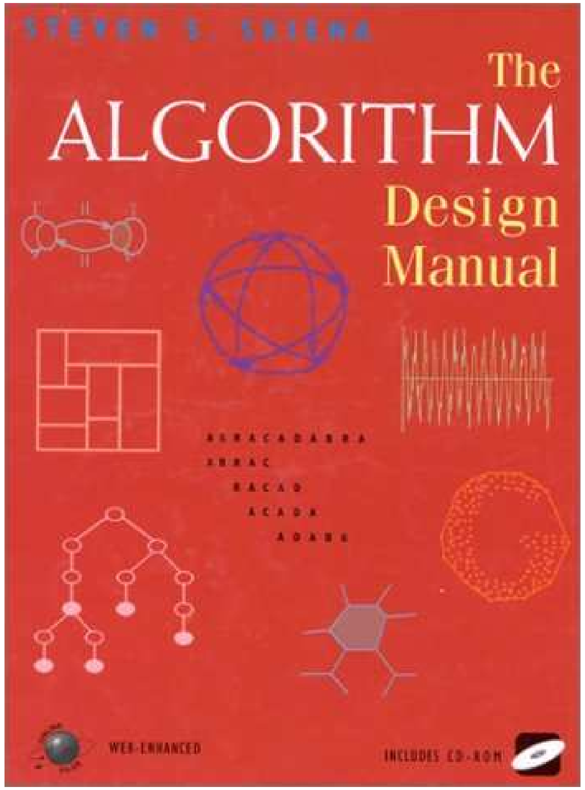
Inhalt

1. Flußprobleme
2. Matching
3. Lineares Programmieren
4. Ganzzahliges Programmieren
5. *NP*-Vollständigkeit
6. Approximationsalgorithmen
7. Backtracking und Branch-and-Bound
8. Randomisierte Algorithmen
9. Online-Algorithmen

Literatur

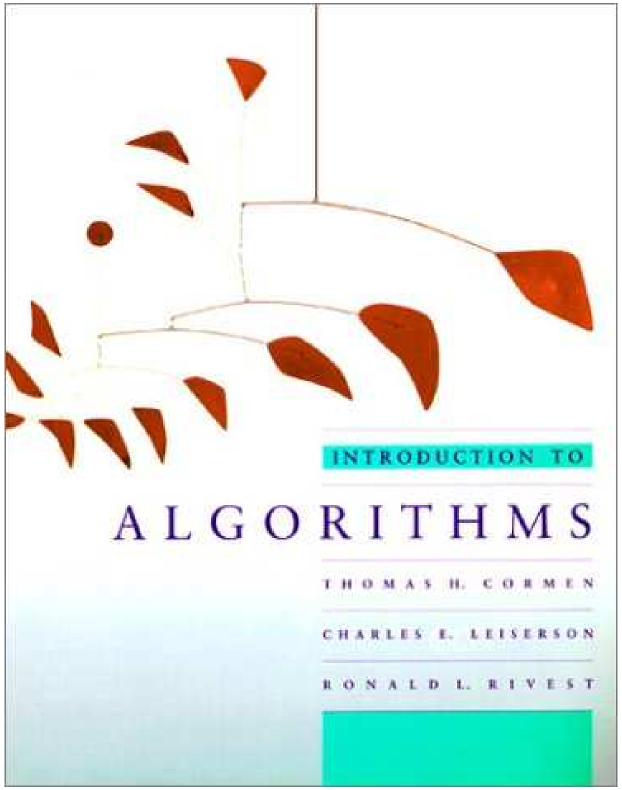


Sehr empfehlenswertes Buch. Es enthält einen Überblick über die wichtigsten Entwurfstechniken für Algorithmen und einen sehr vollständigen Überblick über bekannte Algorithmen mit Verweisen.

Steven S. Skiena: *The Algorithm Design Manual.* Springer Verlag.

Preis: US $69.95, Hardcover mit CD-ROM.

Literatur



Sehr umfangreiches, modernes Buch.
Sehr gut geschrieben.
Enthält alle wichtigen Algorithmen.
Nur ein Buch $\Rightarrow$ dieses!

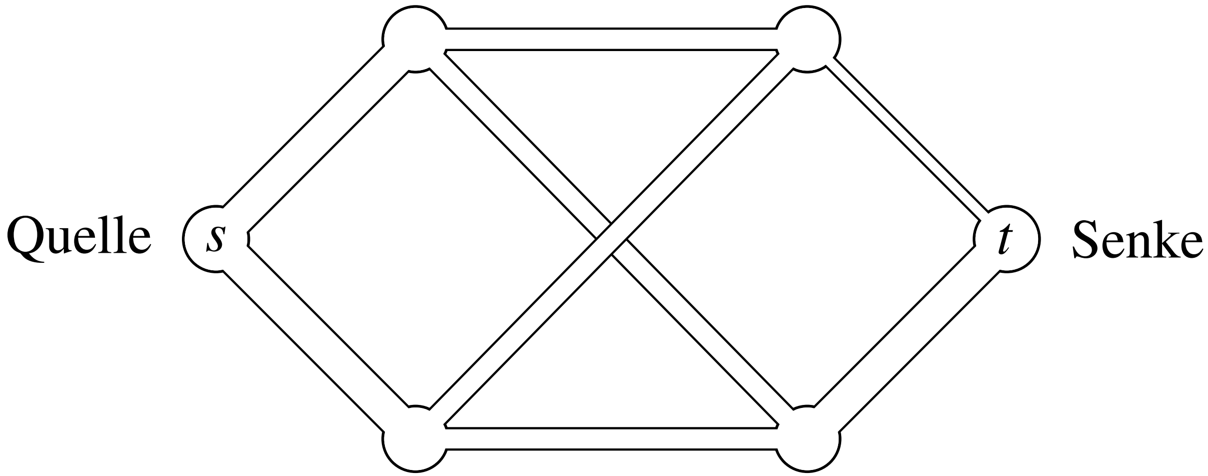
Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest:
Introduction To Algorithms. MIT Press.

Preis: US $69.95, Hardcover, 1028 Seiten!

Netzwerkfluß

(Wiederholung der Grundlagen aus DSAL)

Gegeben ist ein System von Wasserrohren:

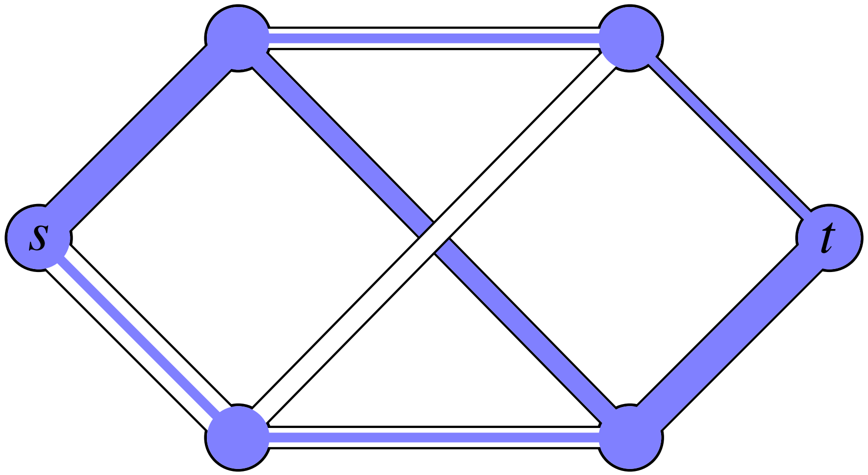


Die Kapazität jedes Rohres ist 3, 5 oder $8\,l/s$.

Frage: Wieviel Wasser kann von der Quelle zur Senke fließen?

Netzwerkfluß

Antwort: Maximal 11 l/s sind möglich.



s-t-Netzwerke

Ein *s-t-Netzwerk* (flow network) ist ein gerichteter Graph $G = (V, E)$, wobei

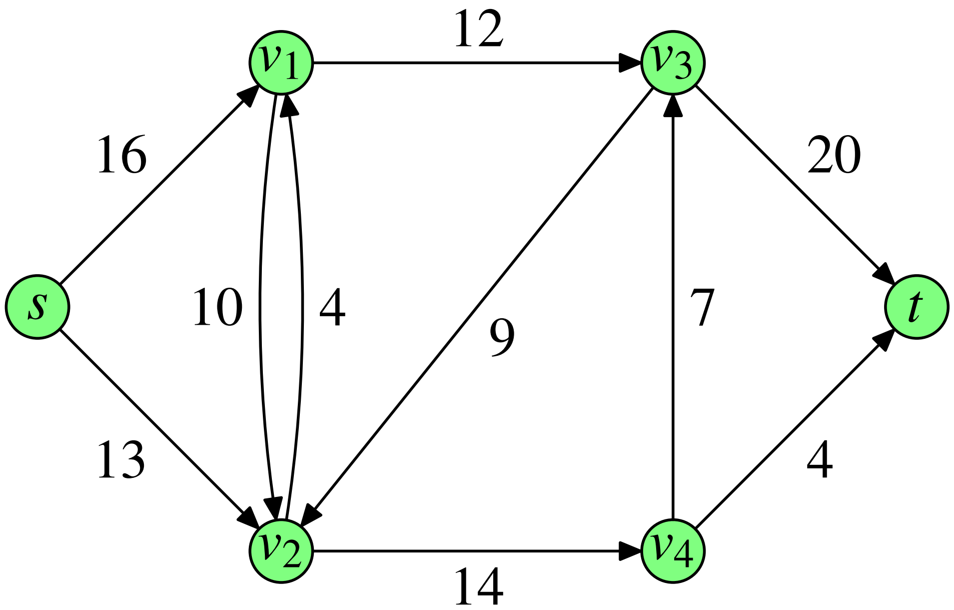
1. jede Kante $(u, v) \in E$ eine *Kapazität* $c(u, v) \geq 0$ hat,
2. es eine *Quelle* $s \in V$ und eine *Senke* $t \in V$ gibt.

Es ist bequem, anzunehmen daß jeder Knoten auf einem Pfad von s nach t liegt.

Falls $(u, v) \notin E$ setzen wir $c(u, v) = 0$.

Es kann Kanten (u, v) und (v, u) mit verschiedener Kapazität geben.

Beispiel eines s-t-Netzwerks



Die Kanten sind mit den Kapazitäten $c(u, v)$ beschriftet.

Flüsse

Ein *Fluß* ist eine Funktion $f\colon V \times V \to \mathbf{R}$, die Paare von Knoten auf reelle Zahlen abbildet und diese Bedingungen erfüllt:

- *Zulässigkeit:* Für $u, v \in V$ gilt $f(u,v) \le c(u,v)$.
- *Symmetrie:* Für $u, v \in V$ gilt $f(u,v) = -f(v,u)$.
- *Flußerhaltung:* Für $u \in V - \{s,t\}$ gilt $\displaystyle\sum_{v \in V} f(u,v) = 0$.

Der *Wert* $|f|$ eines Flusses ist definiert als $|f| = \displaystyle\sum_{u \in V} f(s,u)$.

Dies ist gerade der Gesamtfluß aus der Quelle heraus.

Maximale Flüsse

Das Problem des *maximalen Flusses*:

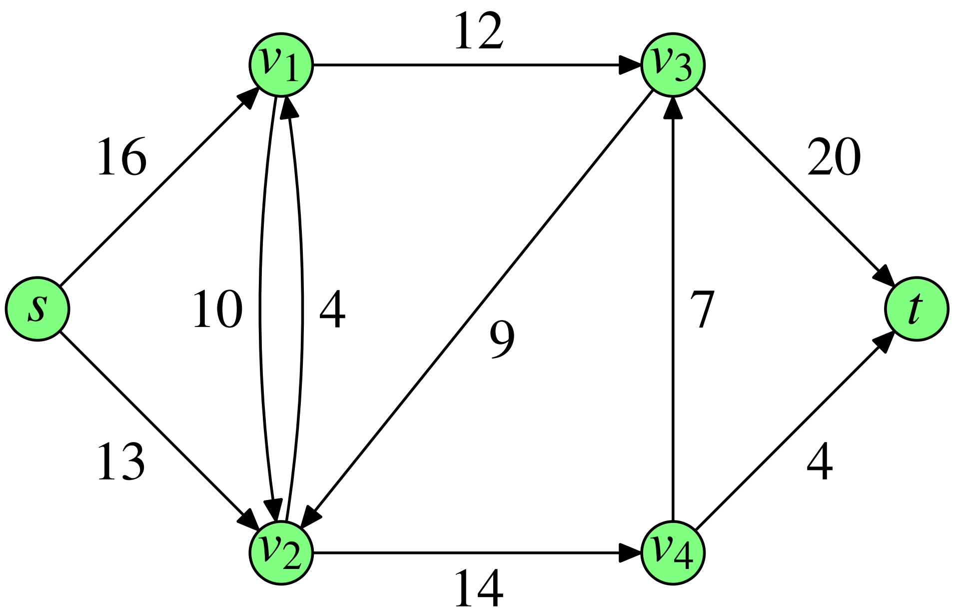
Gegeben:

Ein s-t-Netzwerk.

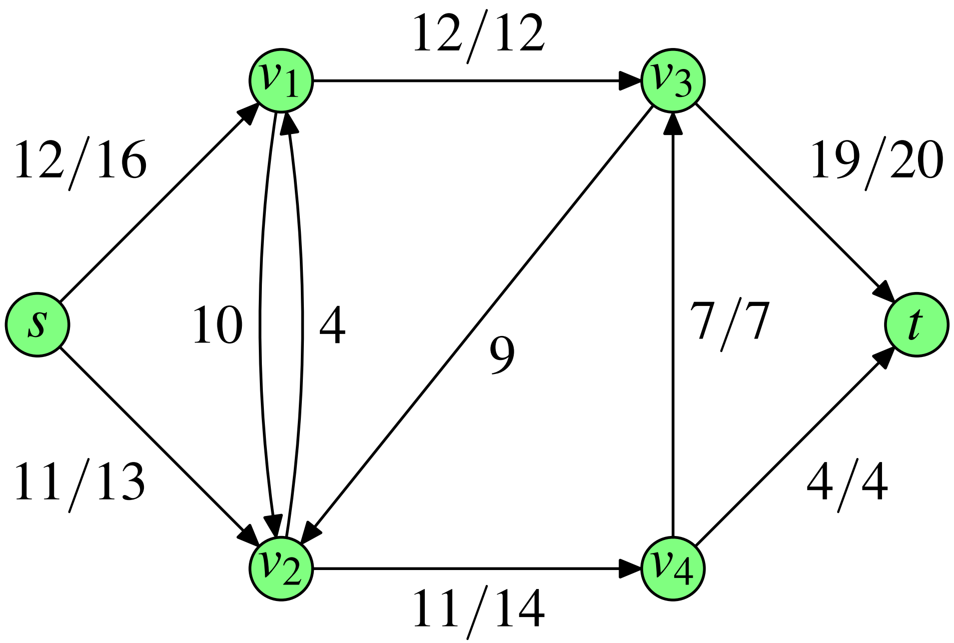
Gesucht:

Ein Fluß mit maximalem Wert.

Was ist der maximale Fluß?

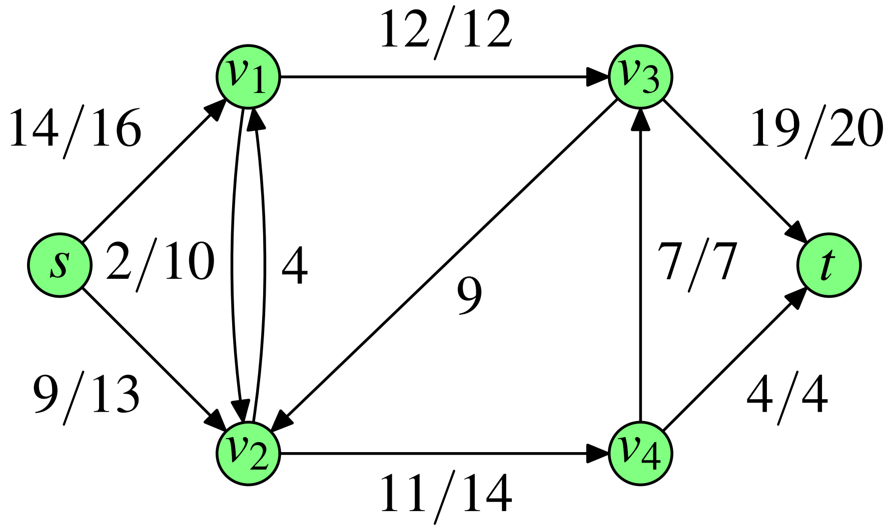


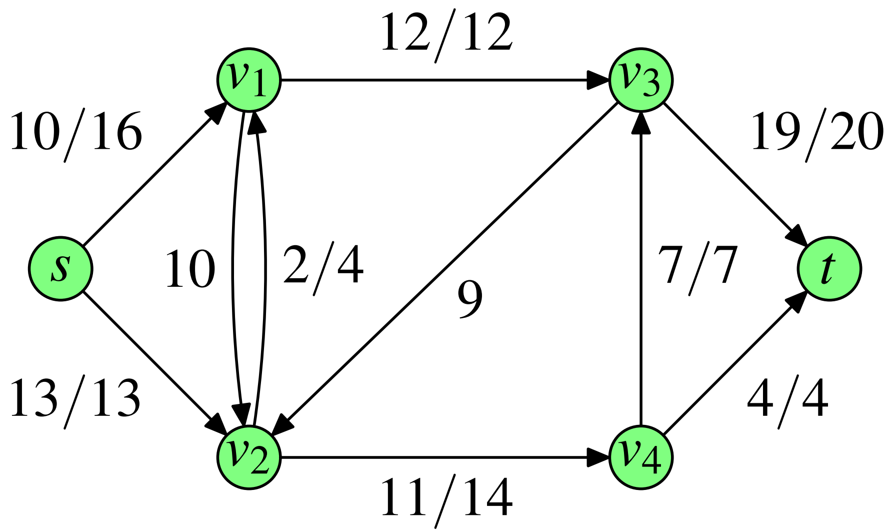
Der maximale Fluß ist 23



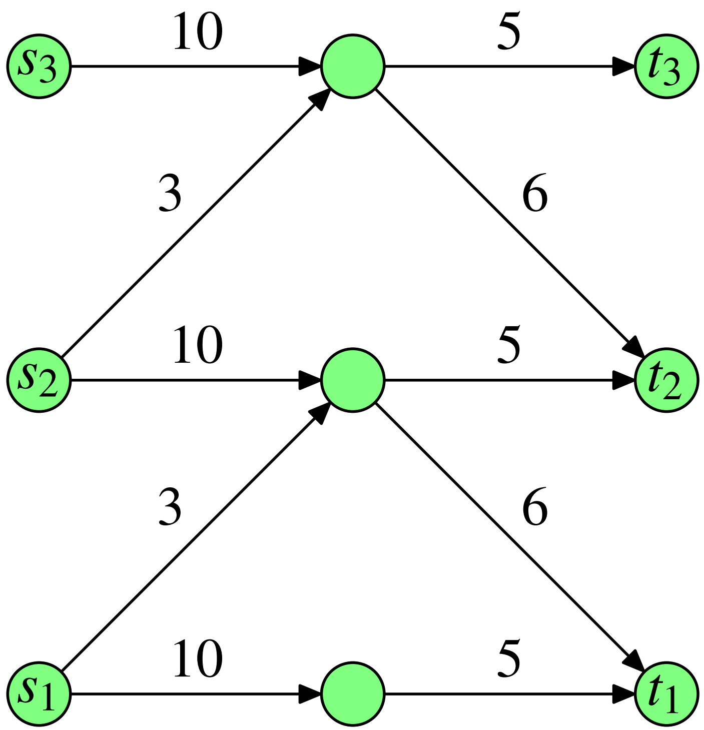
Die Kanten sind mit $c(u, v)$ beschriftet oder mit $f(u, v)/c(u, v)$, falls $f(u, v) > 0$.

Andere optimale Lösungen



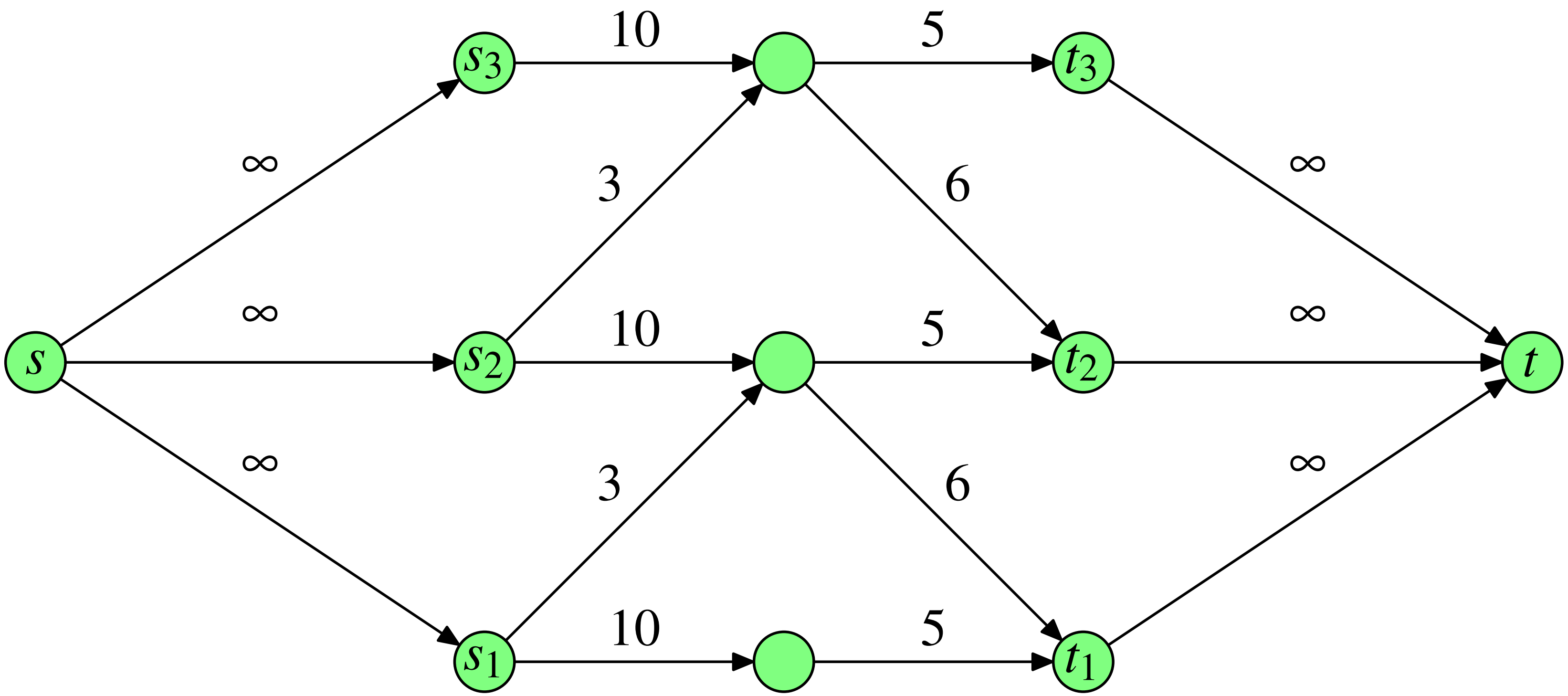


Mehrfache Quellen oder Senken



Mehrfache Quellen oder Senken sind eine **Verallgemeinerung** des Problem des maximalen Flusses.

Mehrfache Quellen oder Senken



$\rightarrow$ Neue „Superquelle“ und „Supersenke“ hinzufügen

Existenz des maximalen Flusses

Existiert stets der maximale Fluß

$$\max\left\{\, |f| \;\middle|\; f \text{ ist ein } s\text{-}t\text{-Fluß in } G \,\right\}?$$

Ja, denn die Menge aller Flüsse ist abgeschlossen im $\mathbf{R}^m$ und sie ist nicht leer.

Die stetige Funktion, die einen Fluß auf ihren Wert abbildet, hat daher ein Maximum:

$$|f| = \sum_{u \in V} f(s,u) \text{ ist stetig!}$$

Einige Notationen

Es ist bequem einige Abkürzungen zu verwenden:

- $f(X, Y) = \sum_{x \in X} \sum_{y \in Y} f(x, y)$ für $X, Y \subseteq V$
- $f(x, Y) = \sum_{y \in Y} f(x, y)$ für $Y \subseteq V$
- $f(X, y) = \sum_{x \in X} f(x, y)$ für $X \subseteq V$
- $X - y$ statt $X - \{y\}$

Lemma A

Falls f ein s-t-Fluß für $G = (V, E)$ ist, dann gilt:

1. $f(X, X) = 0$ für $X \subseteq V$
2. $f(X, Y) = -f(Y, X)$ für $X, Y \subseteq V$
3. $f(X \cup Y, Z) = f(X, Z) + f(Y, Z)$ für $X, Y, Z \subseteq V$ mit $X \cap Y = \emptyset$
4. $f(Z, X \cup Y) = f(Z, X) + f(Z, Y)$ für $X, Y, Z \subseteq V$ mit $X \cap Y = \emptyset$

Dieses Lemma ist sehr nützlich, um wichtige Eigenschaften über Flüsse abzuleiten.

Lemma A

Um Lemma A zu beweisen, dürfen wir nur die Eigenschaften eines s-t-Flusses verwenden, also Zulässigkeit, Symmetrie und Flußerhaltung.

Beweis für $f(X,X) = 0$:

$$\begin{aligned} f(X,X) &= \frac{1}{2}\Big(\sum_{x_1 \in X}\sum_{x_2 \in X} f(x_1,x_2) + \sum_{x_1 \in X}\sum_{x_2 \in X} f(x_1,x_2)\Big) \\ &= \frac{1}{2}\Big(\sum_{x_1 \in X}\sum_{x_2 \in X} f(x_1,x_2) + \sum_{x_1 \in X}\sum_{x_2 \in X} f(x_2,x_1)\Big) \\ &= \frac{1}{2}\sum_{x_1 \in X}\sum_{x_2 \in X}\Big(f(x_1,x_2) + f(x_2,x_1)\Big) = 0 \end{aligned}$$

Hier genügt die Symmetrie allein! (Rest als Übungsaufgabe.)

Anwendung des Lemmas

Der Fluß in die Senke sollte intuitiv dem Fluß aus der Quelle entsprechen:

$$f(s, V) = f(V, t)$$

Beweis mit Lemma A:

$$\begin{aligned} f(s, V) &= f(V, V) - f(V - s, V) \\ &= -f(V - s, V) \\ &= f(V, V - s) \\ &= f(V, t) + f(V, V - s - t) \\ &= f(V, t) \text{ wegen Flußerhaltung} \end{aligned}$$

Residualnetzwerke

„Netzwerk minus Fluß = Residualnetzwerk“

Definition:

Gegeben ist ein Netzwerk $G = (V, E)$ und ein Fluß f. Das *Residualnetzwerk* $G_f = (V, E_f)$ zu G und f ist definiert vermöge

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

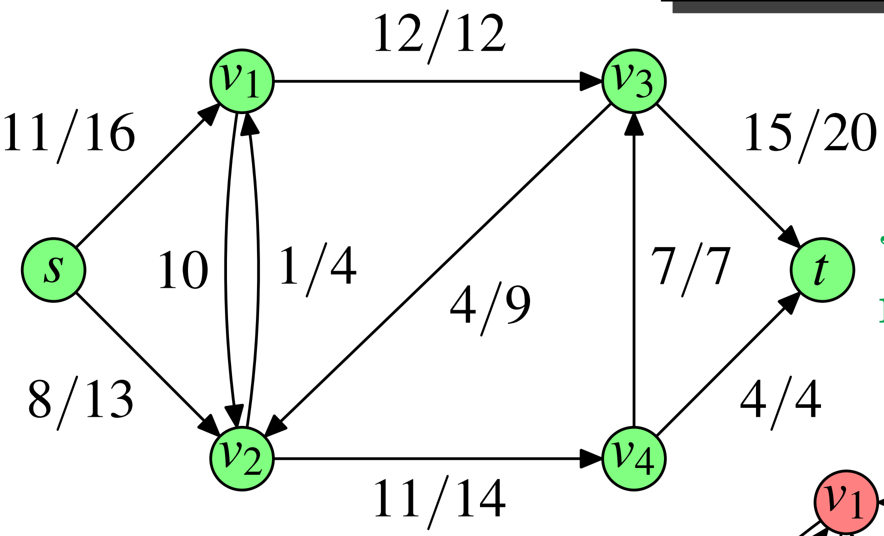
wobei

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

c_f ist die *Restkapazität*.

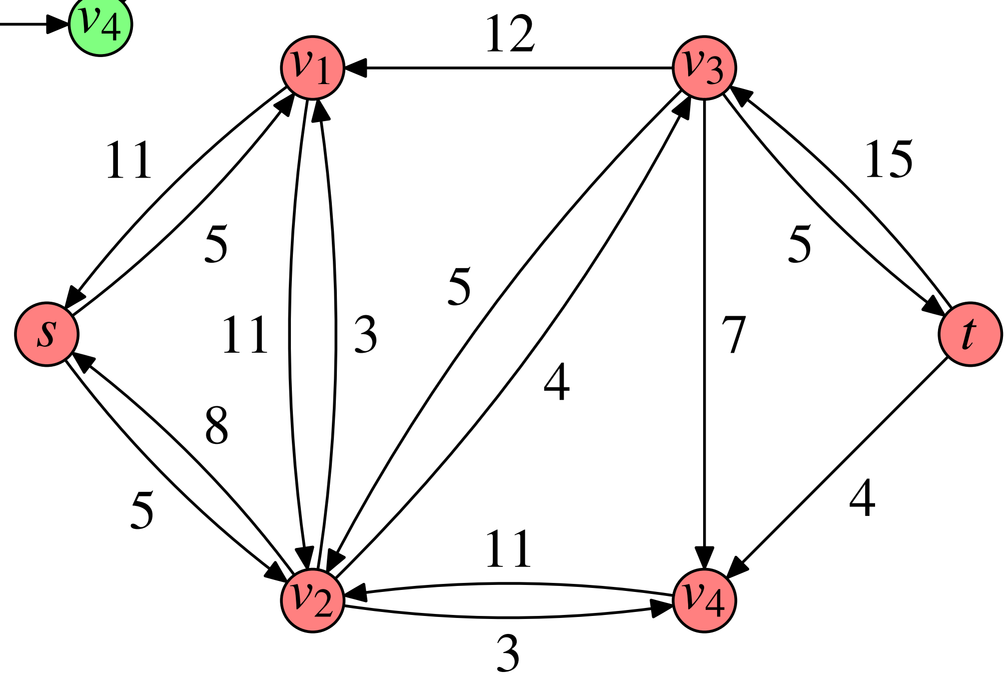
Das s-t-Netzwerk G_f hat die Kapazitäten c_f.

Beispiel



s-t-Netzwerk mit Fluß f

Residualnetzwerk G_f

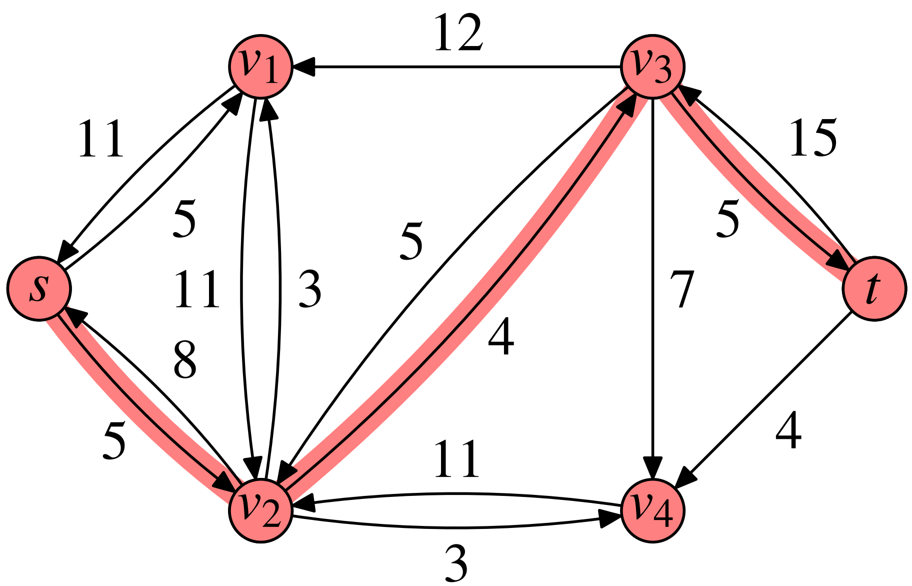


Augmentierende Pfade

Ein s-t-Pfad p in G_f heißt *augmentierender Pfad*.

$c_f(p) = \min\{\, c_f(u,v) \mid (u,v) \text{ ist auf } p\,\}$ heißt *Restkapazität* von p.

Beispiel:



Die Restkapazität dieses Pfades ist 4.

Die Ford–Fulkerson–Methode

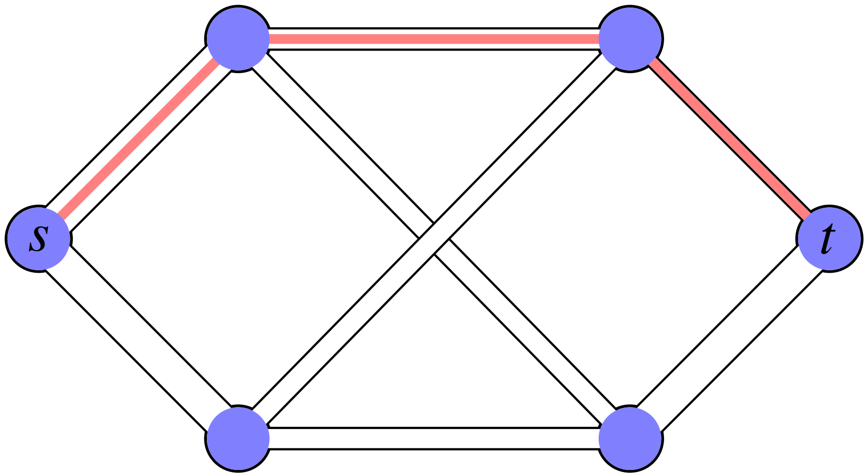
Initialisiere Fluß f zu 0
while es gibt einen augmentierenden Pfad p
  do augmentiere f entlang p
return f

$$f_p(u,v) = \begin{cases} c_f(p) & \text{falls } (u,v) \text{ auf } p \\ -c_f(p) & \text{falls } (v,u) \text{ auf } p \\ 0 & \text{sonst} \end{cases}$$

Augmentiere f entlang p: $f := f + f_p$

f_p ist ein Fluß in G_f

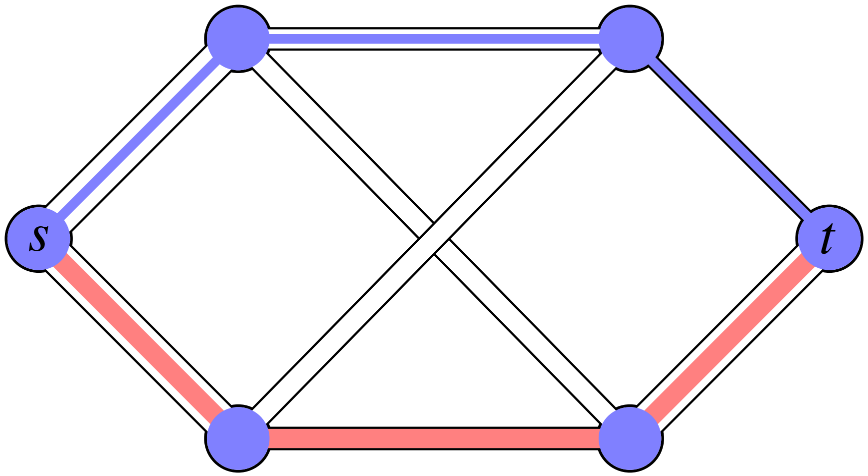
Die Ford–Fulkerson–Methode



Anfangs ist der Fluß 0.

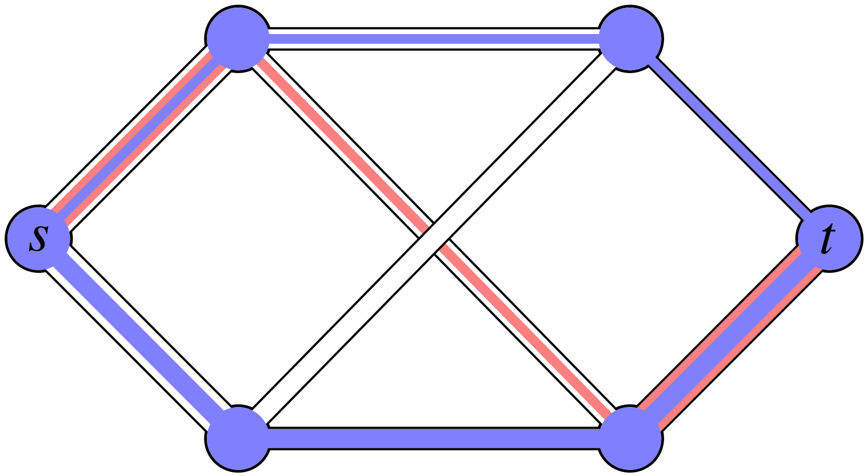
Der augmentierende Pfad ist rot eingezeichnet.

Die Ford–Fulkerson–Methode



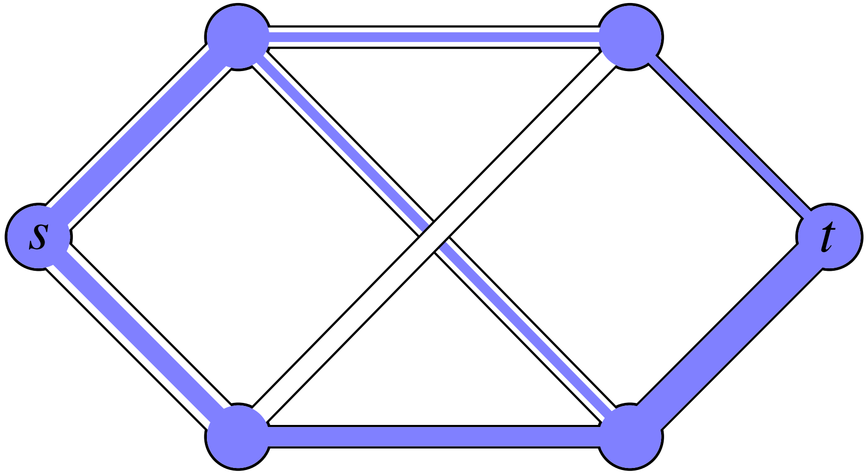
Der augmentierende Pfad hat Kapazität 5.

Die Ford–Fulkerson–Methode



Der augmentierende Pfad hat Kapazität 3.

Die Ford–Fulkerson–Methode



Jetzt gibt es keinen augmentierenden Pfad mehr.

Der Fluß ist maximal.

Korrektheit

Lemma B

Set $G = (V, E)$ ein s-t-Netzwerk und f ein Fluß in G.

Sei f' ein Fluß in G_f.

Dann ist $f + f'$ ein Fluß in G.

Konsequenz:

Die Ford–Fulkerson–Methode berechnet einen Fluß.

Beweis:

Wir müssen zeigen, daß $f + f'$ zulässig, symmetrisch und flußerhaltend ist.

Beweis (Symmetrie)

$$
\begin{aligned}
(f+f')(u,v) &= f(u,v)+f'(u,v)\\
&= -f(v,u)-f'(v,u)\\
&= -(f(v,u)+f'(v,u))\\
&= -(f+f')(v,u)
\end{aligned}
$$

Beweis (Flußerhaltung)

Sei $u \in V - \{s, t\}$.

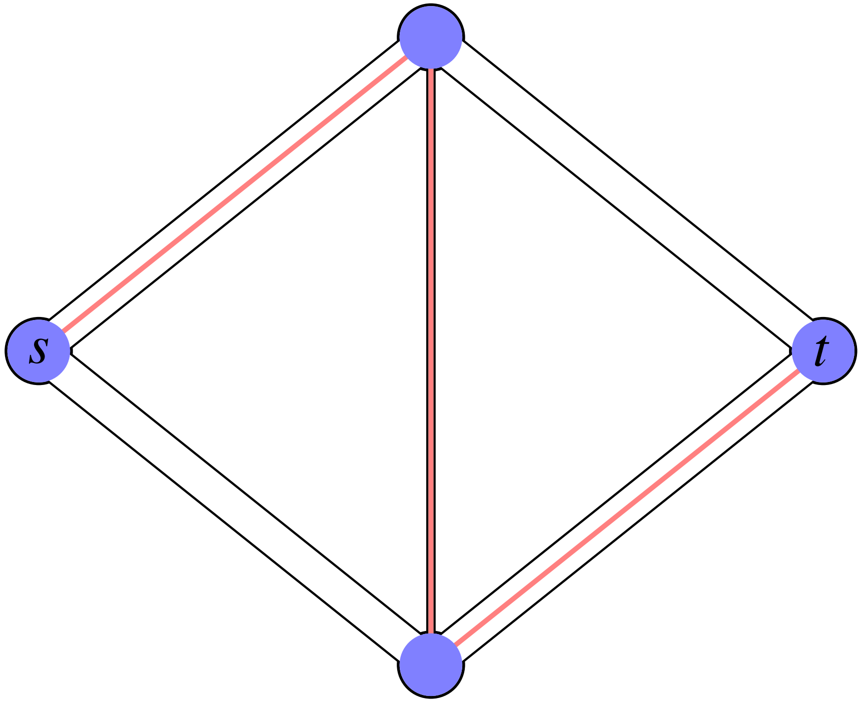
$$\begin{aligned} (f + f')(u, V) &= f(u, V) + f'(u, V) \\ &= 0 + 0 \\ &= 0 \end{aligned}$$

Beweis (Zulässigkeit)

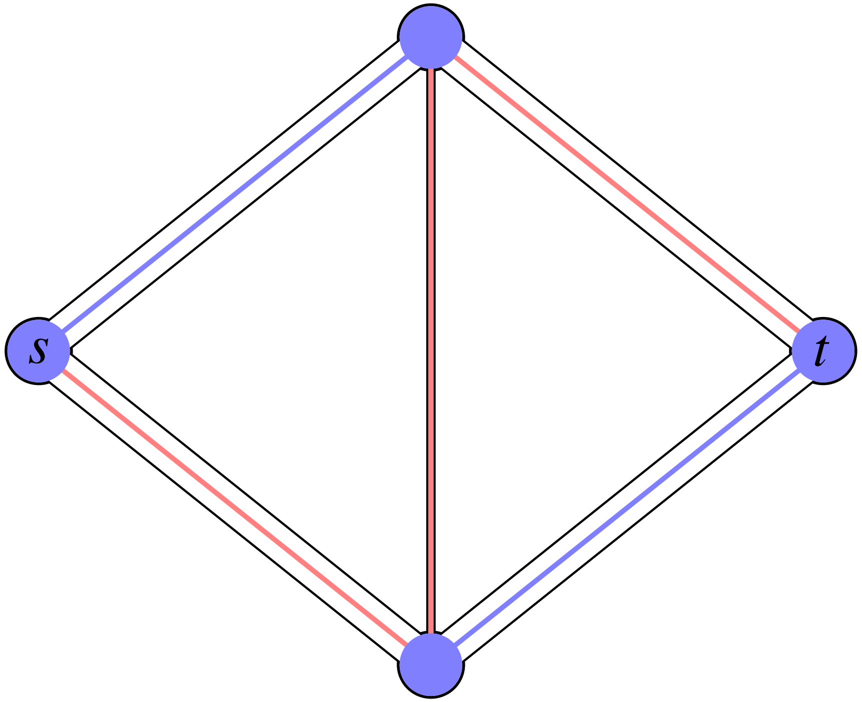
$$\begin{aligned}(f+f')(u,v) &= f(u,v)+f'(u,v)\\ &\leq f(u,v)+c_f(u,v)\\ &= f(u,v)+(c(u,v)-f(u,v))\\ &= c(u,v)\end{aligned}$$

Der Beweis verwendet, daß f' ein Fluß in G_f ist, aber nicht, daß f ein Fluß in G ist.

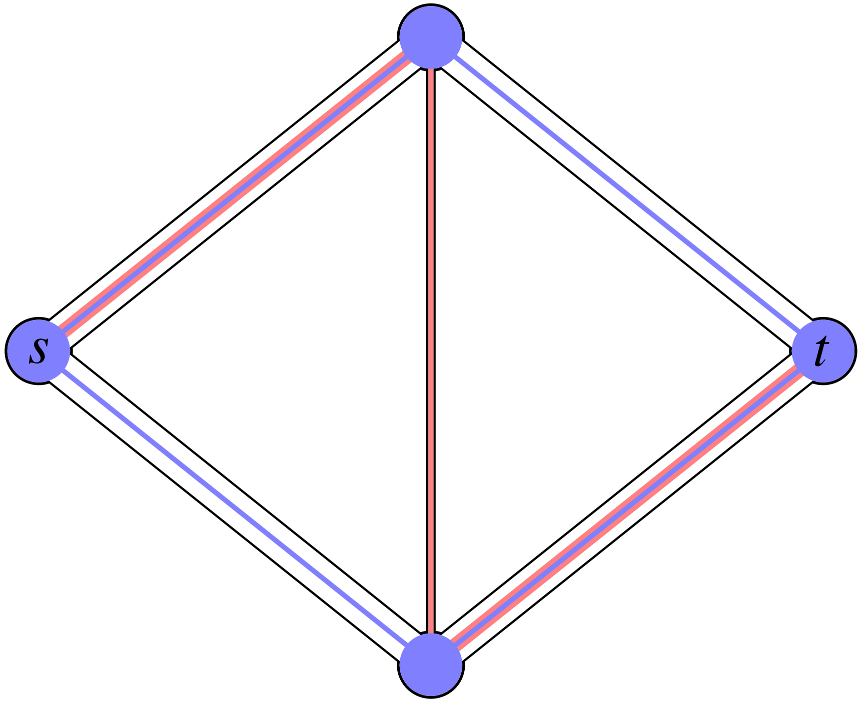
Laufzeit der Ford–Fulkerson–Methode



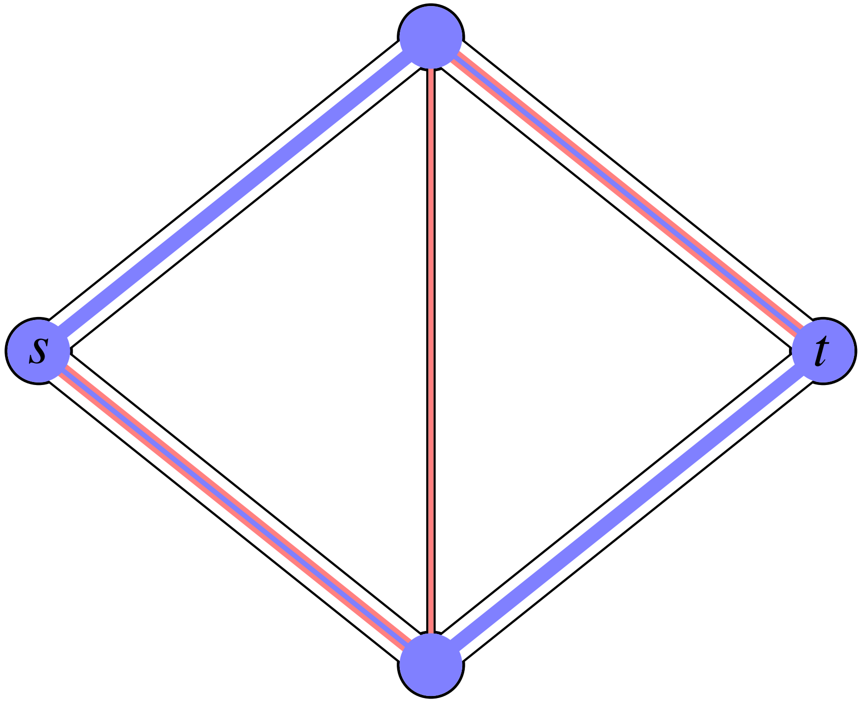
Laufzeit der Ford–Fulkerson–Methode



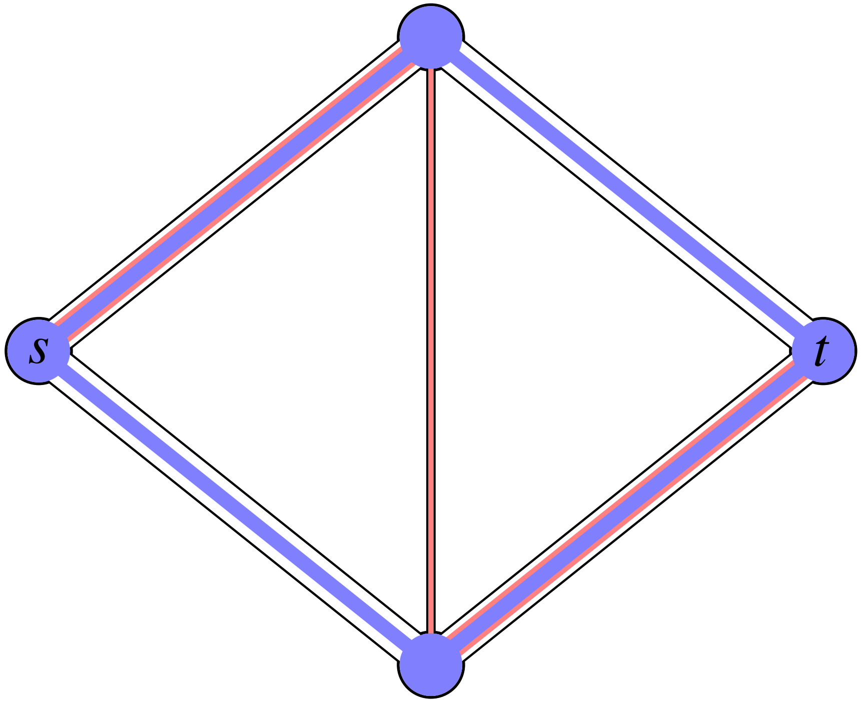
Laufzeit der Ford–Fulkerson–Methode



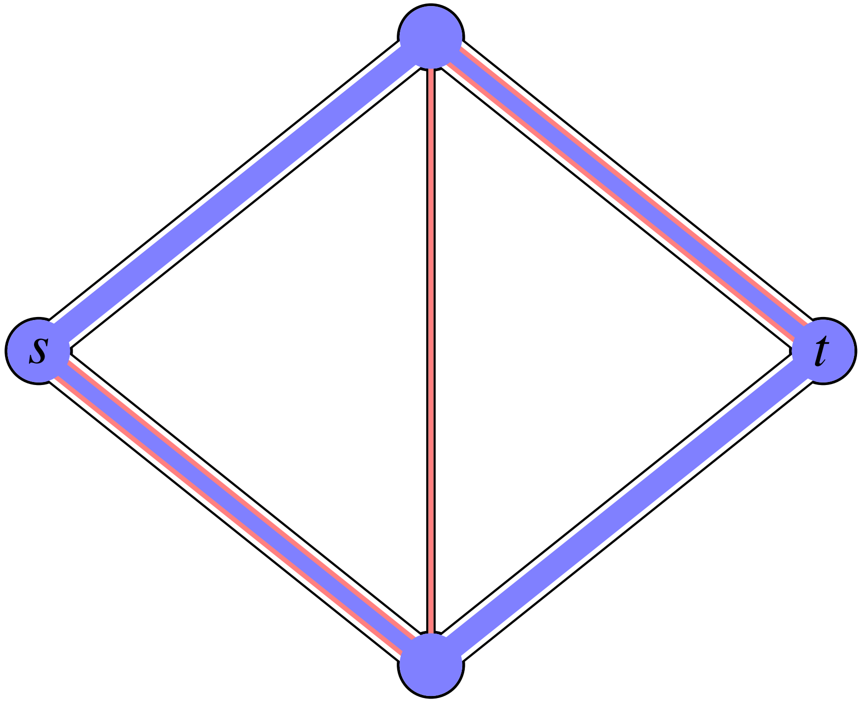
Laufzeit der Ford–Fulkerson–Methode



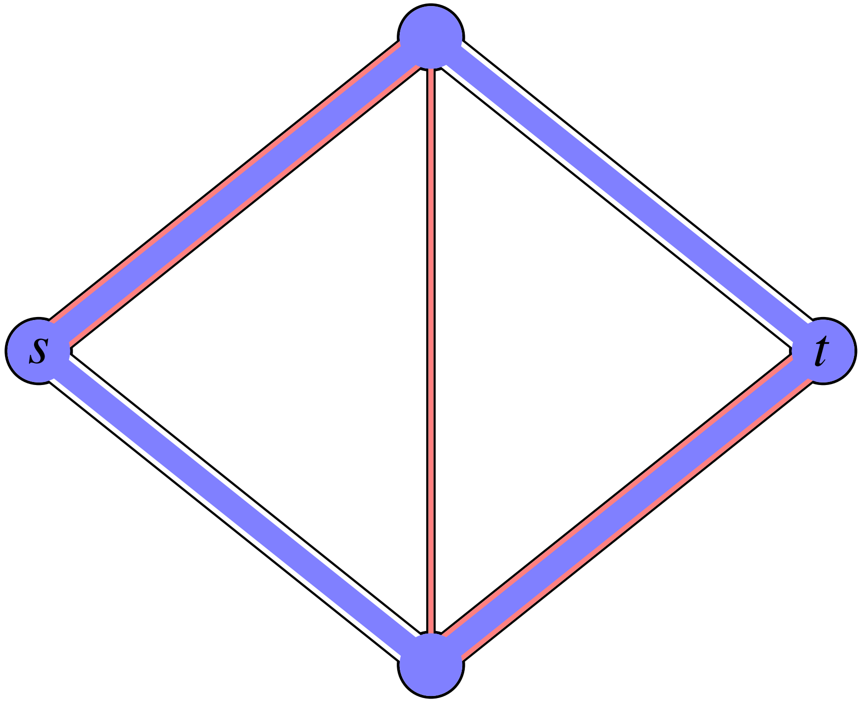
Laufzeit der Ford–Fulkerson–Methode



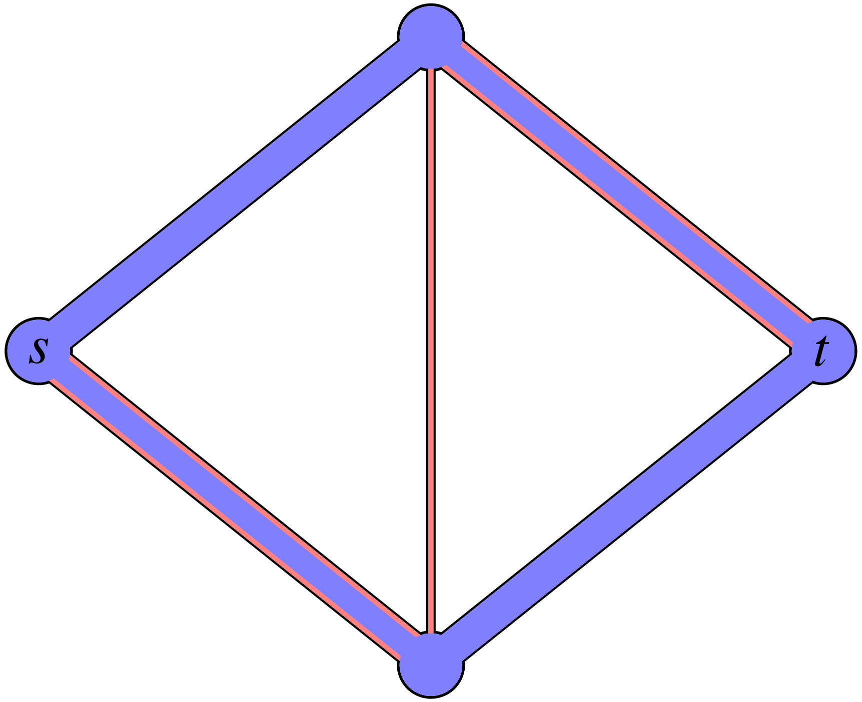
Laufzeit der Ford–Fulkerson–Methode



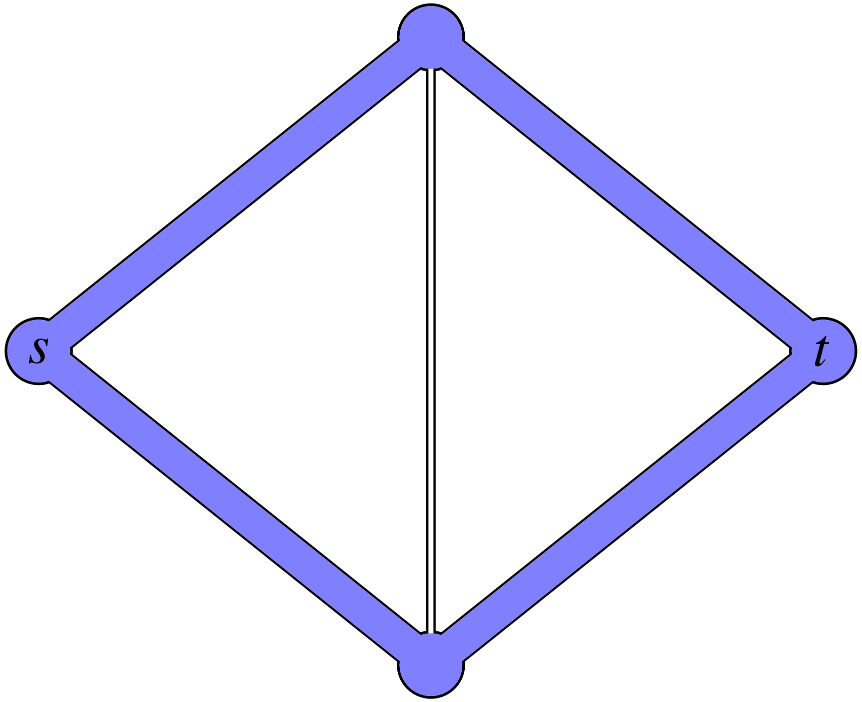
Laufzeit der Ford–Fulkerson–Methode



Laufzeit der Ford–Fulkerson–Methode



Laufzeit der Ford–Fulkerson–Methode



Laufzeit der Ford–Fulkerson–Methode

Ein Flußproblem ist *integral*, wenn alle Kapazitäten ganzzahlig sind.

Theorem.

Die Ford–Fulkerson–Methode benötigt nur $O(f^*)$ Iterationen, um ein integrales Flußproblem zu lösen, falls der Wert eines maximalen Flusses f^* ist.

Beweis.

In jeder Iteration wird der Wert des Flusses um $c_f(p) \geq 1$ erhöht. Er ist anfangs 0 und am Ende f^*.

Korollar.

Bei rationalen Kapazitäten terminiert die Ford–Fulkerson–Methode.

Schnitte in Netzwerken

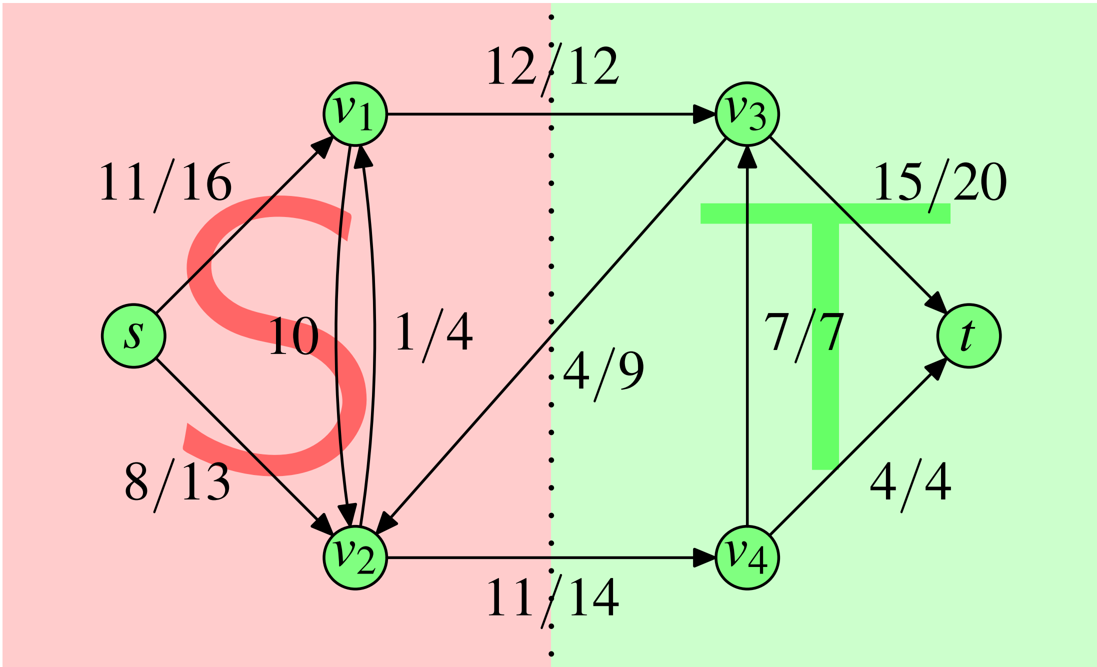
Ein *Schnitt* (S,T) in einem s-t-Netzwerk $G=(V,E)$ ist eine Partition $S \cup T = V$, $S \cap T = \emptyset$ mit $s \in S$ und $t \in T$.

Wenn f ein Fluß in G ist, dann ist $f(S,T)$ der *Fluß über* (S,T).

Die *Kapazität von* (S,T) ist $c(S,T)$.

Ein *minimaler Schnitt* ist ein Schnitt mit minimaler Kapazität.

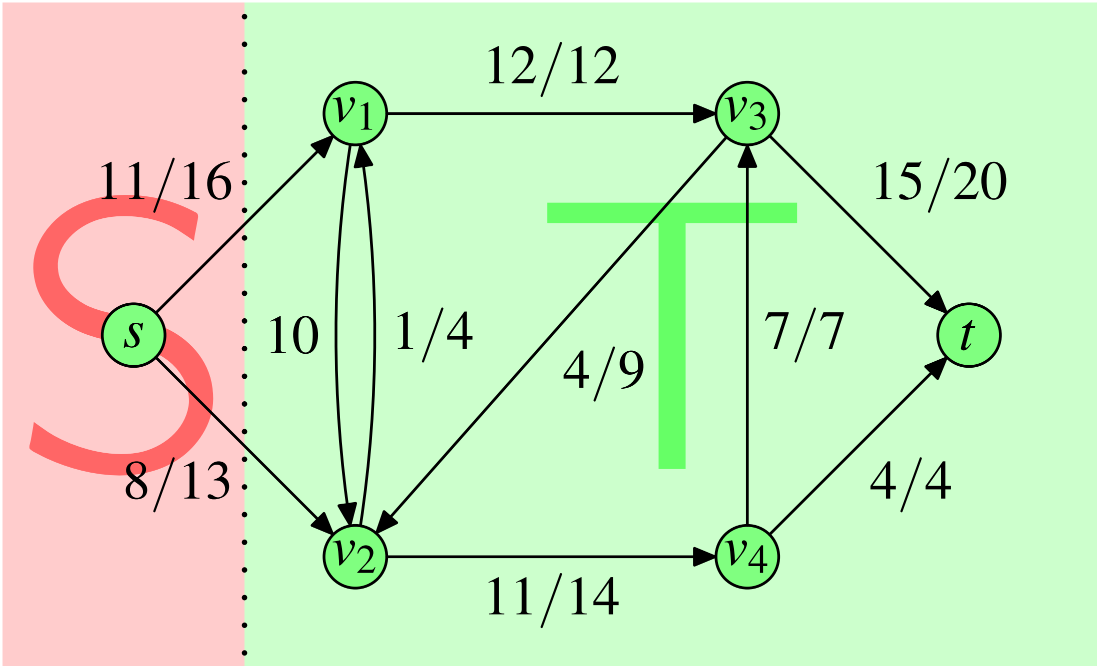
Schnitte in Netzwerken



Der Fluß über (S, T) ist 19.

Die Kapazität von (S, T) ist 26.

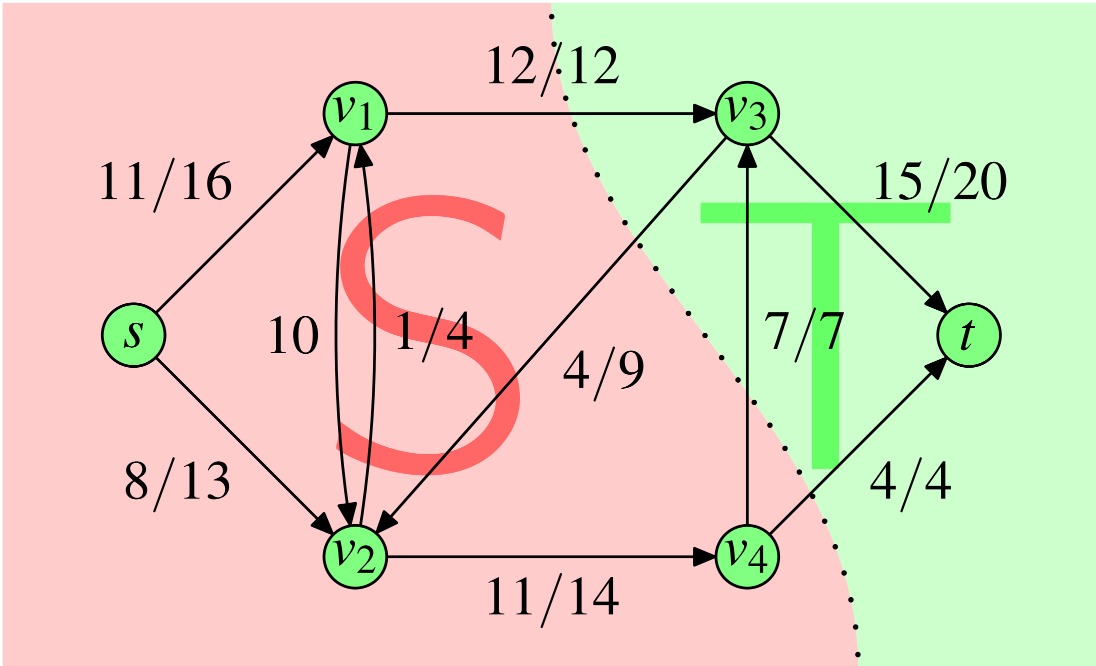
Schnitte in Netzwerken



Der Fluß über (S, T) ist 19.

Die Kapazität von (S, T) ist 29.

Schnitte in Netzwerken



Der Fluß über (S, T) ist 19.

Die Kapazität von (S, T) ist 23.

Fluß über einen Schnitt

Lemma C

Der Fluß über einen Schnitt und der Wert des Flusses sind identisch, d.h. $f(S,T) = |f|$.

Beweis

$$\begin{aligned} f(S,T) &= f(S,V) - f(S,V-T) = f(S,V) - f(S,S) \\ &= f(S,V) = f(s,V) + f(S-s,V) \\ &= f(s,V) = |f| \end{aligned}$$

Spezialfälle

$|f| = f(s, V-s) = f(V-t, t)$

Max-flow Min-cut Theorem

Theorem

Sei f ein Fluß im s-t-Netzwerk $G = (V, E)$.

Dann sind äquivalent:

1. f ist ein maximaler Fluß
2. In G_f gibt es keinen augmentierenden Pfad
3. $|f| = c(S, T)$ für einen Schnitt (S, T)

Folgerungen

Falls die Ford–Fulkerson–Methode terminiert, berechnet sie einen maximalen Fluß.

Die Kapazität eines kleinsten Schnittes gleicht dem Wert eines größten Flusses.

Beweis 1. $\rightarrow$ 2.

Sei f ein maximaler Fluß.

Nehmen wir an, es gebe einen augmentierenden Pfad p.

Dann ist $f + f_p$ ein Fluß in G mit $|f + f_p| > |f|$.

Das ist ein Widerspruch zur Maximalität von f.

Beweis 2. $\rightarrow$ 3.

G_f hat keinen s-t-Pfad.

$S := \{\, v \in V \mid$ es gibt einen s-v-Pfad in $G_f\,\}$

$T := V - S$

Dann ist (S, T) ein Schnitt und es gilt $f(u, v) = c(u, v)$ für alle $u \in S$, $v \in T$.

Nach Lemma C gilt dann $f(S, T) = c(S, T) = |f|$.

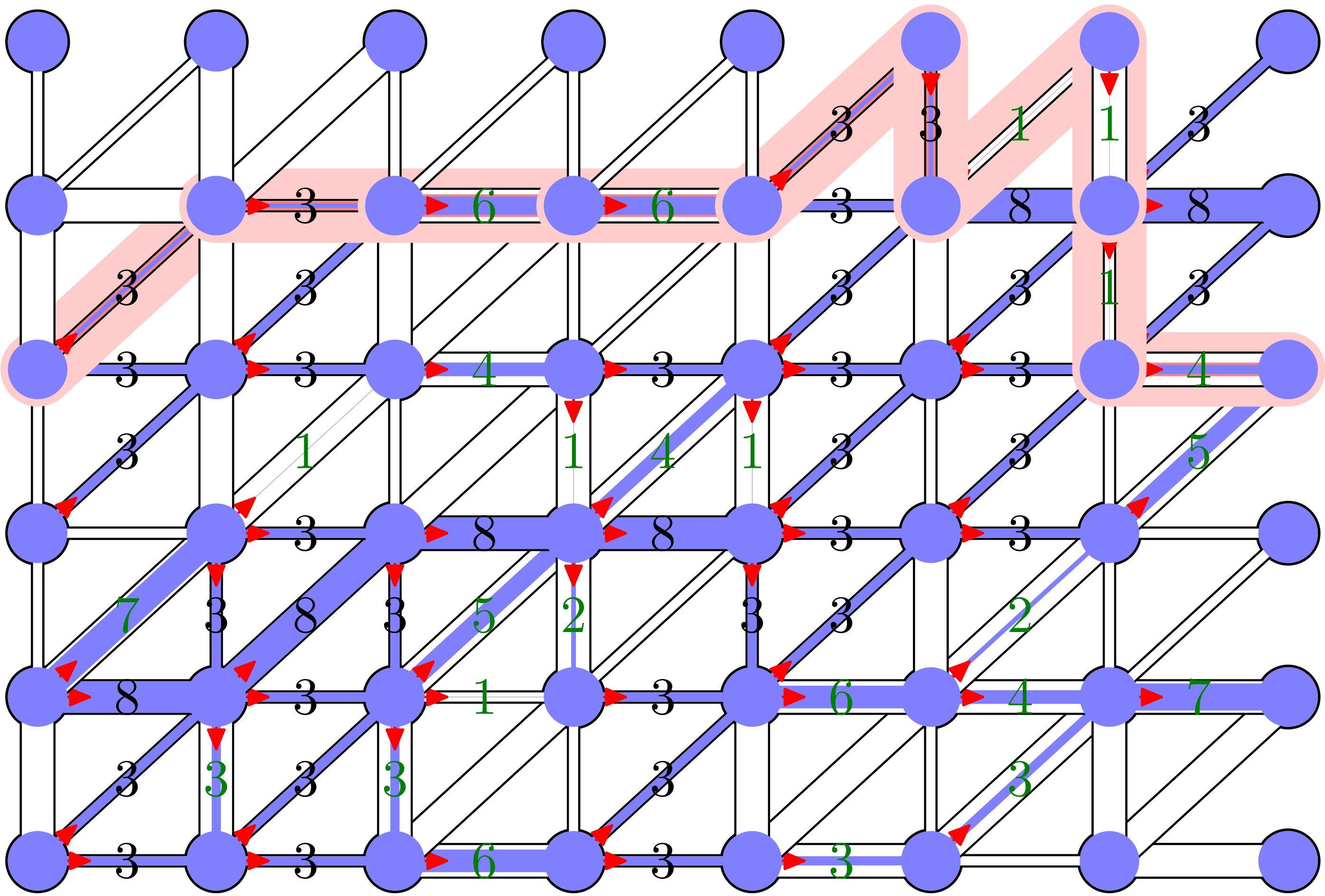
Beweis 3. $\rightarrow$ 1.

Sei f ein beliebiger Fluß.

$$\begin{aligned}
|f| &= f(S,T) \\
&= \sum_{u \in S} \sum_{v \in T} f(u,v) \\
&\leq \sum_{u \in S} \sum_{v \in T} c(u,v) \\
&= c(S,T)
\end{aligned}$$

Der Wert jedes Flusses ist also höchstens $c(S,T)$. Erreicht er sogar $c(S,T)$ ist er folglich maximal.

Wo ist ein minimaler Schnitt?



Wie findet man einen minimalen Schnitt?

1. Einen maximalen Fluß berechnen.
2. Eine Kante (u, v) ist *kritisch*, wenn $c(u, v) = f(u, v)$.
3. S besteht aus Knoten, die von s aus über unkritische Kanten erreicht werden können.
4. T besteht aus allen anderen Knoten.

Es gibt aber bessere, direkte Methoden!