Datenstrukturen und Algorithmen, SS 2024
Prof. Dr. P. Rossmanith
D. Cloerkes, M. Gehnen, D. Mock, N. Speetzen

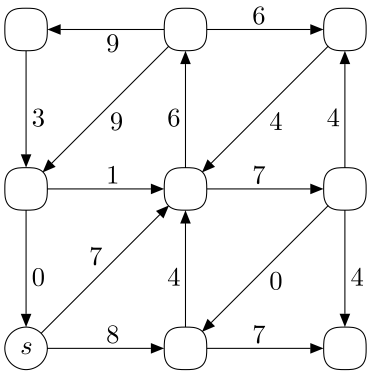
Abgabe: 14.06.2024

Übungsblatt 08

Aufgabe T27

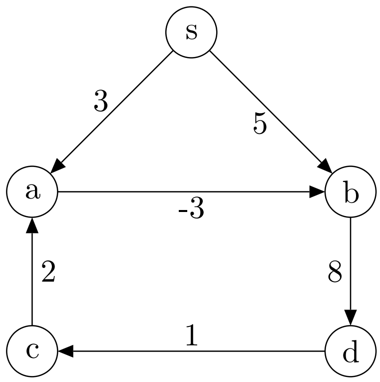
Finden Sie die kürzesten Wege zu allen Knoten vom Startknoten s, indem Sie den Algorithmus von Dijkstra verwenden.

Geben Sie dazu die resultierenden Distanzen von s zu allen Knoten an, indem Sie diese in die Knoten des Graphen eintragen. Markieren Sie außerdem jede Kante, die zu einem kürzesten Weg von s aus gehört.



Aufgabe T28

Wenden Sie den Algorithmus von Bellman und Ford auf folgendes Netzwerk an. Verwenden Sie s als Startknoten.

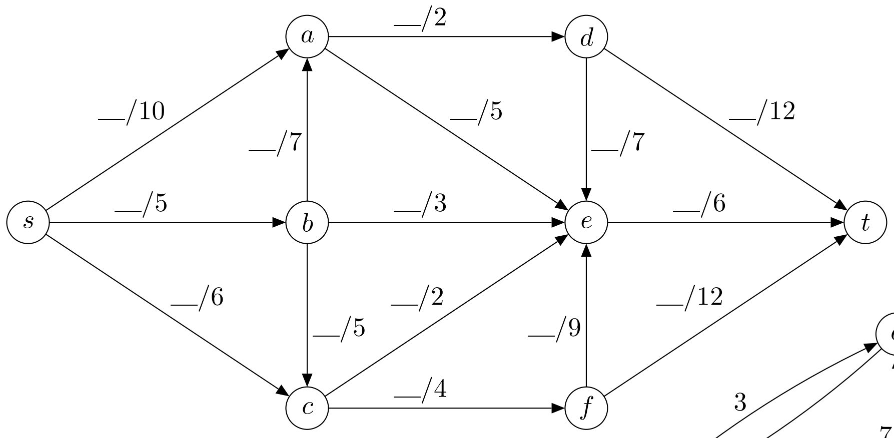


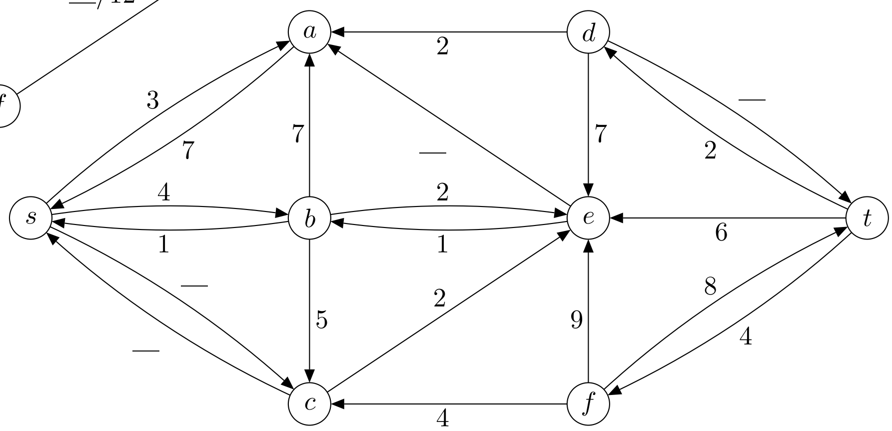
Aufgabe T29

Beweisen Sie den zweiten Punkt von Lemma A: $f(X, Y) = -f(Y, X)$ für $X, Y \subseteq V$, falls f ein Fluss für $G = (V, E)$ ist.

Aufgabe T30

a) Sie finden links ein Flussnetzwerk G. Rechts ist das dazugehörige Residualnetzwerk G_f bezüglich eines Flusses f. Zeichnen Sie den Fluss f in die linke Zeichnung auf den Strichen ein und rechts die fehlenden Kapazitäten in G_f.





b) Berechnen sie den Wert des Flusses f.

c) Ist f maximal? Begründen Sie Ihre Antwort kurz.

Aufgabe H26 (12 Punkte)

Der Algorithmus von Dijkstra kann auf gerichteten Graphen mit negativen gewichteten Kanten nicht verwendet werden. Das ist schade, da der Algorithmus eine deutlich geringere Zeitkomplexität als der allgemeinere Bellman-Ford Algorithmus hat.

Überlegen Sie sich, wie es möglich ist, mit gleicher Zeitkomplexität wie der des Algorithmus von Dijkstra kürzeste Pfade zwischen zwei Knoten s und t zu berechnen, sollte der gerichtete Graph genau eine negativ gewichtete Kante enthalten.

Aufgabe H27 (18 Punkte)

Der Alien Glorbo ist bei seinen Weltraumreisen auf ein Problem gestoßen: Er hat sich zu viel Wackelpudding gekauft. Um diesen nicht zurücklassen zu müssen, scheint die einzige Möglichkeit zu sein, einen seiner Treibstofftanks vollständig zu entleeren, um den Wackelpudding im Tank transportieren zu können.

Die Statusanzeige zeigt Glorbo die momentanen Füllmengen $x_i \in \mathbb{N}^+$ seiner drei (für unsere Zwecke unbegrenzt großen) Tanks. Glorbos Ziel ist, den Treibstoff so umzufüllen, dass ein Tank leer ist. Leider hat er hierbei ein paar Einschränkungen zu beachten:

- Er kann in einem Schritt immer nur ganzzahlige Litermengen auf einmal umfüllen, nie fraktionale Anteile.
- Er kann in einem Schritt immer nur von einem Tank i nach j umfüllen, wenn $x_i \geq x_j$.
- Damit der Druck im Zieltank nicht zu schnell ansteigt, wird immer genau soviel umgefüllt, dass sich die Füllmenge im Zieltank pro Schritt verdoppelt. Somit ist die Menge, die in einem Schritt umgefüllt wird immer $\min(x_i, x_j) = x_j$.

Da jeder Umfüllschritt unabhängig von der umgefüllten Menge sehr lange dauert, möchte Glorbo natürlich möglichst wenig Schritte durchführen.

Um Ihnen die Problemdefinition klarer zu machen, hat Glorbo sich bereits ein Beispiel überlegt: $x_0 = 5$ l, $x_1 = 17$ l und $x_2 = 42$ l. In diesem Beispiel sind die einzigen möglichen Umfüllschritte zu Beginn:

- $1 \to 0$: 10 l, 12 l, 42 l
- $2 \to 0$: 10 l, 17 l, 37 l
- $2 \to 1$: 5 l, 34 l, 25 l

Eine optimale Lösung hat hier z.B. vier Schritte: $1 \to 0, 2 \to 0, 2 \to 0, 1 \to 2$.

Nun liest Glorbo seine tatsächlichen Füllstände ab, welche zufälligerweise zu Teilen mit der kleinsten Matrikelnummer Ihrer Gruppe übereinstimmen. Von dieser Matrikelnummer bilden die ersten drei Ziffern x_0 und die letzten drei x_1 (z.B. "345678"$\to$ $x_0 = 345$ l, $x_1 = 678$ l). Unabhängig davon ist $x_2 = 1234$ l. Helfen Sie Glorbo!

Sein Bordcomputer beherrscht die Programmiersprachen Python, Java, C++ und Rust. Stellen Sie sicher, dass Sie Glorbo sowohl die optimalen Umfüllschritte, als auch ihren Programmcode zukommen lassen! (Bitte den Code ebenfalls in moodle hochladen.) Das Programm soll natürlich auf beliebigen Eingaben funktionieren, damit Glorbo es ggfs. wiederverwenden kann. Beschreiben Sie auch in Worten, wie Ihr Programm funktioniert.

Aufgabe H28 (10 Punkte)

Wenden Sie die Ford–Fulkerson-Methode auf das folgende Flussnetzwerk an. Zeichnen Sie das resultierende Residualnetzwerk nach der ersten und letzten Augmentierung.