe man zu Schritt 2.

Algorithmus von Moore Bellman und Ford:

Eingabe: Ein Digraph $D := (V, A)$ mit einer ausgezeichneten Ecke $s \in V$ und einer Längenfunktion $l : A \to \mathbb{R}$. D darf dabei keine negativen Kreise enthalten.
Schritt 1: Setze $\text{dist}(s) := 0$ und $\text{dist}(v) = \infty$ für alle $v \in V \setminus \{s\}$.
Schritt 2: Für alle Bögen $a = (v, w) \in A$ überprüfe man, ob $\text{dist}(w) > \text{dist}(v) + l(a)$. Falls ja, so setze man $\text{dist}(w) := \text{dist}(v) + l(a)$ und $\text{pred}(w) := v$.
Schritt 3: Schritt 2 wiederhole man $(n-1)$–mal.