Übungsblatt 9. Abgabe am **Mittwoch, 13.1.2010 (vor der Vorlesung)**.

Aufgabe 1. (Ein Codebuch)

Die Nachricht

`babbbaabba`

soll mit dem Codebuch

Text	Code	Länge
`a`	`00`	2
`b`	`010`	3
`ba`	`0110`	4
`bb`	`0111`	4
`abb`	`1`	1

codiert werden. Ein Beispiel für eine mögliche Codierung mit Gesamtlänge 20 ist:

ba	bb	ba	a	bb	a
0110	0111	0110	00	0111	00

Finden Sie eine Codierung mit minimaler Gesamtlänge. Erläutern Sie hierzu kurz, wie Sie dieses Problem mit Hilfe der Graphentheorie lösen können und berechnen Sie eine Lösung.

Hinweis: Entwerfen Sie einen geeigneten gewichteten Graphen und formulieren Sie die Aufgabe als kürzeste–Wege–Problem.

(10 Punkte)

Aufgabe 2. (Zusammenhang)

Zeigen Sie: Ein ungerichteter Graph $G = (V, E)$ mit $|E| > \binom{|V|-1}{2}$ ist zusammenhängend.

(10 Punkte)

Aufgabe 3. (Mittlere Baumhöhe)

Die Höhe eines Knotens in einem *out-tree* ist erklärt als der Abstand zur Wurzel. Die Wurzel hat Höhe 0. Die Höhe eines *out-tree* ist die Höhe des Knotens mit maximaler Höhe im *out-tree*. Siehe auch Aufgabe 1 von Übungsblatt 7.

Sei nun ein nicht leerer *out-tree* $\mathcal{T}$ dargestellt durch Zeiger von jedem Knoten zu dessen Söhnen. Die Anzahl der Knoten n sei *nicht* bekannt. Entwerfen Sie einen Algorithmus, der

a) die Höhe $H(\mathcal{T})$ von $\mathcal{T}$

b) die mittlere Höhe $\overline{H}(\mathcal{T})$ von $\mathcal{T}$ (entspricht der mittleren Pfadlänge in Aufgabe 1 von Übungsblatt 7)

in Zeit $O(n)$ berechnet und jeden Knoten von $\mathcal{T}$ höchstens einmal besucht.

Eingabe: Ein Zeiger auf den Wurzelknoten von $\mathcal{T}$.

Ausgabe: Die Anzahl der Knoten n sowie die berechneten Werte $H(\mathcal{T})$ und $\overline{H}(\mathcal{T})$.

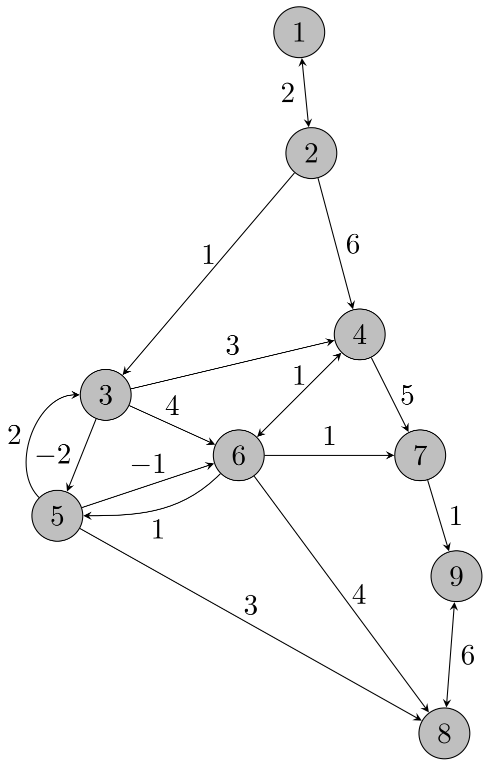
(10 Punkte)

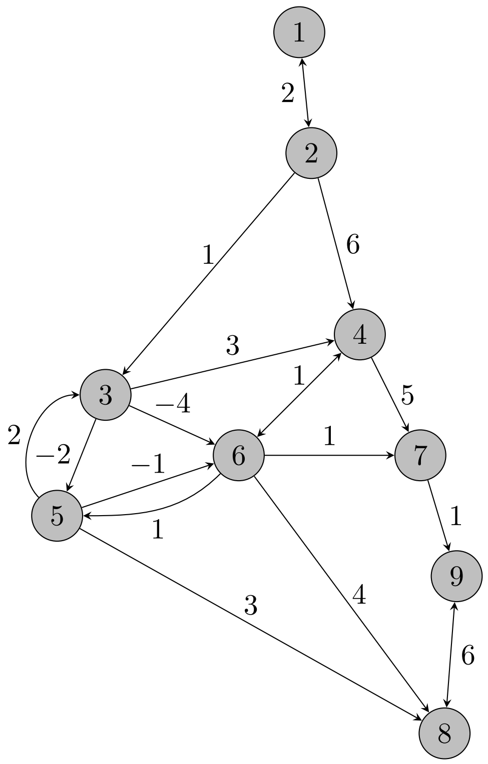
Programmieraufgabe 4. (Moore-Bellman-Ford-Algorithmus)

Implementieren Sie den Moore-Bellman-Ford-Algorithmus. Die Version aus der Vorlesung ist nur für konservative Gewichte geeignet. Erweitern Sie Ihre Implementation so, dass im Falle von *nicht* konservativen Gewichten das Programm eine entsprechende Meldung ausgibt. Folgende Anforderungen sollte Ihr Programm erfüllen:

1. `Input:` Ein gerichteter Graph $G = (V, E)$, $c : E \to \mathbb{R}$, $s \in V$
2. `Output:` Für konservative Gewichte: $\ell(x) = \mathrm{dist}_{(G,c)}(s, x)$ für alle $x \in V$.
   Andernfalls: Meldung über negativen Zyklus.
3. Die Laufzeitanforderungen dürfen nicht größer als $O(|V| \cdot |E|)$ sein.

Berechnen Sie durch Anwendung Ihrer Implementation des Moore-Bellmann-Ford-Algorithmus von Knoten 1 aus kürzeste Wege zu allen anderen Knoten für die beiden folgenden Digraphen:





Geben Sie dazu als Ausgabe eine Tabelle an, in der zu jedem Knoten sein Vorgänger sowie die Distanz von 1 aus verzeichnet ist. Der Vorgänger zu Knoten 2 ist beispielsweise 1 mit Distanz 2.

(10 Punkte)

Achtung: Bezüglich des konkreten Termins der Abgabe der Programmieraufgabe hängen in der Woche vom 4.1.-8.1.2010 im CIP-Pool Listen aus. Man beachte die eingeschränkten Öffnungszeiten: 4.1.-6.1.2010 von 13-16 Uhr; 7.1.-8.1.2010 von 9-18 Uhr.

Abgabe innerhalb der Woche 11.1.–15.1.2010 im CIP-Pool

Wir wünschen eine schöne Weihnachtszeit und ein frohes neues Jahr!