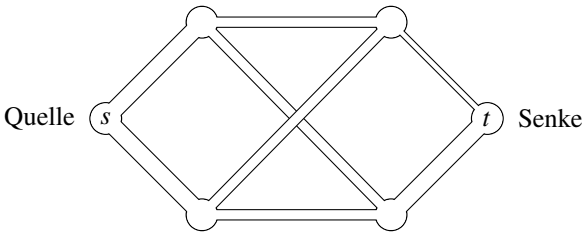
Übersicht

Netzwerkfluß

Gegeben ist ein System von Wasserrohren:

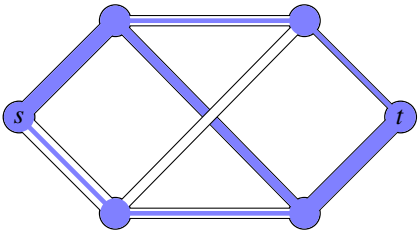


Die Kapazität jedes Rohres ist 3, 5 oder 8 l/s.

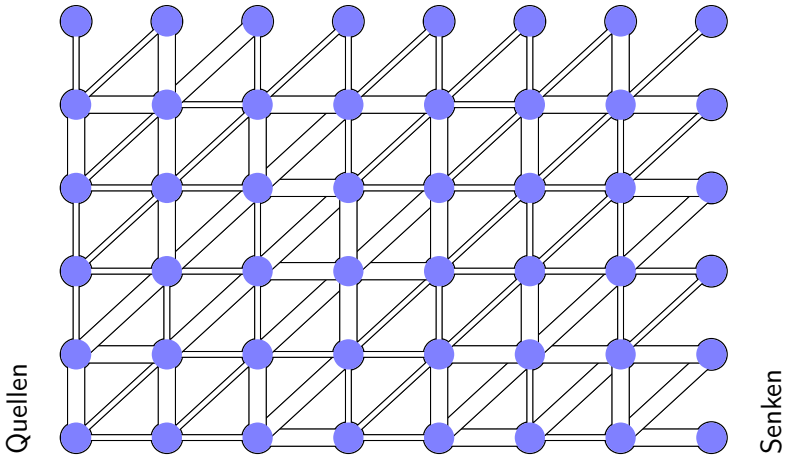
Frage: Wieviel Wasser kann von der Quelle zur Senke fließen?

Netzwerkfluß

Antwort: Maximal 11 l/s sind möglich.

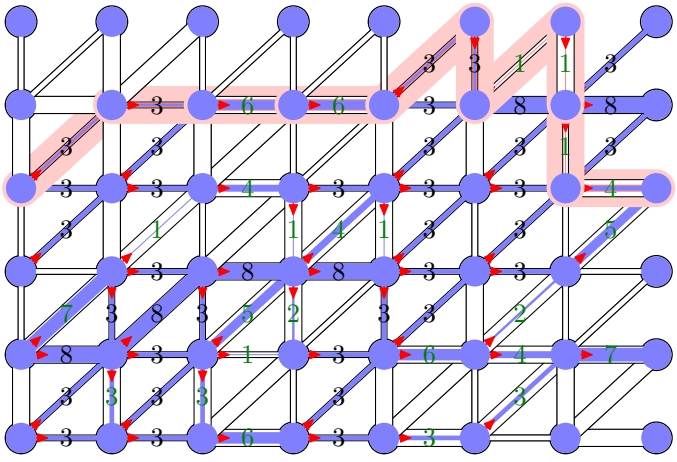


Netzwerkfluß – Aufgabe



Die Kapazitäten sind 3 und 8.

Netzwerkfluß – Lösung



Der maximale Fluß beträgt 30.

s-t-Netzwerke

Definition

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.

s-t-Netzwerke

Definition

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

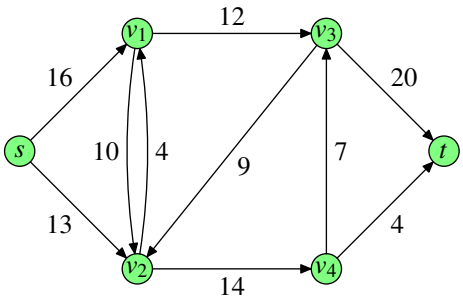
1. jede Kante $(u, v) \in E$ eine Kapazität $c(u, v) \geq 0$ hat,
2. es eine Quelle $s \in V$ und eine Senke $t \in V$ gibt.

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

Definition

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

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

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

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.

Viele Anwendungen

Beispiel: Wieviele Leitungen müssen zerstört sein, damit zwei Computer nicht mehr miteinander kommunizieren können.

Weiteres Beispiel: ISS-Problem

Maximale Flüsse

Das Problem des maximalen Flusses:

Gegeben: Ein s-t-Netzwerk.

Gesucht: Ein Fluß mit maximalem Wert.

Viele Anwendungen

Beispiel: Wieviele Leitungen müssen zerstört sein, damit zwei Computer nicht mehr miteinander kommunizieren können.

Weiteres Beispiel: ISS-Problem

Maximale Flüsse

Das Problem des maximalen Flusses:

Gegeben: Ein s-t-Netzwerk.

Gesucht: Ein Fluß mit maximalem Wert.

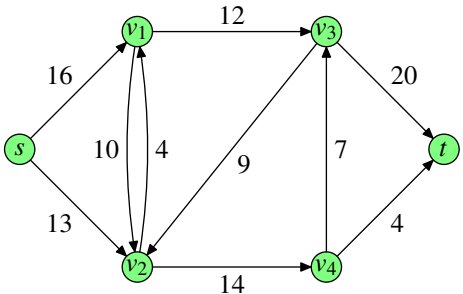
Viele Anwendungen

Beispiel: Wieviele Leitungen müssen zerstört sein, damit zwei Computer nicht mehr miteinander kommunizieren können.

Weiteres Beispiel: ISS-Problem

- Weltraumtouristen machen Angebote
- Sie benötigen spezielle Ausrüstung
- Ausrüstung kann mehrfach benutzt werden
- Wer soll mitgenommen werden?

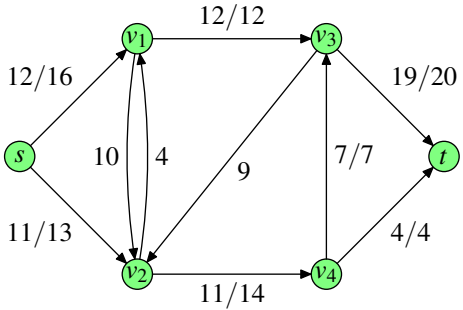
Was ist der maximale Fluß?



Der maximale Fluß ist 23.

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

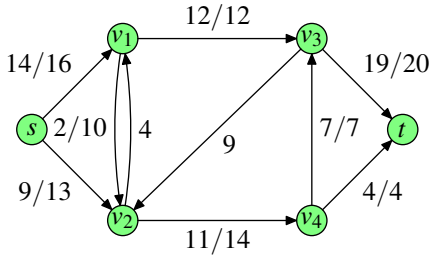
Was ist der maximale Fluß?

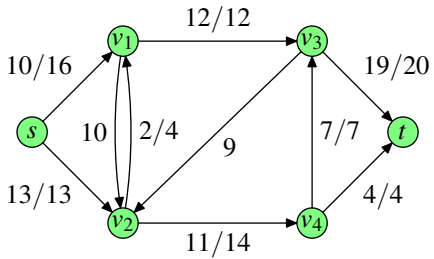


Der maximale Fluß ist 23.

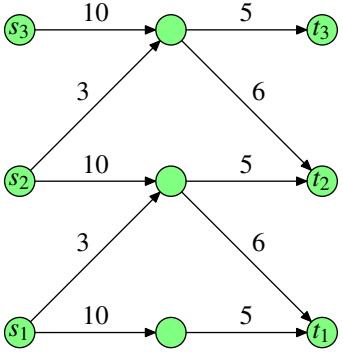
Die Kanten sind mit $c(u, v)$ beschriftet oder mit $f(u, v)/c(u, v)$, falls $f(u, v) > 0$.

Andere optimale Lösungen



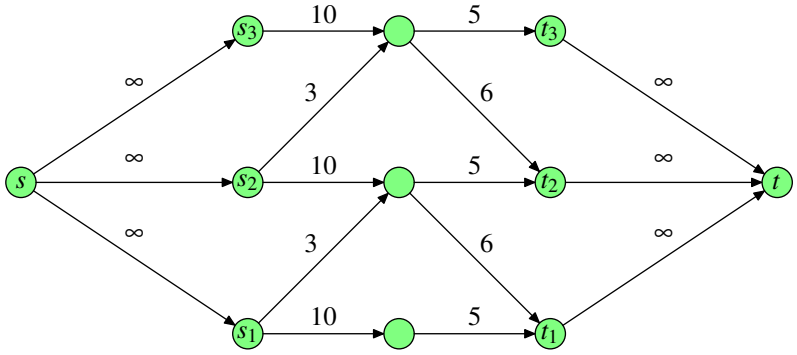


Mehrfache Quellen oder Senken



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

$$\max\{\, |f| \mid f \text{ ist ein } s\text{-}t\text{-Fluß in } G \,\}?$$

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:

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

(Satz von Weierstrass)

Existenz des maximalen Flusses

Existiert stets der maximale Fluß

$$\max\{\, |f| \mid f \text{ ist ein } s\text{-}t\text{-Fluß in } G \,\}?$$

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:

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

(Satz von Weierstrass)

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 $f(X, X) = 0$:

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

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:

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

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:

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

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:

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

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:

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

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:

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

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

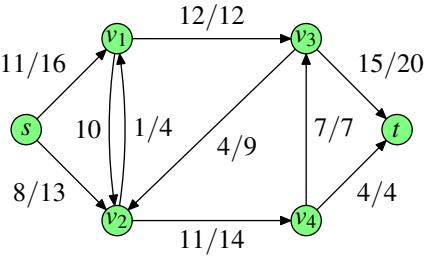
$$E_f = \{ (u, v) \in V \times V \mid c_f(u, v) > 0 \},$$

wobei

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

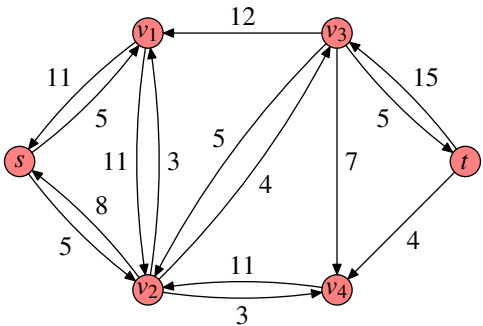
c_f ist die Restkapazität.
Das s-t-Netzwerk G_f hat die Kapazitäten c_f.

Beispiel



s-t-Netzwerk G mit Fluß f

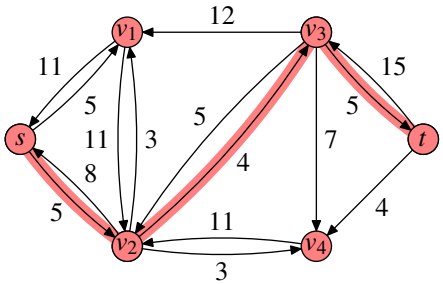
Residualnetzwerk G_f



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

Algorithmus

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

Algorithmus

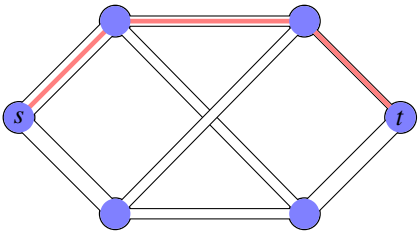
Initialisiere Fluß f zu 0
while es gibt einen augmentierenden Pfad p
  do augmentiere f entlang p
return f

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

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

f_p ist ein Fluß in G_f

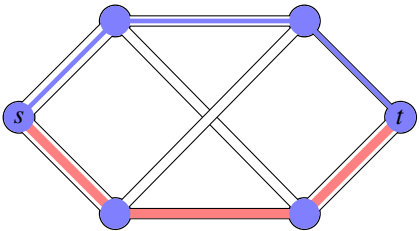
Die Ford–Fulkerson–Methode



Anfangs ist der Fluß 0.

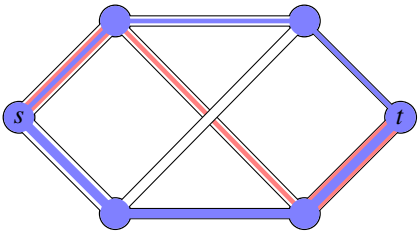
Der augmentierende Pfad ist rot eingezeichnet.

Die Ford–Fulkerson–Methode



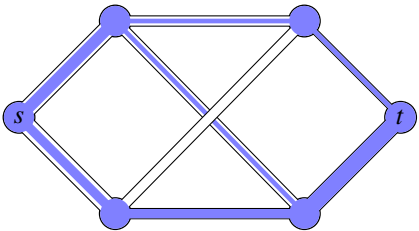
Der augmentierende Pfad hat Kapazität 5.

Die Ford–Fulkerson–Methode



Der augmentierende Pfad hat Kapazität 3.

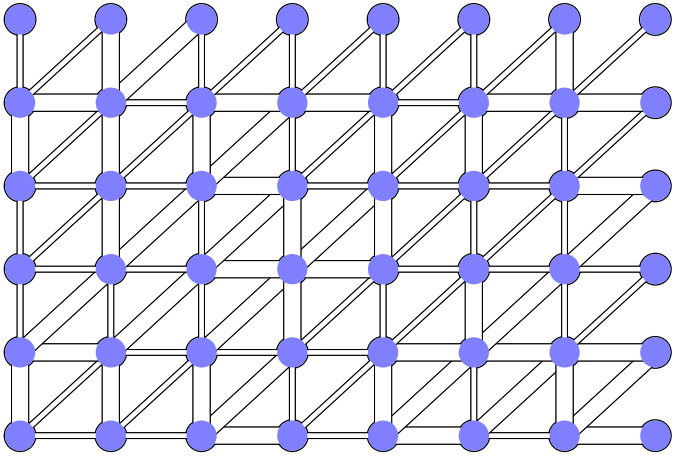
Die Ford–Fulkerson–Methode



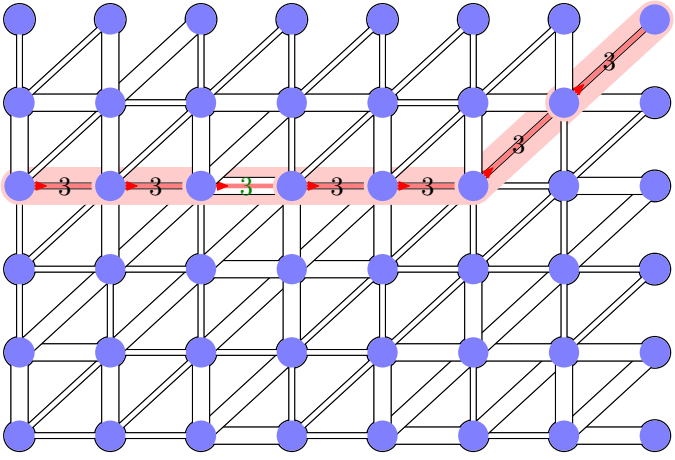
Jetzt gibt es keinen augmentierenden Pfad mehr.

Der Fluß ist maximal.

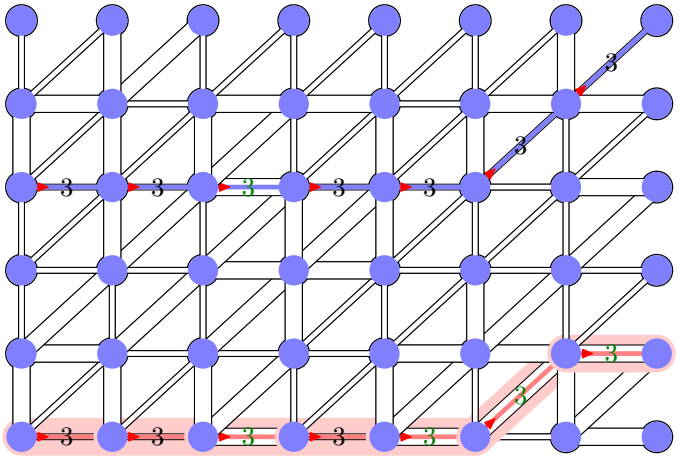
Beispiel



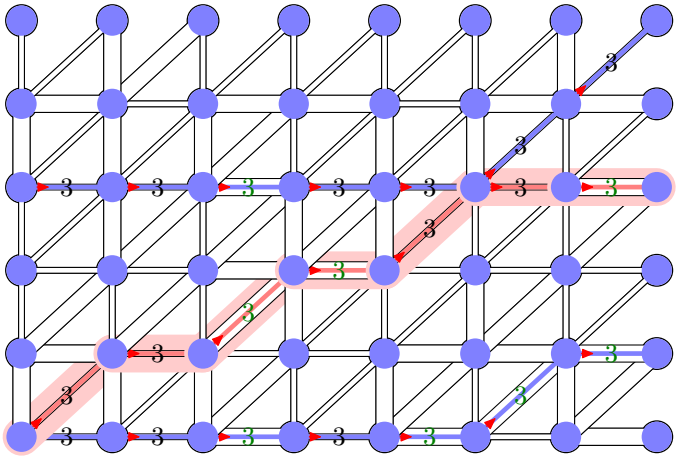
Beispiel



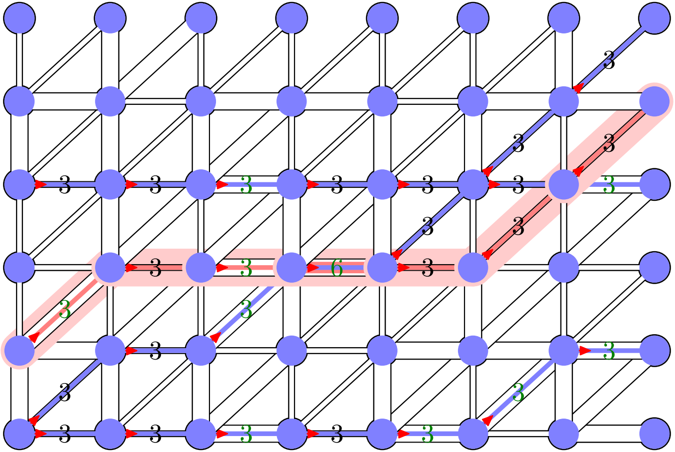
Beispiel



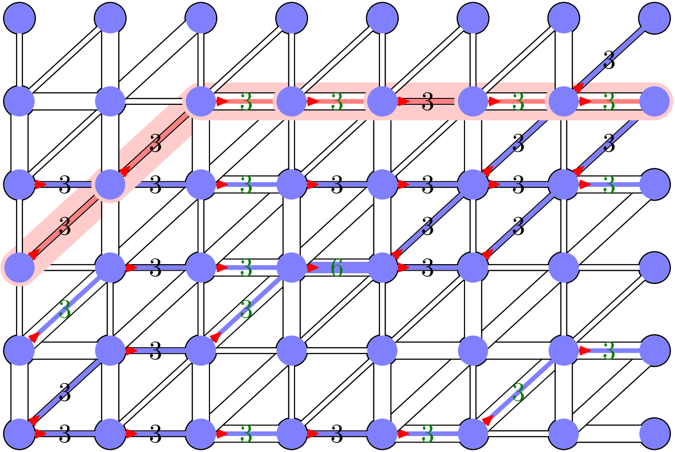
Beispiel


3
3
3

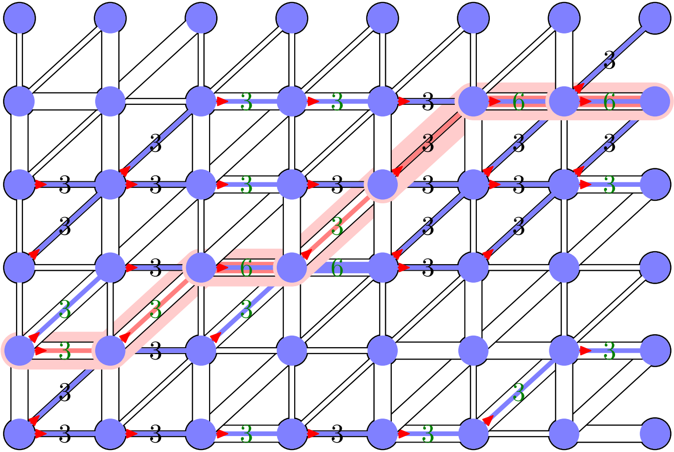
Beispiel



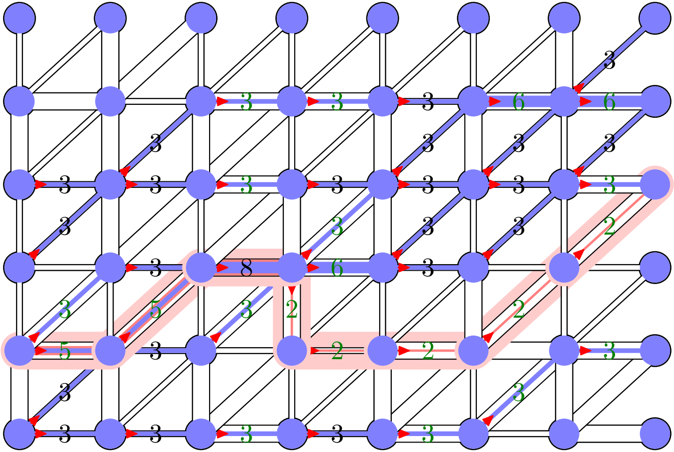
Beispiel



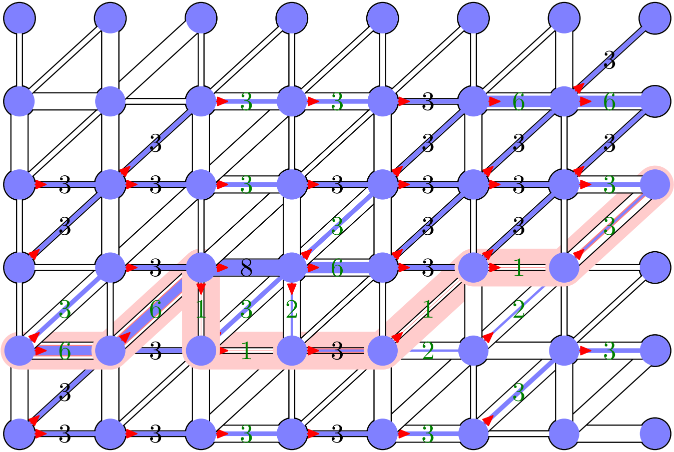
Beispiel



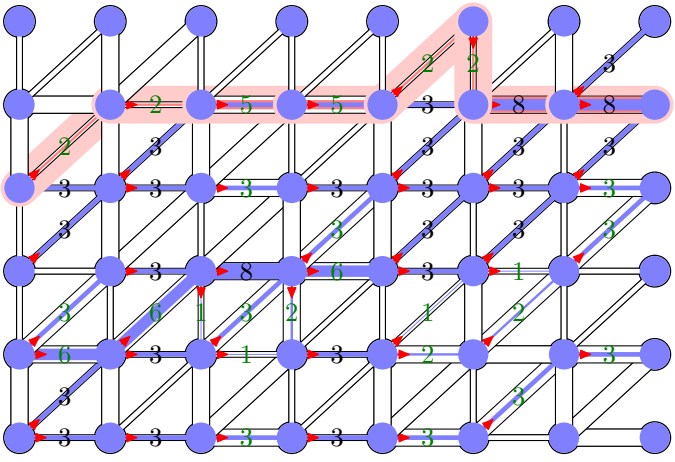
Beispiel



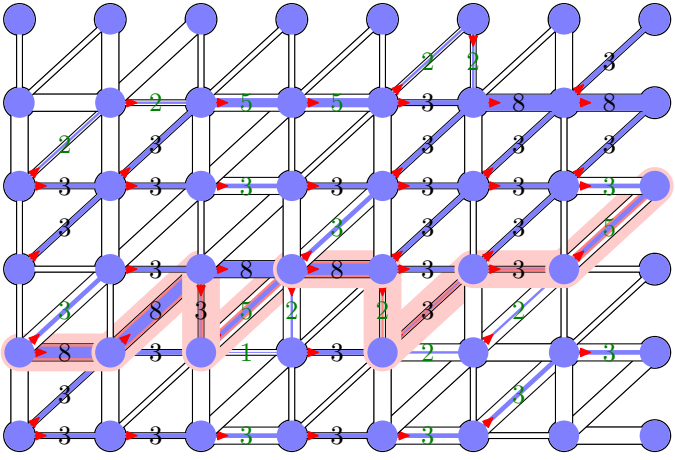
Beispiel



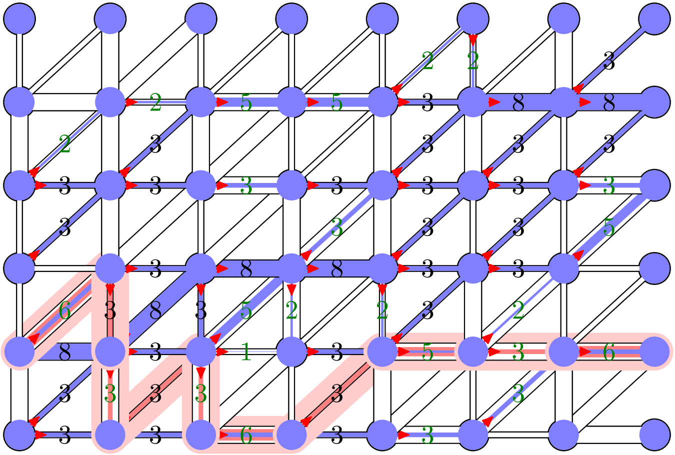
Beispiel



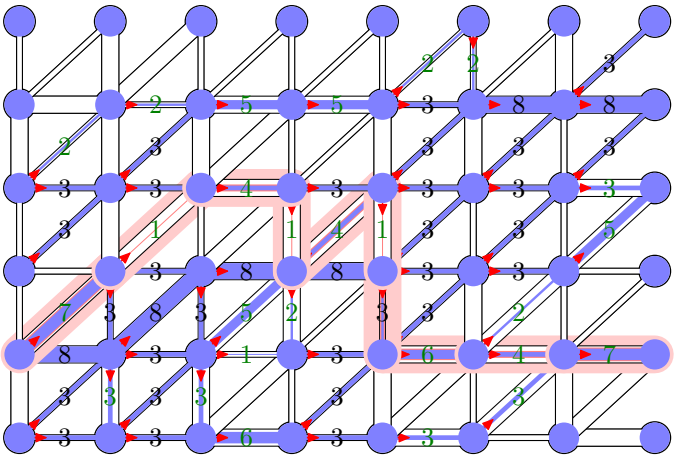
Beispiel



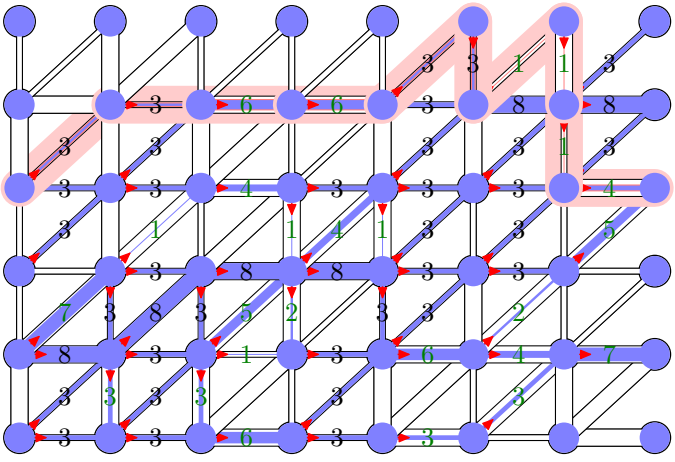
Beispiel


2 2 3
2 5 5 3 8 8
2 3 3 3 3
3 3 3 3 3 3 3
3 3 3 3 5
3 8 8 3 3
6 3 8 3 5 2 2 3 2
8 3 1 3 5 3 6
3 3 3 3 3 3
3 3 6 3 3

Beispiel


2 2 3
2 5 5 3 8 8
2 3 3 3 3
3 3 4 3 3 3
3 1 1 4 1 3 3 5
3 8 8 3 3
7 3 8 3 5 2 3 3 2
8 3 1 3 6 4 7
3 3 3 3 3 3
3 3 6 3 3

Beispiel


3 3 1 1 3
3 6 6 3 8 8
3 3 3 3 3 3
3 3 4 3 3 4
3 1 1 4 1 3 3 5
3 3 8 8 3 3
7 3 8 3 5 2 3 3 2
8 3 1 3 6 4 7
3 3 3 3 3 3
3 3 6 3 3

Korrektheit

Lemma B

Sei $G = (V, E)$ 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. □

Korrektheit

Lemma B

Sei $G = (V, E)$ 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 (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 (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 (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 (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.

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.

Beweis (Zulässigkeit)

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

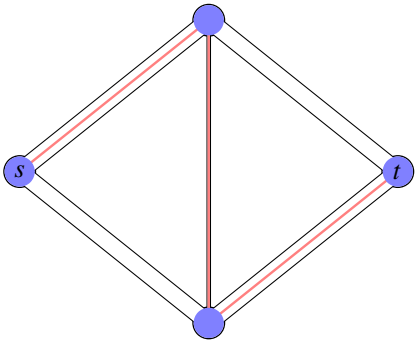
Der Beweis verwendet, daß f' ein Fluß in G_f ist, aber nicht, daß f ein Fluß in G ist.

Beweis (Zulässigkeit)

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

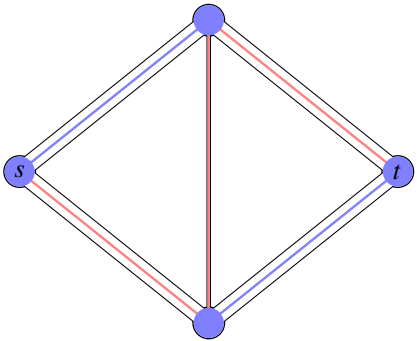
Der Beweis verwendet, daß f' ein Fluß in G_f ist, aber nicht, daß f ein Fluß in G ist.

Laufzeit der Ford–Fulkerson–Methode



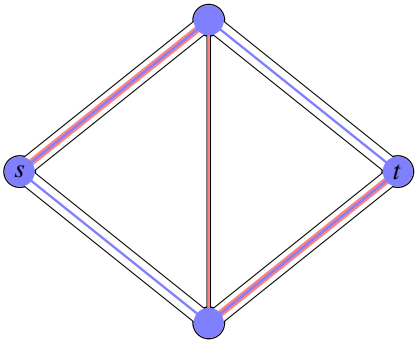
Die Laufzeit kann beliebig schlecht sein.

Laufzeit der Ford–Fulkerson–Methode



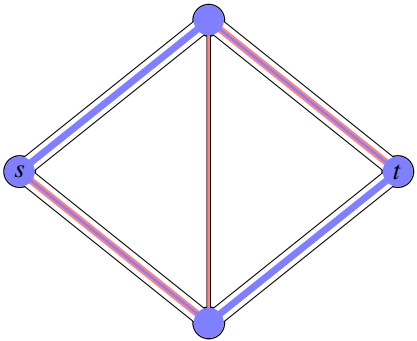
Die Laufzeit kann beliebig schlecht sein.

Laufzeit der Ford–Fulkerson–Methode



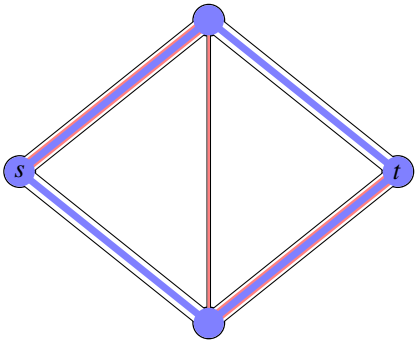
Die Laufzeit kann beliebig schlecht sein.

Laufzeit der Ford–Fulkerson–Methode



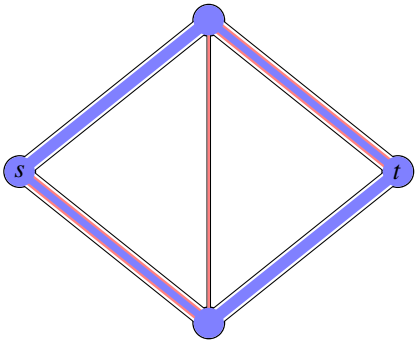
Die Laufzeit kann beliebig schlecht sein.

Laufzeit der Ford–Fulkerson–Methode



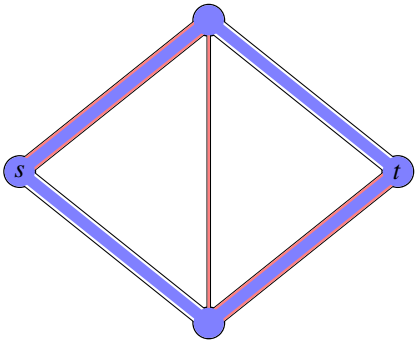
Die Laufzeit kann beliebig schlecht sein.

Laufzeit der Ford–Fulkerson–Methode



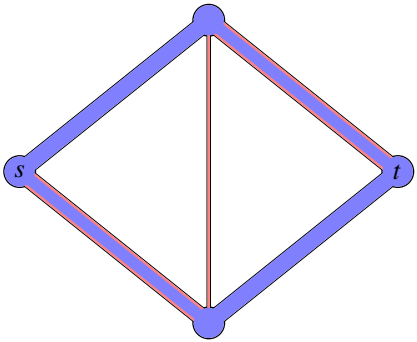
Die Laufzeit kann beliebig schlecht sein.

Laufzeit der Ford–Fulkerson–Methode



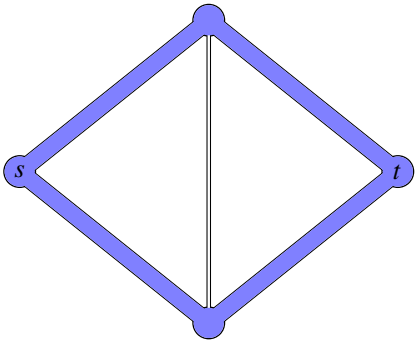
Die Laufzeit kann beliebig schlecht sein.

Laufzeit der Ford–Fulkerson–Methode



Die Laufzeit kann beliebig schlecht sein.

Laufzeit der Ford–Fulkerson–Methode



Die Laufzeit kann beliebig schlecht sein.

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

Definition

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

Definition

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

Definition

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

Definition

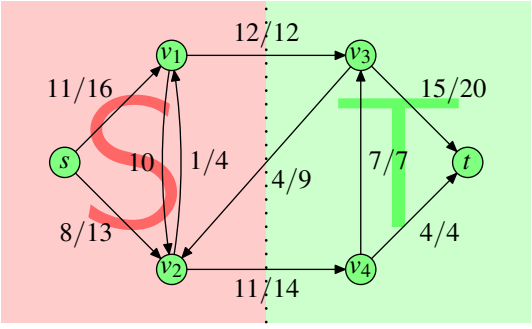
Ein Schnitt (S, T) in einem s-t-Netzwerk $G = (V, E)$ ist eine Partition $S \cup T = V$, $S \cap T = \emptyset$ mit $s \in S$ und $t \in T$.

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.

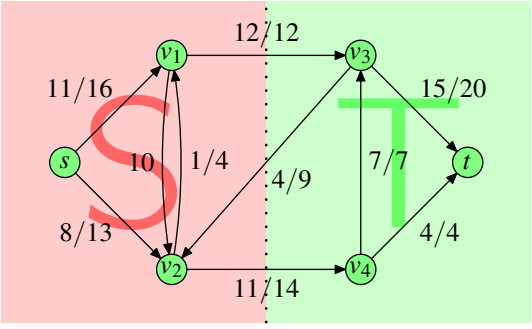
Schnitte in Netzwerken



Der Fluß über (S, T) ist 19.

Die Kapazität von (S, T) ist 26.

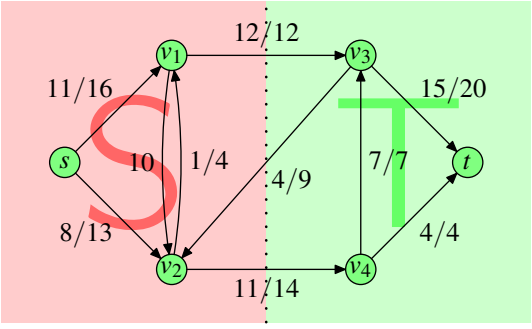
Schnitte in Netzwerken



Der Fluß über (S, T) ist 19.

Die Kapazität von (S, T) ist 26.

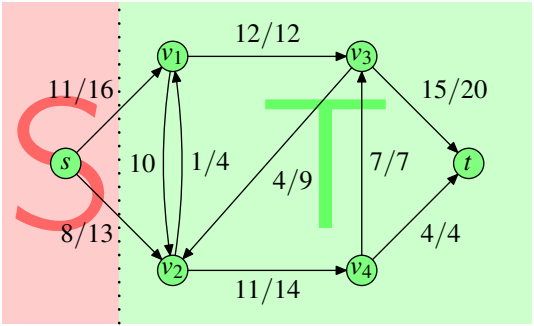
Schnitte in Netzwerken



Der Fluß über (S, T) ist 19.

Die Kapazität von (S, T) ist 26.

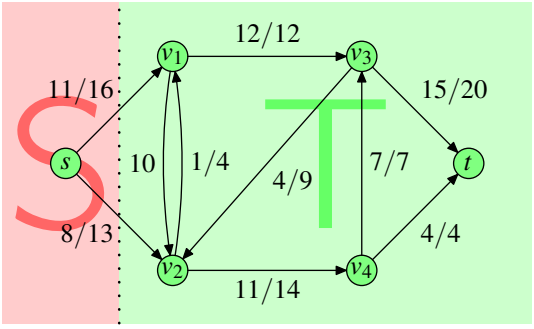
Schnitte in Netzwerken



Der Fluß über (S, T) ist 19.

Die Kapazität von (S, T) ist 29.

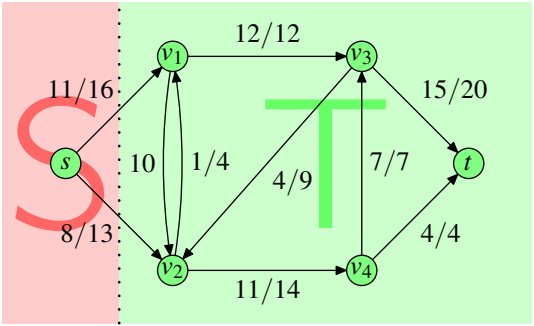
Schnitte in Netzwerken



Der Fluß über (S, T) ist 19.

Die Kapazität von (S, T) ist 29.

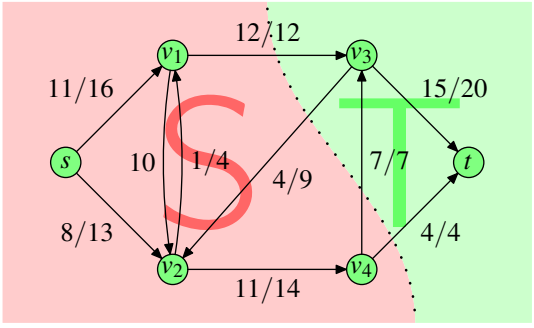
Schnitte in Netzwerken



Der Fluß über (S, T) ist 19.

Die Kapazität von (S, T) ist 29.

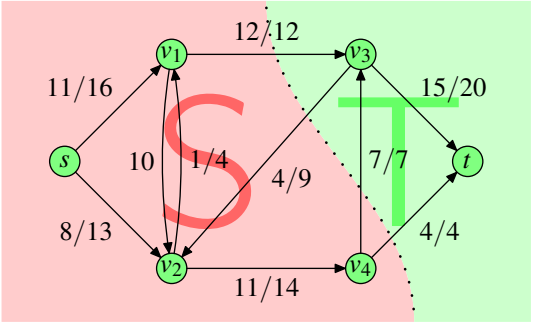
Schnitte in Netzwerken



Der Fluß über (S, T) ist 19.

Die Kapazität von (S, T) ist 23.

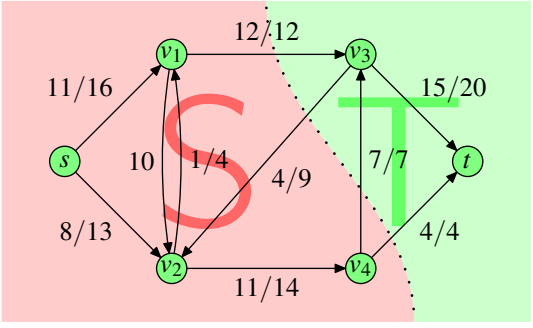
Schnitte in Netzwerken



Der Fluß über (S, T) ist 19.

Die Kapazität von (S, T) ist 23.

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:

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

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:

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

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

1. Falls die Ford–Fulkerson–Methode terminiert, berechnet sie einen maximalen Fluß.
2. Die Kapazität eines kleinsten Schnittes gleicht dem Wert eines größten Flusses.

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

1. Falls die Ford–Fulkerson–Methode terminiert, berechnet sie einen maximalen Fluß.
2. 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. $\square$

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 $f(S, T) = c(S, T) = |f|$. $\square$

Beweis.

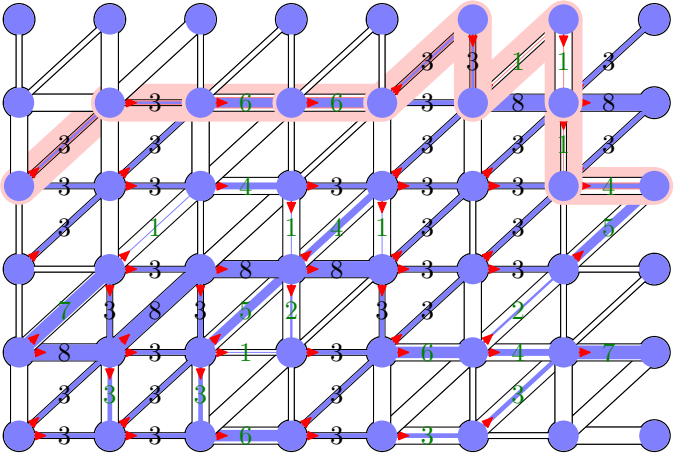
3. $\rightarrow$ 1.

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. $\square$

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!

Ganzzahlige Flüsse

Theorem

Wenn alle Kapazitäten ganzzahlig sind, dann findet die Ford–Fulkerson–Methode einen maximalen Fluß f, so daß alle $f(u, v)$ ganzzahlig sind.

Beweis.

Induktion zeigt, daß die Kapazität eines augmentierenden Pfads ganzzahlig ist und $f(u, v)$ stets ganzzahlig bleiben. □

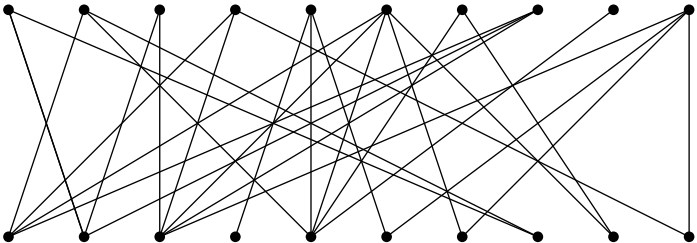
Bipartites Matching

Gegeben: Ein bipartiter, ungerichteter Graph (V_1, V_2, E).

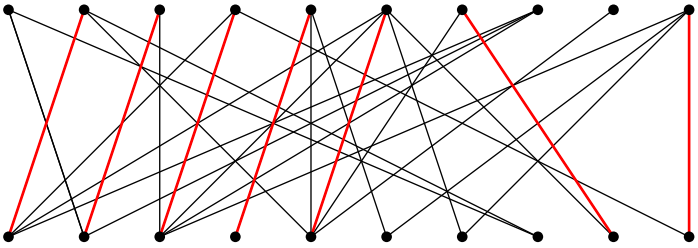
Gesucht:
Ein Matching (Paarung) maximaler Kardinalität.

Ein Matching ist eine Menge paarweise nicht inzidenter Kanten, also $M \subseteq E$ mit $m_1, m_2 \in M,\ m_1 \neq m_2 \Rightarrow m_1 \cap m_2 = \emptyset$.

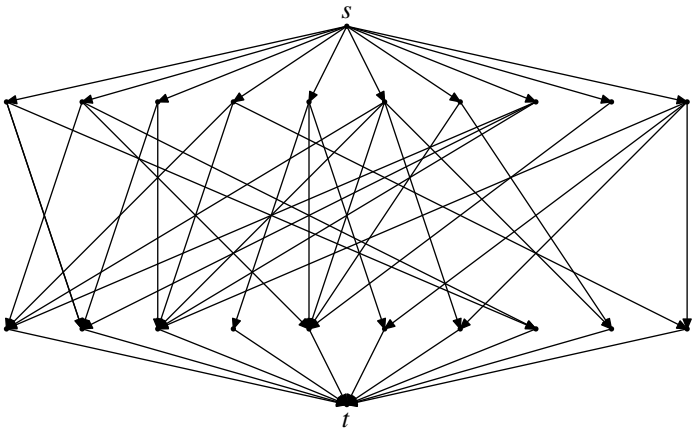
Beispiel



Beispiel



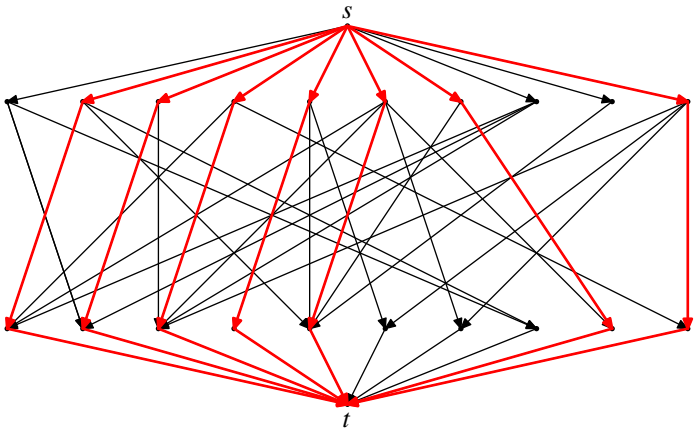
Lösung als Flußproblem



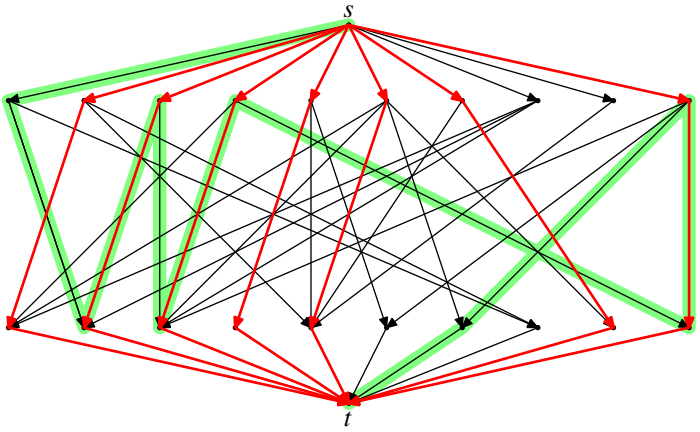
Alle Kapazitäten sind 1.

Maximaler ganzzahliger Fluß entspricht einem Matching maximaler Kardinalität.

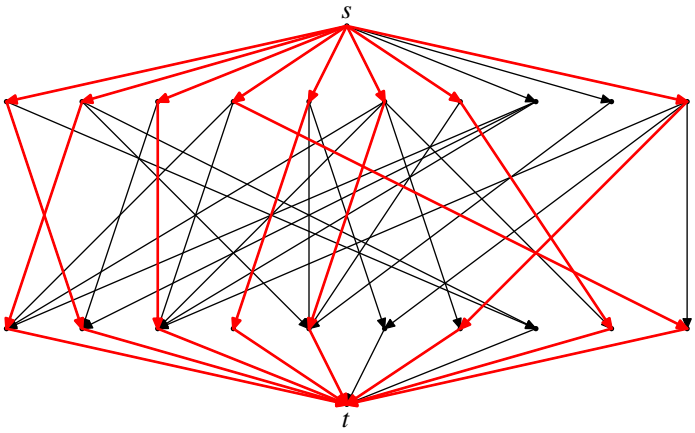
Gibt es einen augmentierenden Pfad?



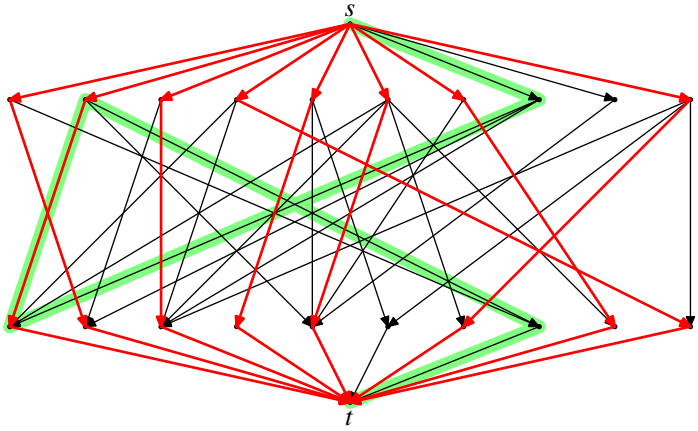
Gibt es einen augmentierenden Pfad? – Ja



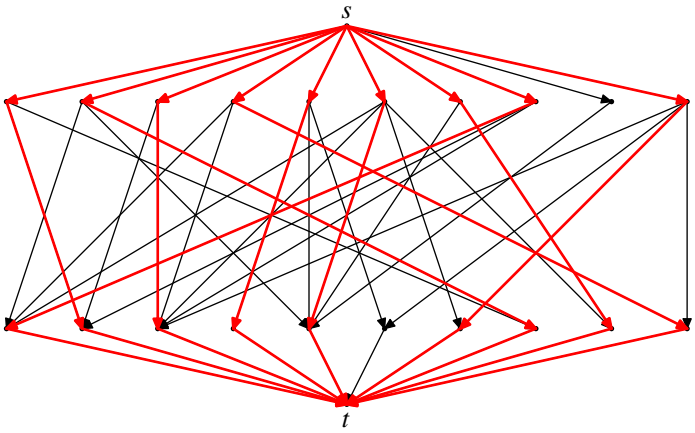
$\Rightarrow$ Neues Matching



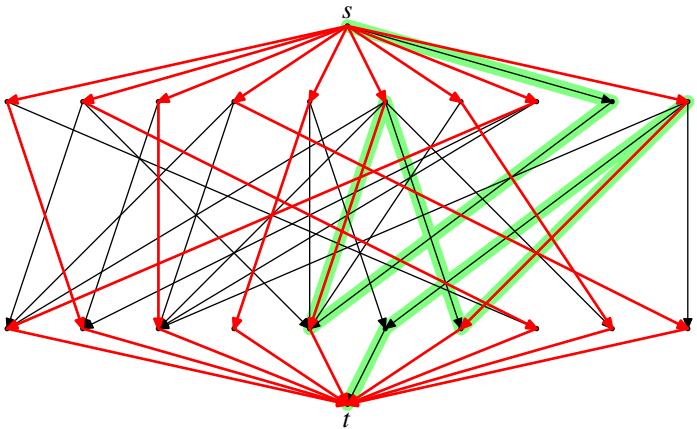
Es gibt wieder einen augmentierenden Pfad



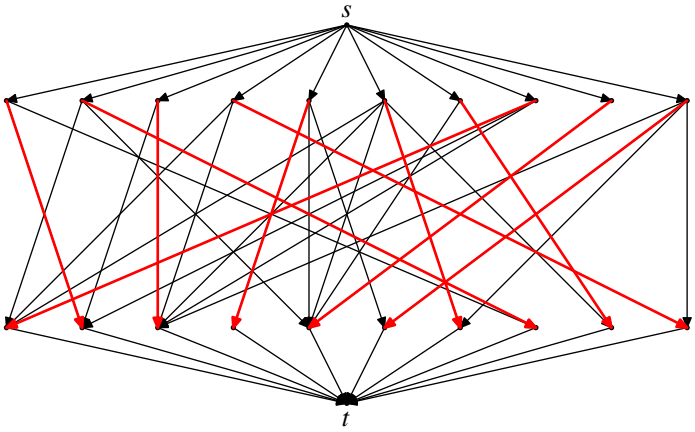
$\Rightarrow$ Neues Matching



Es gibt wieder einen augmentierenden Pfad



Ergebnis: Perfektes Matching



Laufzeit

Finden eines Matchings maximaler Kardinalität dauert nur $O(|E| \cdot \min\{|V_1|, |V_2|\})$ mit der Ford–Fulkerson–Methode.

- Der Fluß ist höchstens $f^* = \min\{|V_1|, |V_2|\}$.
- Finden eines Pfads dauert $O(|E|)$.

Der Edmonds–Karp–Algorithmus

Die Ford–Fulkerson–Methode kann sehr langsam sein, auch wenn das Netzwerk klein ist.

Der Edmonds–Karp–Algorithmus ist polynomiell in der Größe des Netzwerks.

Algorithmus

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

Unterschied: Es wird ein kürzester Pfad gewählt

Der Edmonds–Karp–Algorithmus

Algorithmus

for each edge $(u, v) \in E$ **do**
  $f(u, v) \leftarrow 0$
  $f(v, u) \leftarrow 0$
while there exists a path from s to t in G_f **do**
  $p \leftarrow$ a shortest path from s to t in G_f
  $c_f(p) \leftarrow \min\{\, c_f(u, v) \mid (u, v) \text{ is in } p \,\}$
  for each edge (u, v) in p **do**
    $f(u, v) \leftarrow f(u, v) + c_f(p)$
    $f(v, u) \leftarrow -f(u, v)$
return f