



Übungen zu Woche 3: Network Flow II

- 3.1 **Residualgraphen zeichnen.** Betrachte den Fluss f und das Flussnetzwerk (G, s, t, c) , das in Abbildung 1 dargestellt ist. Zeichne den Residualgraphen G_f und den eingeschränkten Residualgraphen $G_f(3)$.

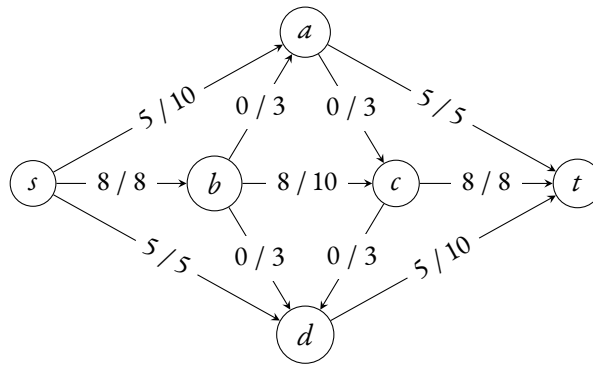
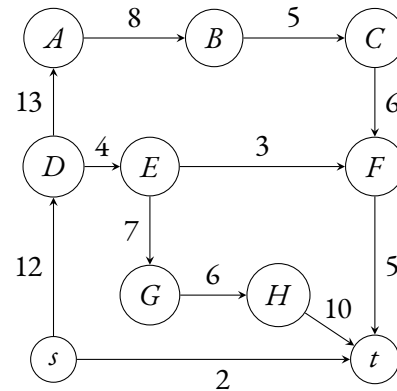
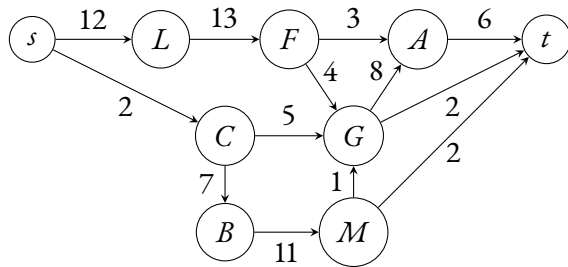


Abbildung 1

- 3.2 **Edmonds-Karp Algorithmus und capacity-scaling algorithm.** Benutze sowohl den Edmonds-Karp Algorithmus, als auch den *capacity-scaling algorithm*, um in den folgenden beiden Graphen einen maximalen s - t -Fluss und einen minimalen s - t -Schnitt zu berechnen. Schreibe für jeden augmentierenden Pfad die zugehörigen Knoten und den Wert, um die der Pfad augmentiert wird, auf.





3.3 Blutspende. Statistisch gesehen erhöht sich die Anzahl an Unfällen mit Beginn des Frühlings und damit auch der Bedarf an medizinischen Notfallbehandlungen, sodass auch vermehrt Bluttransfusionen notwendig sind. Stell dir vor, du arbeitest für ein Krankenhaus, das beurteilen muss, ob seine Blutreserven ausreichend sind.

Beim Blutspenden gibt es eine grundlegende Regel: Eine Person hat in ihrem Blut bestimmte Antigene (du kannst dir Antigene als eine Art molekulare Signatur vorstellen). Bei einer Blutspende kann eine Person nur Blut empfangen, welches die gleichen Antigene hat, wie das eigene. Konkret gesagt unterteilt dieses Prinzip das Blut in vier Gruppen: A, B, AB und 0. Blutgruppe A hat das Antigen A, Blutgruppe B hat das Antigen B, Blutgruppe AB hat beide und Blutgruppe 0 hat keines der Antigene. Demnach können Patienten mit Blutgruppe A nur Blut der Gruppen A und 0, Patienten der Blutgruppe B nur Blut der Gruppen B und 0, Patienten der Blutgruppe 0 nur Blut der Gruppe 0 und Patienten der Blutgruppe AB Blut aller Blutgruppen empfangen.

- a) Das Krankenhaus hat für jede der vier Blutgruppen eine bestimmte Anzahl an ganzen Einheiten vorrätig. Diese Anzahlen sind gegeben durch $s_0, s_A, s_B, s_{AB} \in \mathbb{N}$. Außerdem kennt das Krankenhaus den prognostizierten Bedarf $d_0, d_A, d_B, d_{AB} \in \mathbb{N}$ an Blut für die nächste Woche. Entwirf einen effizienten Algorithmus, der entscheidet, ob das vorrätige Blut den prognostizierten Bedarf der nächsten Woche deckt. (*Tipp: verwende ein Flussnetzwerk als Entwurf für ein geeignetes Flussnetzwerk zur Entscheidungsfindung*)
- b) Wir schauen uns nun ein konkretes Beispiel an. Die folgende Tabelle gibt den Vorrat und die Nachfrage an Blut an:

Blutgruppe	Vorrat	Nachfrage
0	50	45
A	36	42
B	11	10
AB	8	3

Sind die 105 vorrätigen Einheiten Blut ausreichend, um der Nachfrage von 100 Einheiten gerecht zu werden? Finde eine Zuordnung, sodass die maximale Anzahl an Patienten behandelt werden kann. Benutze die Idee minimaler Schnitte in einem passenden Flussnetzwerk, um zu begründen, warum nicht alle Patienten behandelt werden können. Stelle außerdem eine allgemeinverständliche Erklärung dieses Umstandes für diejenigen Mitglieder der Krankenhausverwaltung bereit, die ALGO2 nicht belegt haben. (Beispielsweise soll deine Erklärung die Worte *Fluss*, *Schnitt* oder *Graph* nicht in dem Sinne enthalten, wie wir sie in der Vorlesung benutzen.)

3.4 Weihnachtsbäume. Professor Regloh hat dich beauftragt, die jährliche Weihnachtsfeier für alle Student:innen der Goethe-Universität zu organisieren. Du musst einen Plan erstellen, auf dem die Platzierung der Tische im Casino-Festsaal vermerkt ist. Damit die Brandschutzmaßnahmen eingehalten werden können, hat die Frankfurter Feuerwehr den Festsaal in ein rechteckiges $n \times m$ Gitter aufgeteilt und festgelegt, dass höchstens zwei Tische in jeder Zeile und höchstens ein Tisch in jeder Spalte aufgestellt werden dürfen. Leider liebt Professor Regloh Weihnachtsbäume sehr und hat schon einige aufstellen lassen. Du darfst diese nicht verschieben und kannst einen Tisch nur dort aufstellen, wo noch kein Weihnachtsbaum steht.

★							
	★	★		★	★		★
	★	★		★	★		★
	★		★			★	

Abbildung 2

★	<i>T</i>						<i>T</i>
<i>T</i>	★	★	<i>T</i>	★	★		★
	★	★		★	★	<i>T</i>	★
	★	<i>T</i>	★		<i>T</i>	★	

Abbildung 3

Beispiel. Es gilt $n = 4$ und $m = 8$. Die ★ stellen Weihnachtsbäume dar und *T* die Tische. In [Abbildung 2](#) ist die Anfangssituation (Weihnachtsbäume positioniert, Tische nicht) dargestellt. In [Abbildung 3](#) ist eine bestmögliche Positionierung der maximalen Anzahl an Tischen (7) dargestellt.

- 🧠 Modelliere das Problem als ein Netzwerkflussproblem. Erkläre, wie du dir das überlegt hast und zeichne den zum oben angegebenen Beispiel passenden Graphen.
- 🧠🔑 Gegeben sind n, m und die Platzierung der Weihnachtsbäume. Beschreibe einen Algorithmus, der eine maximale Anzahl an Tischen, die im Festsaal platziert werden können, ausgibt. Analysiere die asymptotische Laufzeit deines Algorithmus und denke daran, die Korrektheit deines Algorithmus zu zeigen.
- 🔑 Implementiere deinen Algorithmus in Python, Java, oder C/C++. (In Moodle steht hierfür eine Coderunner-Aufgabe zur Verfügung!)

🧠🔑 **3.5 Fluchtwege.** Wir definieren das Fluchtproblem wie folgt: Gegeben ist ein gerichteter Graph $G = (V, E)$ (stell dir ein Straßennetzwerk vor). Eine Teilmenge $X \subseteq V$ aller Knoten sind bewohnte Knoten und eine andere Teilmenge $S \subseteq V$ sind sichere Knoten. Du kannst annehmen, dass X und S disjunkt sind. Für den Notfall soll es Fluchtwege von den bewohnten Knoten zu den sicheren Knoten geben. Eine Menge von Fluchtwegen ist als eine Menge von Pfaden in G definiert, sodass

- jeder Knoten aus X der Anfang eines Pfades ist,
- der letzte Knoten jedes Pfades in S liegt und
- die Pfade paarweise keine gemeinsamen Kanten besitzen.

Diese Menge von Pfaden sorgt dafür, dass die bewohnten Knoten sicher evakuiert werden können, ohne dass eine Kante in G überlastet wird.

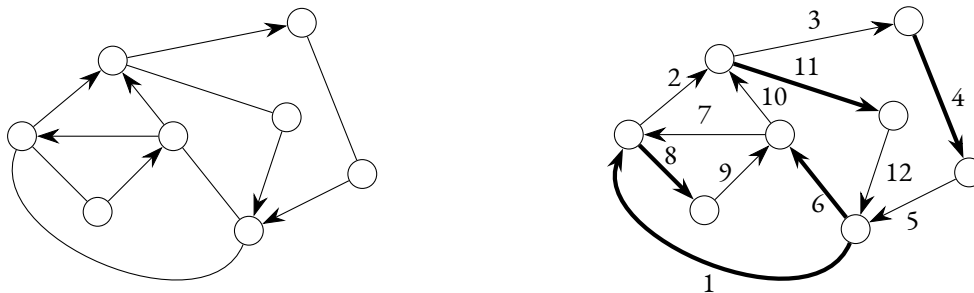
- Zeige, wie bei einer Eingabe von G, X, S in polynomieller Zeit entschieden werden kann, ob eine solche Menge von Fluchtwegen existiert.
- Wir wollen das gleiche Problem wie in Aufgabenteil a) mit einer strengeren Version von Bedingung iii. lösen, das heißt die Pfade dürfen keine gemeinsamen Knoten mehr haben. Zeige, wie in polynomieller Zeit entschieden werden kann, ob eine Menge von Fluchtwegen gemäß dieser neuen Bedingungen existiert.
Gib außerdem eine Beispiel-Eingabe G_0, X_0, S_0 an, für die der Algorithmus aus Aufgabenteil a) „Ja“ ausgibt und der Algorithmus aus Aufgabenteil b) „Nein“.



3.6 Eulerkreise in gemischten Graphen. Ein Eulerkreis ist ein Kreis, der jede Kante genau einmal enthält. Es ist bekannt, dass ein stark zusammenhängender Graph einen Eulerkreis genau dann besitzt, wenn jeder Knoten die gleiche Anzahl eingehender und ausgehender Kanten hat.

Gemischte Graphen sind Graphen, in denen einige Kanten gerichtet und einige ungerichtet sind. Wir nehmen an, dass die Graphen schwach zusammenhängend sind, also dass der zugrunde liegende ungerichtete Graph zusammenhängend ist.

Gib einen Algorithmus an, der entscheidet, ob man allen ungerichteten Kanten eine Richtung zuweisen kann, sodass der dadurch entstehende gerichtete Graph einen Eulerkreis besitzt. Im folgenden Beispiel können die ungerichteten Kanten des gemischten Graphen (*links*) so gerichtet werden (*rechts*), dass ein Eulerkreis entsteht (die Zahlen an den Kanten geben die Besuchsreihenfolge an).



Hinweis: Konstruiere einen bipartiten Graphen mit den Knoten aus G auf der einen Seite und den Kanten aus G auf der anderen Seite, und nutze dann den Algorithmus der vorherigen Aufgabe.



3.7 Puzzle der Woche: Das Zwölf-Münzen Problem. Vor dir liegen 12 Münzen. 11 davon sind identisch, aber eine Münze ist schwerer oder leichter als der Rest. Du hast eine traditionelle Balkenwaage mit zwei Waagschalen. Um die Waage zu benutzen, legst du eine Münze in jede Schale. Die Waage zeigt dann an, welche der beiden Seiten (und damit welche der beiden Münzen) schwerer ist. Die Balkenwaage darf dreimal benutzt werden, um die andere Münze zu finden und um festzustellen, ob diese leichter oder schwerer ist als der Rest.