



Übungen zu Woche 2: Network Flow I

🧑‍🎓 **2.1 Max-Flow-Min-Cut I.** In [Abbildungen 1](#) and [2](#) sind zwei Flussnetzwerke G_1 und G_2 dargestellt. Die Kapazität $c(e)$ von Kante e steht dabei neben der jeweiligen Kante.

- Gib irgendeinen s - t -Schnitt in G_1 an, der *nicht* minimal ist. *Warum* ist dein Schnitt nicht minimal?
- Zähle *alle* minimalen s - t -Schnitte in G_1 auf. Sicher, dass du alle erwischt hast?
- Was ist die minimale Kapazität eines s - t -Schnittes in G_2 ? Warum?

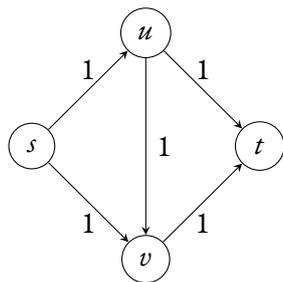


Abbildung 1: Das Flussnetzwerk G_1

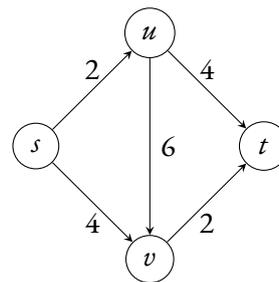


Abbildung 2: Das Flussnetzwerk G_2

🧑‍🎓 **2.2 Max-Flow-Min-Cut II.** [Abbildung 3](#) zeigt ein Flussnetzwerk G_3 sowie einen darauf definierten s - t -Fluss. Der Ausdruck f/c auf jeder Kante gibt an, wie viel Fluss $f \in \mathbb{N}$ über die Kante geschickt wird und welche Kapazität $c \in \mathbb{N}$ die Kante hat.

- Was ist der Wert des Flusses? Ist dieser Fluss ein maximaler s - t -Fluss in diesem Graphen? Warum?
- Finde einen minimalen s - t -Schnitt und gib dessen Kapazität an.

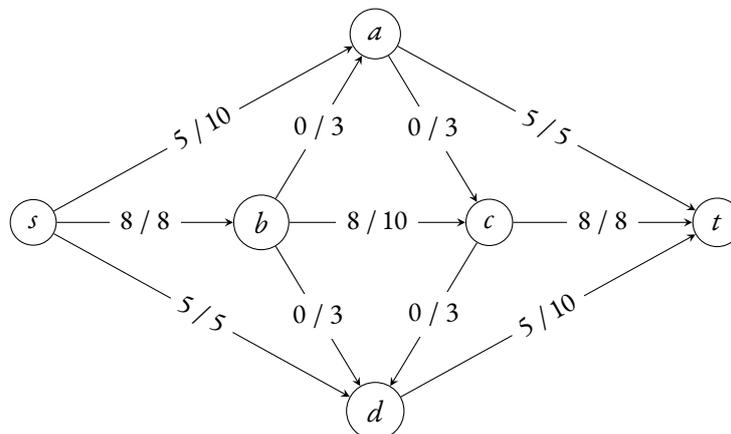


Abbildung 3: Das Flussnetzwerk G_3

🧑‍🚒 **2.3 Ford-Fulkerson Algorithmus.** *Abbildung 4* zeigt ein Flussnetzwerk G_4 . Führe den Ford-Fulkerson Algorithmus Schritt für Schritt aus, und berechne dadurch einen maximalen s - t -Fluss und einen minimalen s - t -Schnitt auf G_4 .

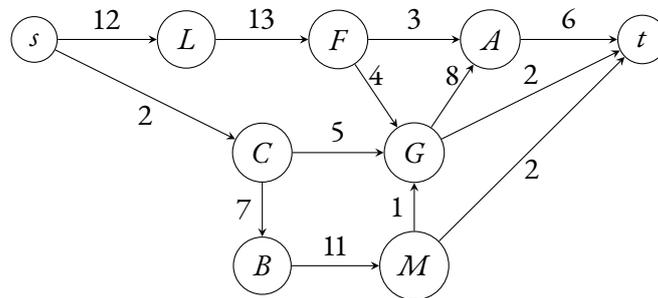


Abbildung 4: Das Flussnetzwerk G_4

🧑‍🚒 **2.4 Eigenschaften maximaler Flüsse.** Entscheide, ob die folgende Aussage wahr oder falsch ist. Wenn sie wahr ist, erkläre, warum sie das ist. Wenn sie falsch ist, gebe ein Gegenbeispiel an.

Sei G ein beliebiges Flussnetzwerk mit einer Quelle s , einer Senke t und einer positiven, ganzzahligen Kapazität c_e auf jeder Kante e . Wenn f ein maximaler s - t -Fluss in G ist, dann saturiert f jede von s ausgehende Kante mit einem Fluss (d.h., für alle Kanten e , die von s ausgehen, gilt $f(e) = c_e$).

(Tipp: Prüfe die Aussage zunächst für kleine Beispiele wie etwa G_1 , G_2 , G_3 und G_4 .)

🧑‍🚒 **2.5 Eigenschaften minimaler Schnitte.** Entscheide, ob die folgende Aussage wahr oder falsch ist. Wenn sie wahr ist, erkläre, warum sie das ist. Wenn sie falsch ist, gebe ein Gegenbeispiel an.

Sei G ein beliebiges Flussnetzwerk mit einer Quelle s , einer Senke t und einer positiven, ganzzahligen Kapazität c_e auf jeder Kante e . Sei (A, B) ein minimaler s - t -Schnitt bezüglich der Kapazitäten $\{c_e : e \in E\}$. Wenn wir jede Kapazität um 1 erhöhen, dann ist (A, B) immer noch ein minimaler s - t -Schnitt bezüglich der neuen Kapazitäten $\{1 + c_e : e \in E\}$.

🧑‍🚒 **2.6 Zombie-Ausbruch.** Im Land 1D hat ein Zombie-Ausbruch stattgefunden. Der Premierminister des Landes bittet dich, eine Lösung zu finden, um diesen Ausbruch zu stoppen. Du erhältst eine Karte des ganzen Landes, auf der die Koordinaten aller Städte verzeichnet sind. Es gibt X infizierte und Y noch nicht infizierte Städte. Die Karte enthält ebenso Informationen über Straßen, die die Städte miteinander verbinden. Alle Straßen sind Einbahnstraßen, das heißt sie führen in genau eine Richtung. Jede Straße führt direkt von einer Stadt in eine andere. Es gibt insgesamt R Straßen. Sowohl Zombies als auch Menschen haben die Straßenverkehrsordnung so sehr verinnerlicht, dass sie sich nur auf Straßen bewegen können und auch nur in der vorgesehenen Richtung.

Eine Stadt a ist von einer Stadt b aus erreichbar, wenn man von b aus einer oder mehreren Straßen folgen kann, um nach a zu gelangen. Du kannst annehmen, dass die Reise von einer Stadt in eine andere erreichbare Stadt nicht länger dauert als einen Tag.

- a) **Rette die Hauptstadt!** Die Hauptstadt ist noch nicht infiziert und der Premierminister möchte wissen, wie viele Straßen zerstört werden müssen, damit die Hauptstadt verschont bleibt. Gebe einen Algorithmus an, der die minimale Anzahl an Straßen ausgibt, die zerstört werden müssen, um die Hauptstadt von den infizierten Städten abzugrenzen. Analysiere die Laufzeit deines Algorithmus in Abhängigkeit von X , Y und R . Zeige, dass dein Algorithmus funktioniert.
- b) **Verteile den Impfstoff!** Wissenschaftler aus der Hauptstadt haben einen wirksamen Impfstoff gefunden. Um diesen in den noch nicht infizierten Städten zu verteilen, müssen in jede noch nicht infizierte Stadt 10 Ärzte geschickt werden. Die Ärzte können nur auf den Straßen reisen. Sie können sogar durch infizierte Städte reisen, allerdings können das nur maximal 50 Ärzte pro Stadt und Tag tun. Gebe einen Algorithmus an, der entscheidet, ob man den Impfstoff innerhalb eines Tages zu allen noch nicht infizierten Städten transportieren kann. Analysiere die Laufzeit des Algorithmus in Abhängigkeit von X , Y und R . Zeige, dass dein Algorithmus funktioniert.



2.7 Eine andere Sicht auf Schnitte. Schnitte werden manchmal als Teilmengen der Kanten eines Graphen definiert anstatt als Partitionen der Knotenmenge. In dieser Aufgabe soll gezeigt werden, dass beide Definitionen *nabezu* äquivalent sind.

Sei G ein gerichteter Graph mit zwei ausgezeichneten Knoten s und t . Wir sagen, dass eine Menge X von Kanten die Knoten s und t *separiert*, falls jeder gerichtete s - t -Pfad mindestens eine Kante aus X enthält. Für eine Menge S von Knoten definieren wir

$$\partial S := \{u \rightarrow v : u \in S, v \notin S\}.$$

Somit ist ∂S die Menge der gerichteten Kanten, die aus S raus führen.

- a) Sei (S, T) ein beliebiger s - t -Schnitt in G . Beweise, dass die Kantenmenge ∂S die beiden Knoten s und t separiert.
- b) Sei X eine beliebige Teilmenge von Kanten, sodass X die Knoten s und t separiert. Beweise, dass es einen s - t -Schnitt (S, T) gibt, für den $\partial S \subseteq X$ gilt.
- c) Sei X eine minimale Teilmenge von Kanten, sodass X die Knoten s und t separiert. Beweise, dass es einen s - t -Schnitt (S, T) gibt, für den $\partial S = X$ gilt.



2.8 Puzzle der Woche: Vier Münzen. Du musst ein Spiel gegen den Henker gewinnen. Bevor das Spiel startet, werden deine Augen verbunden. Es werden vier Münzen auf einem quadratischen Tisch platziert, sodass in jeder Ecke eine Münze liegt. Ob und welche der Münzen mit der Zahl-Seite nach oben liegen wird durch den Henker bestimmt: beliebig und für dich unbekannt. Dein Ziel ist es, dass alle vier Münzen mit der Kopf-Seite nach oben gerichtet auf dem Tisch liegen.

In jeder Runde kannst du irgendeine Teilmenge von Münzen auswählen, die dann vom Henker gleichzeitig umgedreht werden. Ist danach auf allen vier Münzen der Kopf zu sehen, hast du gewonnen. Andernfalls darf der Henker den Tisch um 90, 180, 270 oder 360 Grad drehen; da deine Augen verbunden sind, weißt du nicht, welche Drehung gewählt wurde. Wenn du es nicht schaffst, in höchstens 20 Runden zu gewinnen, dann verlierst du und der Henker geht seiner eigentlichen Berufung nach. Was ist deine Strategie?

Projektideen. Such dir im Tutorium wieder eine Gruppe und ein *Projekt* aus, bei dem für dich das Feedback deines Tutors nützlich wäre. Das kann die schriftliche Ausarbeitung einer oder mehrerer Übungsaufgaben sein (siehe erstes Übungsblatt für die formalen Anforderungen). Oder auch diese Aufgabe:



2.9 Implementierung von Ford-Fulkerson. Implementiere ein Programm, das als Eingabe ein Flussnetzwerk erhält und welches mithilfe des Ford-Fulkerson Algorithmus einen maximalen Fluss zwischen zwei Knoten des Netzwerkes berechnet. Teste dein Programm in Coderunner (siehe Moodle).