Maximale Flüsse

Flussnetzwerk: gerichteter Graph $G=(V,E,c)$

mit Kapazitäten c.

$c(u,v) \geq 0 \qquad \forall u,v \in V$ — positive Kapazitäten

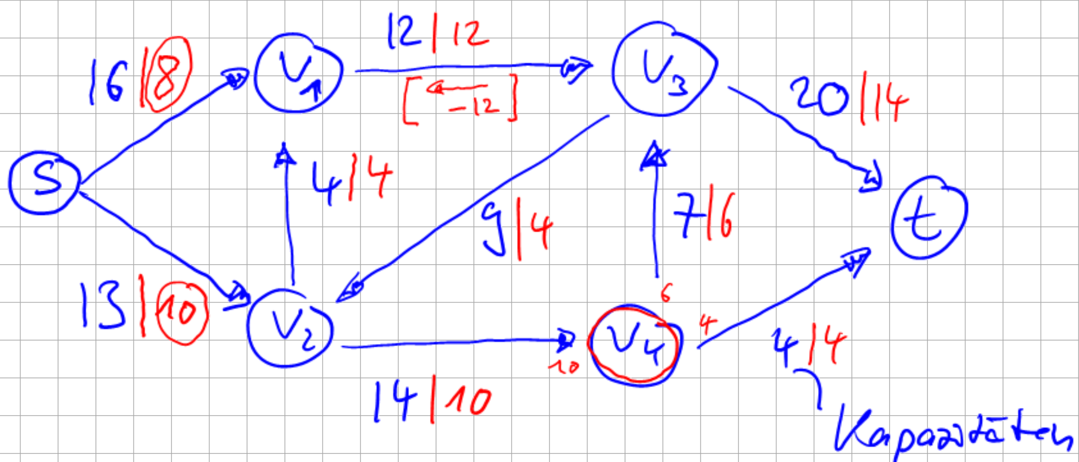
$c(u,v) > 0 \iff (u,v) \in E$ — E sind Kanten mit $c \neq 0$

$(u,v) \in E \Rightarrow (v,u) \notin E$ — keine antiparallele Kanten

und Quelle $s \in V$ und Senke $t \in V$

$\forall u \in V \quad \exists p: s \overset{p}{\leadsto} u \overset{p}{\leadsto} t$ — Jeder Knoten erreichbar von s, jeder Knoten erreicht t.

$|f| = 8+10$
$= 18$



Fluss: $f: V \times V \to \mathbb{R}$ mit Wert $f = \sum_{u \in V} f(s,u)$

Kapazitätsbeschränkung $\quad f(u,v) \leq c(u,v)$

Schiefsymmetrie $\quad f(u,v) = -f(v,u)$

$\Big\}\ \forall u,v \in V$

Flusserhaltung $\quad \forall u \in V - \{s,t\}: \sum_{v \in V} f(u,v) = 0$

gesucht: Fluss f mit maximalem Wert $|f|$.

Notation $\quad f(\mathcal{U},\mathcal{U}') = \sum_{\substack{u \in \mathcal{U} \\ u' \in \mathcal{U}'}} f(u,u') \qquad \mathcal{U},\mathcal{U}' \subseteq V$

$$c(\mathcal{U},\mathcal{U}') = \sum_{\substack{u \in \mathcal{U} \\ u' \in \mathcal{U}'}} c(u,u')$$

Tools:

(a) $f(X,X) = 0 \quad \forall X \subseteq V$ [Schiefsymmetrie]

(b) $f(X,Y) = -f(Y,X) \quad \forall X,Y \subseteq V$ "

(c) $|f| = f(\{s\}, V)$ [Definition]

(d) $f(X, Y \cup Z) = f(X,Y) + f(X,Z)$

wenn $Y \cap Z = \emptyset$ [unmittelbar aus Definition]

(e) $f(R,V) = 0$ wenn $s,t \notin R$

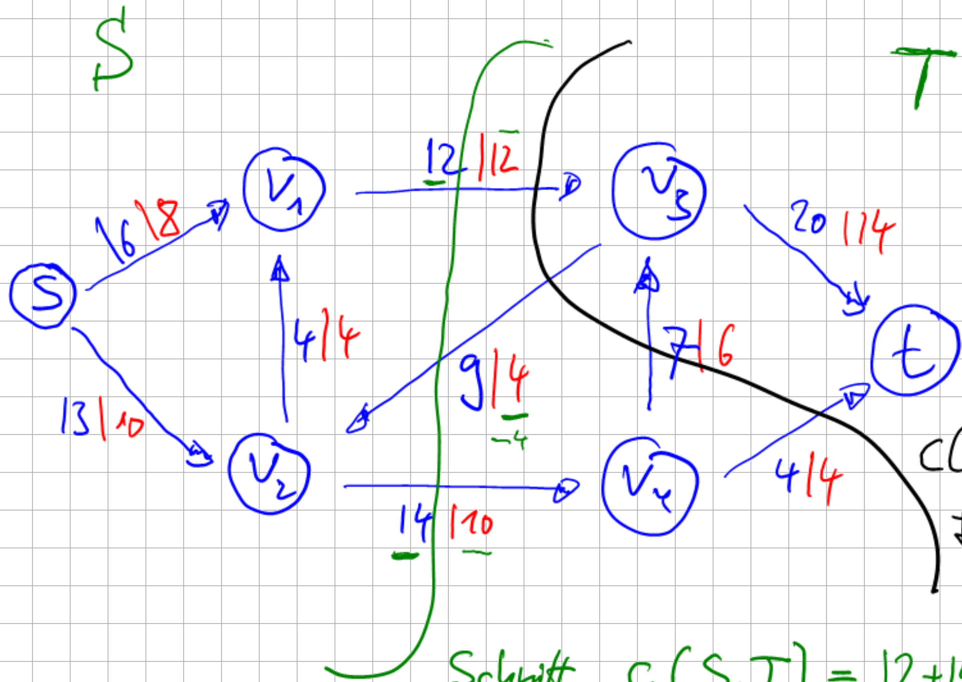
denn $f(R,V) = \sum_{r \in R} \underbrace{\sum_{v \in V} f(r,v)}_{= 0 \text{ (Flusserhaltung)}}$

Schnitt: (S,T): $S \cup T = V \quad s \in S$

$S \cap T = \emptyset \quad t \in T$

Minimaler Schnitt: Schnitt (S,T) mit $c(S,T) = \min$

$|f| = 18$



$c(S,T) = 4+7+12 = 23$

$f(S,T) = 12-4+6+4 = 18$

Schnitt $c(S,T) = 12+14 = 26$

$f(S,T) = 18$

Lemma $f(S,T) = |f| \quad \forall$ Schnitte (S,T)

Beweis $f(S,T) + \underbrace{f(S,S)}_{=0} \overset{(d)}{=} f(S,V)$

$S \cup T = V$

$S \cap T = \emptyset$

$$f(S,V) = \underbrace{f(\underbrace{S \setminus \{s\}}_{\cap \{s,t\} = \emptyset}, V)}_{0} + f(\{s\},V) = |f|.$$

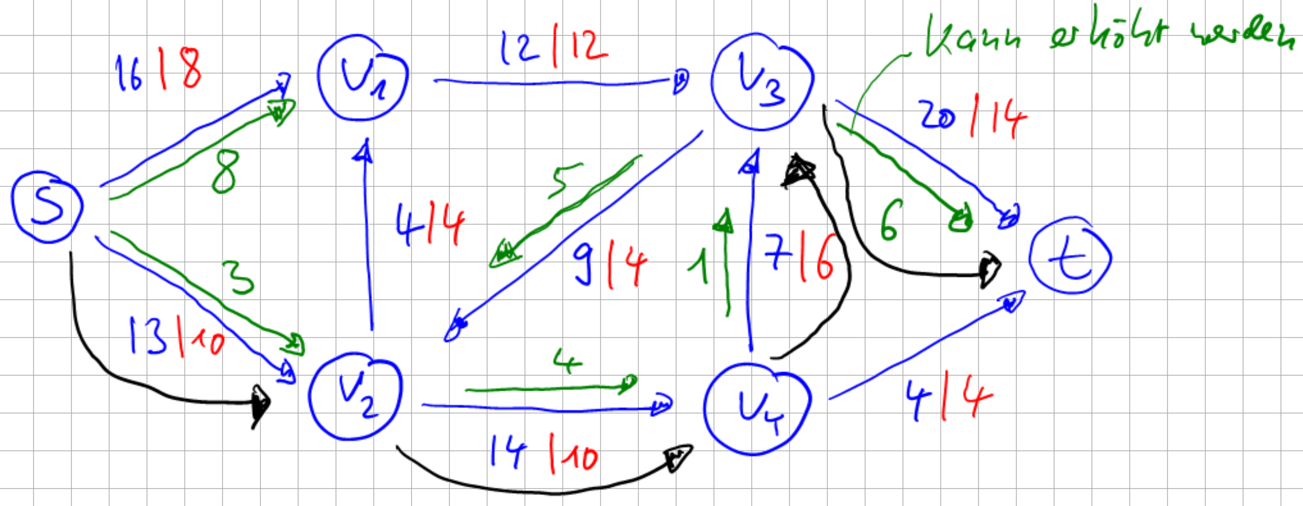
$\blacksquare$

Folgerung $|f| = f(S,T) = \sum_{\substack{u \in S \\ v \in T}} f(u,v) \leq c(S,T)$

$\forall$ Schnitte (S,T)

Werden sehen: sogar $|f| = \min_{\text{Schnitt}(S,T)} c(S,T)$

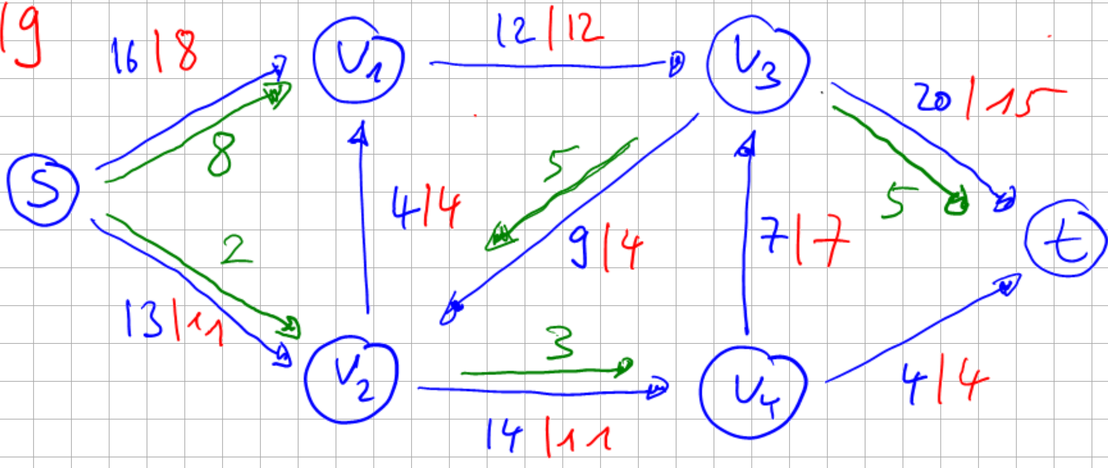
Intuition: Es kann nicht mehr durch das Netz fliessen als der dünnste Querschnitt.



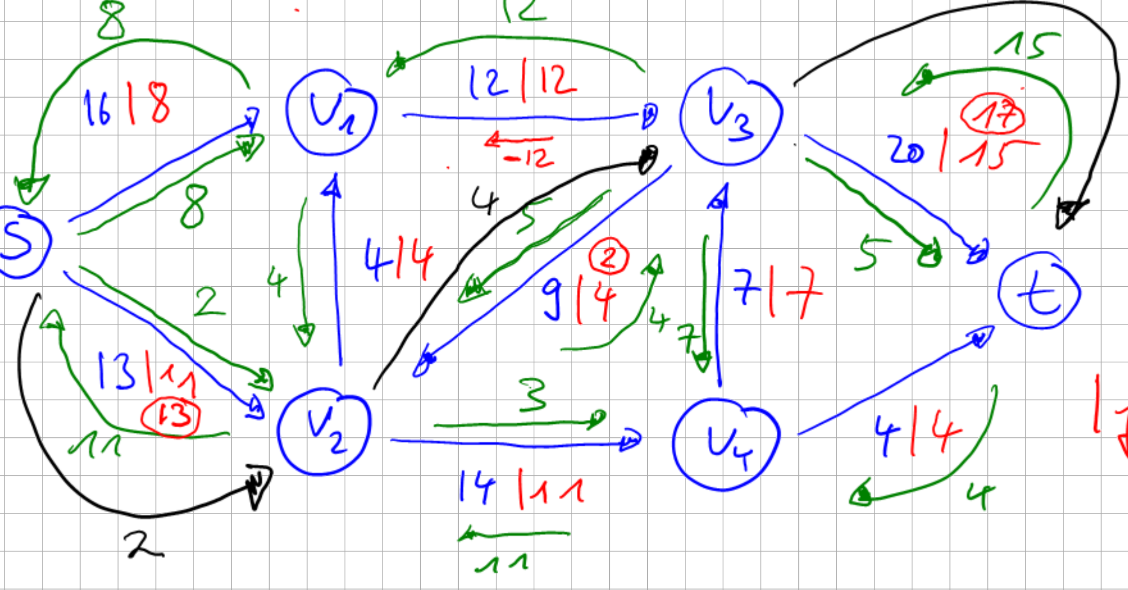
Pfad: $s \xrightarrow{3} v_2 \xrightarrow{4} v_4 \xrightarrow{1} v_3 \xrightarrow{6} t$ erlaubt

Flusserhöhung von 1

$|f| = 19$

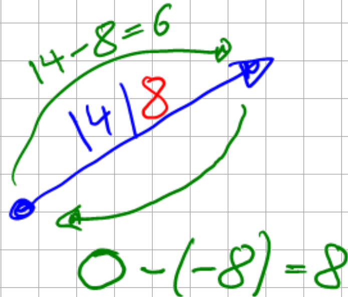


Restnetzwerk



$|f| = 21$

<u>Restnetzwerk</u> $G_f = (V, E_f, c_f)$ mit Restkapazitäten c_f



$$c_f(u,v) := c(u,v) - f(u,v)$$

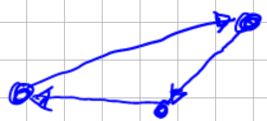
$$(u,v) \in E \Rightarrow c(u,v) \geq f(u,v)$$
$$\Rightarrow c_f(u,v) \geq 0$$

$$(v,u) \in E \Rightarrow f(u,v) < c(u,v) = 0$$
$$\Rightarrow c_f(u,v) = -f(u,v) = f(v,u) \geq 0$$

$$E_f = \{ (u,v) \in V \times V \mid c_f(u,v) > 0 \}$$

Restnetzwerk ist selbst ein Flussnetzwerk, es verletzt nur das Verbot antiparalleler Kanten.

Unproblematisch: das Verbot diente nur der Konstruktion des Restnetzwerkes.



<u>Erweiterungspfad</u>: Pfad $p: s \leadsto t$ in G_f

Wert $c_f(p) := \min\limits_{e \in p} c_f(e)$

<u>Lemma</u>: Der Fluss $f \oplus f_p$ mit

$$f \oplus f_p(u,v) = \begin{cases} f(u,v) + c_f(p) & (u,v) \in p \\ f(u,v) - c_f(p) & (v,u) \in p \\ f(u,v) & \text{sonst} \end{cases} \qquad u,v \in V$$

ist ein gültiger Fluss in G mit $|f \oplus f_p| = |f| + c_f(p)$.

Algorithmus Ford-Fulkerson

Starte mit Fluss $f = 0$

Loop

- Bestimme Restnetzwerk G_f
- Bestimme Pfad $p: s \rightsquigarrow t$ in G_f
- $p = \emptyset \Rightarrow$ return
- Erhöhe Fluss in G: $f \oplus f_p$

Max Flow Min-Cut Theorem

⇒ FF-Algorithmus
⇒ Redundanz von Netzwerken
⇒ Verständnis der Algorithmen

f Fluss in $G(V,E,c)$

(1) f ist maximal

$\Leftrightarrow$ (2) Restnetzwerk G_f hat keinen Erweiterungspfad

$\Leftrightarrow$ (3) $|f| = c(S,T)$ für einen Schnitt (S,T) von G.

---

(3) ⇒ (1) $\quad |f| \le c(S,T) \quad \forall (S,T)$ Schnitt

$\Rightarrow |f| = c(S,T)$ ist maximal

(1) ⇒ (2) $\quad$ sei $|f|$ max und p Erweiterungspfad in G_f

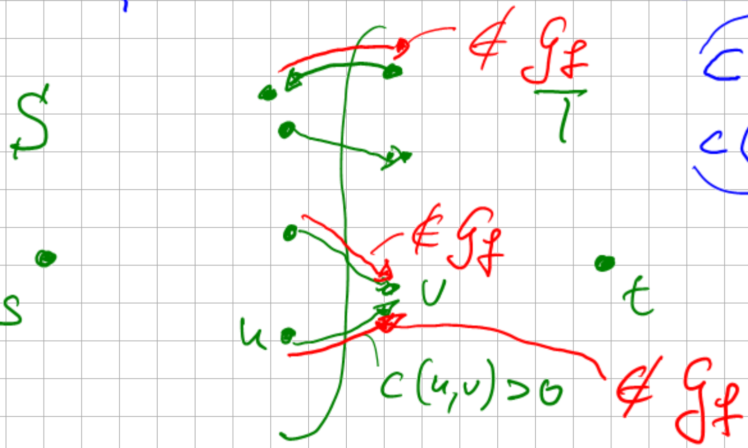
$$\Rightarrow |f \oplus f_p| = |f| + c_p(f) > |f| \quad ⚡$$

(2) ⇒ (3) $\quad$ kein Erweiterungspfad in G_f

def. $S := \{ v \in V \mid \exists p : s \leadsto v \text{ in } G_f \}$

$\Rightarrow s \in S, \; t \notin S$

$(S, \underbrace{V \setminus S}_{T})$ ist ein Schnitt von G



$$\overbrace{\begin{array}{l} c(u,v) > 0, \; c_f(u,v) = 0 \end{array}}^{f(u,v) = c(u,v)}$$

$$\underbrace{\begin{array}{l} c(v,u) > 0, \; c_f(u,v) = 0 \end{array}}_{}$$

$f(u,v) = 0$

$[u \in S, \; v \in T]$

$$|f| = f(S,T) = \sum_{\substack{u \in S \\ v \in T}} f(u,v) = \sum_{\substack{u \in S \\ v \in T}} c(u,v) = c(S,T)$$

$\blacksquare$

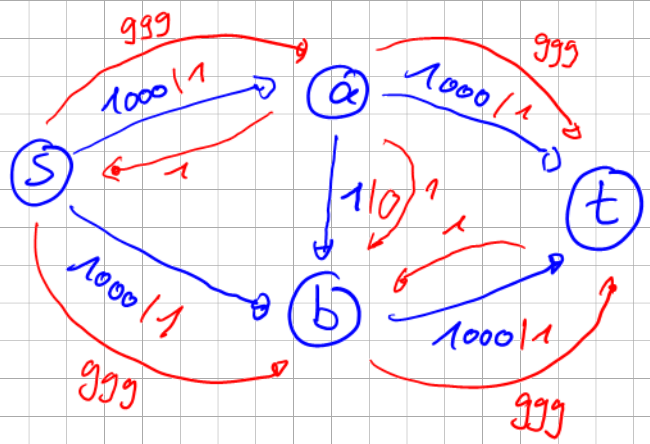
Laufzeit vom FF Algorithmus

$|E| = m$
$|V| = n$

(1) Für irrationale Kapazitäten kann es passieren, dass FF-Algorithmus nicht terminiert

(2) Für ganzzahlige Kapazitäten terminiert FF

(3) " rationale " "



$$\left.\begin{aligned} c_f(p) &= 1 \\ c_f(p) &= 1 \\ &\vdots \end{aligned}\right\} 2000$$

$|f_{max}|$

$$\Rightarrow \Theta(|f_{max}| \cdot m) \qquad m = |E|$$

$\Theta(m+n)$

$n \in O(m)$

jeder Schritt verbessert den Fluss um 1 $\Rightarrow$ f_{max} Schritte

jeder Schritt bedeutet Breitensuche/Tiefensuche $O(|E|)$

$$\Rightarrow \Theta(\underline{|f_{max}|} \cdot m)$$

Edmonds-Karp-Algorithmus

FF mit Breitensuche

Maximal $O(n \cdot m)$ Flusserhöhungen

$\Rightarrow$ Laufzeit $O(m^2 \cdot n)$ $\quad [O(n^5)]$

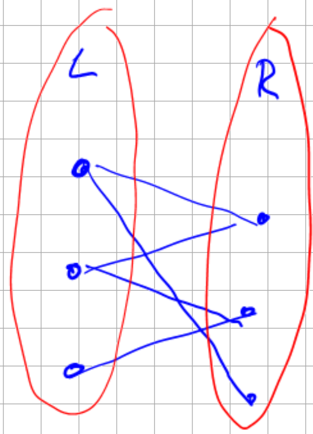
Anwendung: maximales Matching

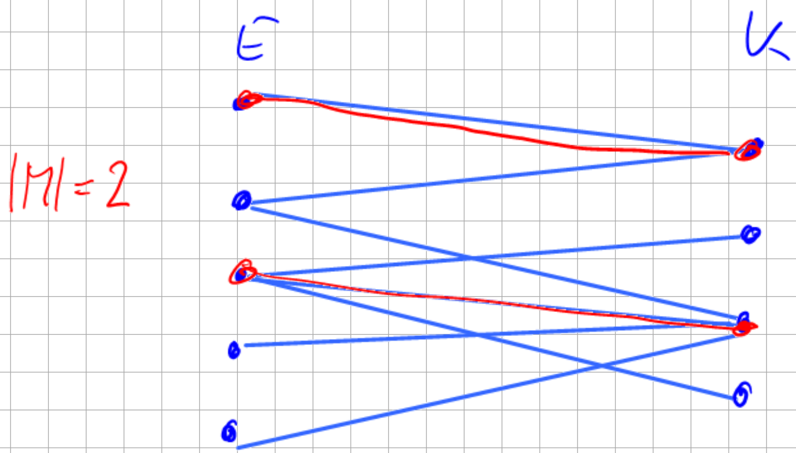
bipartiter Graph $\mathcal{G} = (V, E)$

$\exists\, L \cup R = V, \quad L \cap R = \emptyset$

$\forall (u,v) \in E \;:\; u \in R,\ v \in L$

oder $v \in R,\ u \in L$





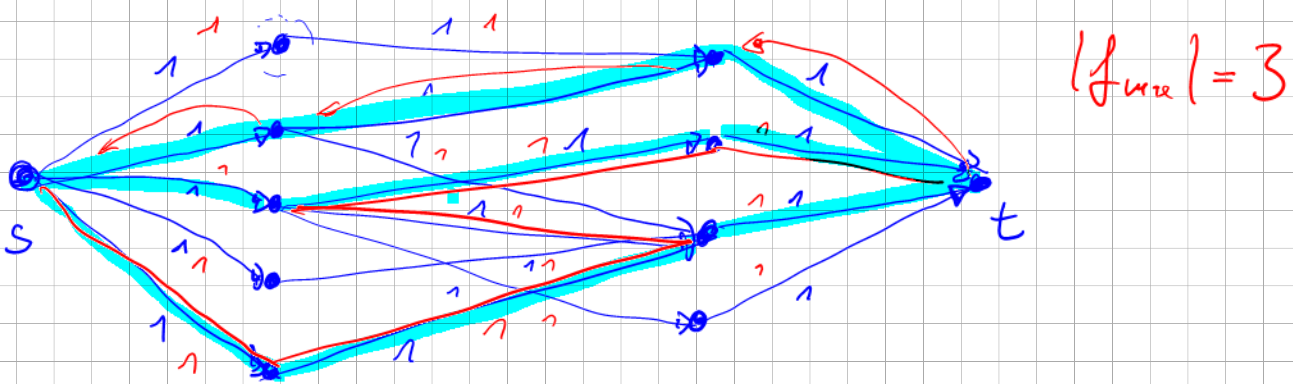
Achterbahn:

- Kinder fahren nur mit bekannten Erwachsenen
- Jeder E und jedes Kind fährt maximal 1x

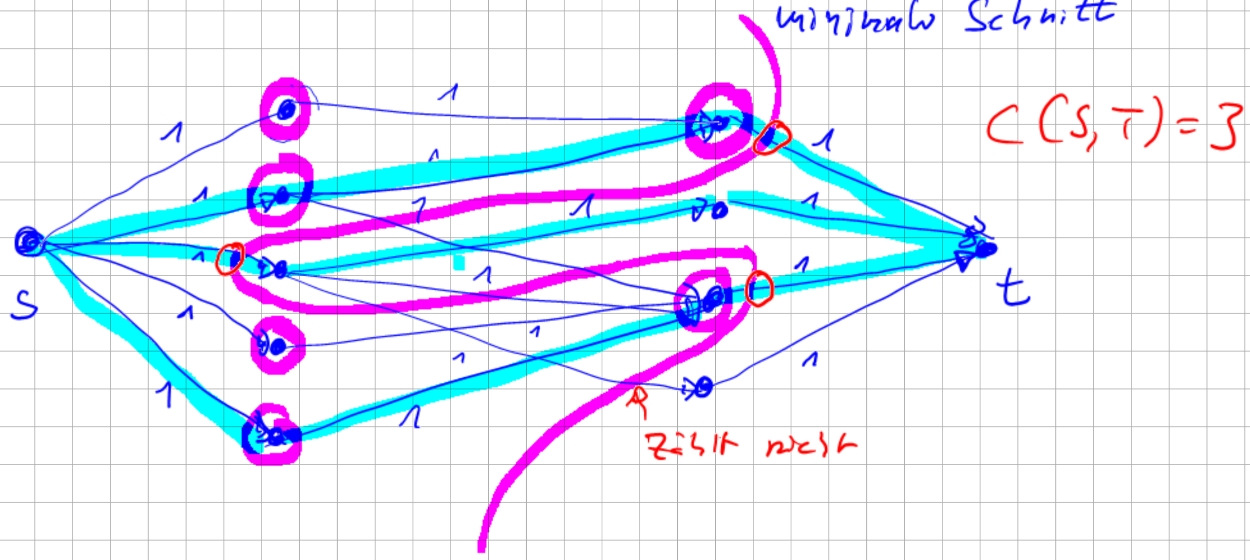
Wie viele Kinder dürfen / können maximal Achterbahn fahren

Matching $M \subseteq E : |\{e \in M \mid u \in e\}| \leq 1 \ \forall u \in V$

Max. Matching ist M mit max. $|M|$



Max. Matching $\hat{=}$ Finde max Flow


minimaler Schnitt
c(S,T) = 3
s
t
zählt nicht