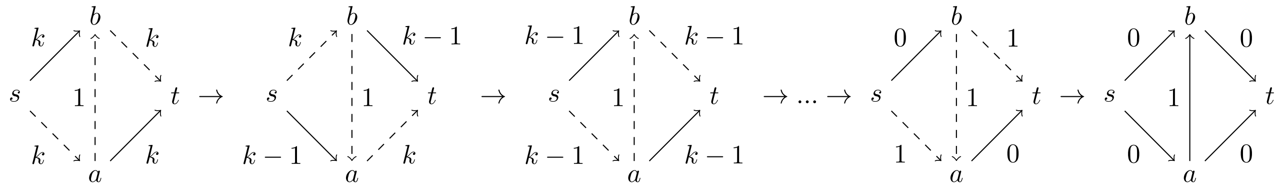
Aufgabe 1

Der Ford-Fulkerson-Algorithmus wählt einen flussvergrößernden Pfad per Tiefensuche. Dabei wird also ein beliebiger, flussvergrößernder Pfad gewählt. Betrachte nun den folgenden Ablauf von Ford-Fulkerson, wobei jeweils die (relevanten) Restkapazitäten angegeben sind.



Die Kapazitäten der äußeren Kanten sind also nach zwei Schritten von k auf $k-1$ gesunken. Nach $2 \cdot k$ Schritten sind die Kapazitäten dieser Kanten von k auf 0 gesunken. Erst in diesem letzten Schritt existiert kein flussvergrößernder Pfad mehr. In jedem Schritt wird der Fluss um 1 erhöht und – da der Fluss jedes flussvergrößernden Pfads im Beispiel durch die mittlere Kante beschränkt ist – daher möglicherweise $2 \cdot k$ Schritte durchgeführt.

Aufgabe 2

Aufgabe 3

Aufgabe 4

(a) maximaler Fluss

Sei f ein Fluss auf dem Netzwerk (D, s, t) mit

$$\begin{array}{rrrr}
f(s,v_1) = 7 & f(s,v_3) = 3 & f(s,v_5) = 8 & f(s,v_6) = 2 \\
f(v_1,v_2) = 3 & f(v_1,v_3) = 4 & f(v_2,v_3) = 1 & f(v_2,v_7) = 0 \\
f(v_2,t) = 5 & f(v_3,v_4) = 8 & f(v_3,v_7) = 0 & f(v_4,t) = 13 \\
f(v_5,v_2) = 3 & f(v_5,v_4) = 5 & f(v_6,v_5) = 0 & f(v_6,v_7) = 2 \\
f(v_7,t) = 2 & & &
\end{array}$$

und dem Betrag 20. Der Fluss ist maximal, da alle eingehenden Kanten von t vollständig genutzt werden. Daraus ergibt sich direkt ein

(b) minimaler Schnitt

Da, wie bereits gezeigt, die eingehenden Kanten von t von f vollständig genutzt werden, ist $\{s, v_1, ..., v_7\}, \{t\}$ ein minimaler Schnitt.

Aufgabe 5

Wir gehen wie folgt vor:

$$\begin{tikzcd}
\nu(G) \arrow[r, dash, "?"] \arrow[d, dash, "1"] & \tau(G) \arrow[d, dash, "2"'] \\
\text{max flow} \arrow[r, dash, "\star"] & \text{min cut}
\end{tikzcd}$$

Aussage $\star$ folgt aus dem Max-Flow-Min-Cut-Theorem.

Zeige Aussage 1:

Sei $G = (A, V = V_1 \cup V_2)$ ein bipartiter Graph. Seien die Kanten in A oBdA gerichtet, also $A \subseteq V_1 \times V_2$. Sei $G' = (A', V')$ mit $V' = V \cup \{s, t\}$ und $A' = A \cup \{s\} \times V_1 \cup V_2 \times \{t\}$. Sei die Kapazität $c(v) = 1$ für alle $v \in V'$.

Analog zum Beweis des Max-Flow-Min-Cut-Theorems (folgend aus Mengers Theorem) gibt es einen Maximalen Fluss, der aus kantendisjunkten Pfaden von s nach t besteht.

Behauptung: Dieser maximale Fluss ist, eingeschränkt auf den Graphen G, ein maximales Matching.

Da die Pfade kantendisjunkt sind und jeder Knoten aus V entweder nur eine eingehende Kante (von s) oder nur eine ausgehende Kante (von t). Somit kann jeder Knoten auch nur maximal eine im Fluss verwendete Kante in A haben. Also ist der auf G eingschränkte Fluss ein Matching.

Wäre der eingeschränkte Fluss kein maximales Matching, so gäbe es eine Kante $(u, v) \in A$, die man dem Matching hinzufügen könnte. Dann gäbe es auch noch keinen Pfad durch u oder v, und der Pfad $(s, u), (u, v), (v, t)$ könnte zu den kantendisjunkten Pfaden hinzugefügt werden. Dann wäre der Fluss jedoch nicht maximal, was einen Widerspruch zur Annahme darstellt.

Somit ist gezeigt, dass aus dem maximalen Fluss auf G' ein maximales Matching auf G gleicher Kardinalität berechnet werden kann.

Zeige Aussage 2:

Betrachte den Graphen $G' = (E, V = V_1 \cup V_2 \cup \{s, t\})$, der mit der für Aussage 1 beschriebenen Transformation aus G hervorgeht. Zeige, dass die Größe eines minimalen Vertex Covers gleich der Größe eines minimalen Schnittes ist.

„$\geq$“:

Sei $C = S_1 \cup S_2$ mit $S_1 \subseteq V_1$ und $S_2 \subseteq V_2$ ein Vertex Cover. Sei nun $(S, V \setminus S)$ mit $S = (V_1 \setminus S_1) \cup S_2 \cup \{s\}$ ein Schnitt. Betrachte alle Kanten in diesem Schnitt:

Alle Kanten, die von s in einen Knoten in S_1 gehen, sind im Schnitt – $|S_1|$ viele.
Alle Kanten, die von einem Knoten in S_2 nach t gehen, sind im Schnitt – $|S_2|$ viele.
Es bleiben Schnittkanten zwischen V_1 und V_2. Kanten von S_1 nach S_2 sind „Rückwärtskanten“ und bleiben daher unberücksichtigt. Kanten von $V_1 \setminus S_1$ nach $V_2 \setminus S_2$ können nicht existieren, da sie durch das Vertex Cover nicht abgedeckt wären.

Somit ist die Größe des Schnitts $|S_1| + |S_2| = |C|$, also ide sie Größe eines minimalen Vertex Covers mindestens so groß wie die eines minimalen Schnitts.

„$\leq$“:

TODO