Algorithmen und Datenstrukturen II
SS 2009 – Serie 3

1. (9 Punkte) **Graphen: Kürzeste Wege**
   Gegeben sei folgender gerichteter, gewichteter Graph $G = (V, E)$:

$$V = \{\mathrm{A, B, C, D, E, F, G, H, I, Q}\}\ ,$$
$$E = \{(\mathrm{A, B}, 1), (\mathrm{A, C}, 5), (\mathrm{B, G}, 7), (\mathrm{B, I}, 1), (\mathrm{C, G}, 1.2), (\mathrm{C, H}, 1.5), (\mathrm{D, H}, 3), (\mathrm{E, F}, 3), (\mathrm{G, Q}, 1.8), (\mathrm{I, D}, 1), (\mathrm{I, E}, 3), (\mathrm{Q, I}, -8)\}$$

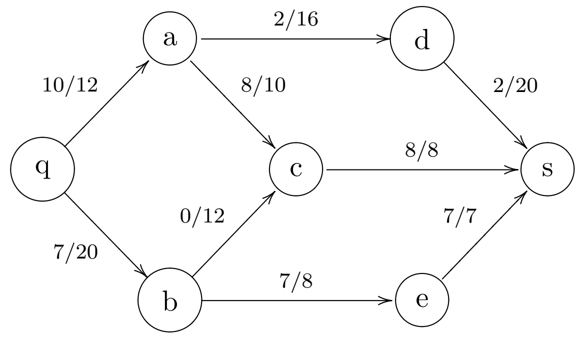
   Dezimalbrüche werden mit dem Dezimalpunkt dargestellt.

   a) (5 Punkte) Verwenden Sie den Dijkstra-Algorithmus, um die kürzesten Wege vom Startknoten A zu allen anderen Knoten in G zu berechnen.
      - (2 Punkte) Geben Sie als Zwischenergebnis jeweils die Werte im Distanzfeld D und die Belegung der Prioritätswarteschlange Q vor jedem Durchlauf der äußeren Schleife an. Falls zwei Knoten mit den gleichen Kosten verbunden sind, soll der alphabetisch Erstere bevorzugt werden (also z. B. K vor L).
      - (1 Punkt) Geben Sie an, an welcher Stelle zu erkennen ist, daß dieser Algorithmus die negative Kante nicht angemessen berücksichtigt.
      - (1 Punkt) Geben Sie das Distanzfeld D nach Beendigung des Algorithmus an.
      - (1 Punkt) Falls lediglich die kürzeste Entfernung vom Knoten A zum Knoten I zu bestimmen gewesen wäre: An welcher Stelle hätte der Algorithmus üblicherweise abgebrochen werden können und wie groß wäre zu diesem Zeitpunkt die Entfernung gewesen?

   b) (4 Punkte) Berechnen Sie nun den kürzesten Weg vom Knoten A zum Knoten F mit dem Bellman-Ford-Algorithmus.
      - (2 Punkte) Geben Sie als Zwischenergebnis jeweils die Werte im Distanzfeld D vor jedem Durchlauf der äußeren Schleife an.
      - (1 Punkt) Geben Sie das Distanzfeld D nach Beendigung des Algorithmus an.
      - (1 Punkt) Hätte der Algorithmus eher abgebrochen werden können? Begründung!

      Die Reihenfolge der Kanten während der Kantenwahl in der inneren Schleife sei **nicht beliebig**, sondern durch obige Auflistung der Kanten innerhalb der Kantenmenge E gegeben. Dies dient der Eindeutigkeit der Zwischenlösungen.

2. (9 Punkte) **Flüsse und Netzwerke**
   Gegeben sei folgendes Flußnetzwerk, q sei die Quelle, s die Senke:



   Dabei bedeutet die Notation $i \xrightarrow{f/k} j$, daß f der aktuelle Fluß von i nach j ist und k die Kapazität dieser Verbindung.

   a) (2 Punkte) Prüfen Sie, ob der oben angegebene Fluß zulässig ist. Geben Sie die (nicht)erfüllten notwendigen Eigenschaften an.
   b) (2 Punkte) Geben Sie den Restgraphen zu dem gegebenen Fluß im Netzwerk an.
   c) (4 Punkte) Optimieren Sie den Fluß in diesem Netzwerk mittels des Ford-Fulkerson-Algorithmus. Stellen Sie das resultierende Netzwerk graphisch dar. Geben Sie als Zwischenschritte die Wege, entlang derer der Fluß erhöht wird, und die Erhöhung mit an. Um wieviel Prozent kann der Fluß erhöht werden?
   d) (1 Punkt) Wie groß ist der maximale Fluß, wenn die Kanten $(\mathrm{c}, \mathrm{a}, 2)$ und $(\mathrm{c}, \mathrm{d}, 2)$ eingefügt werden?

3. (5 Punkte) **Lauflängenkodierung**
   Gegeben sei folgende Zeichenkette:

`OOOOOOOOSE12HOSE#HOSEAAAAAAAAASE#######A####BB#DUDEUUUUEEEE#UUUU#EEEE#DU#DE6`

   a) (3 Punkte) Komprimieren Sie diese Zeichenfolge mittels Lauflängenkomprimierung. Verwenden Sie `#` als Escape-Zeichen und Buchstaben zur Kodierung der Länge der Läufe (vgl. Skript 6, Folien 5 u. 6). Geben Sie die resultierende Zeichenfolge sowie den Kompressionsfaktor an.
   b) (2 Punkte) Die eben ermittelte Zeichenfolge werde nun als unkomprimierter Originaltext behandelt. Komprimieren Sie diese Zeichenfolge einmal unter Verwendung von `#` und einmal von `@` als Escape-Zeichen. Geben Sie jeweils die resultierende Zeichenfolge sowie den entsprechenden Kompressionsfaktor an.