FernUniversität in Hagen
Fakultät für Mathematik und Informatik

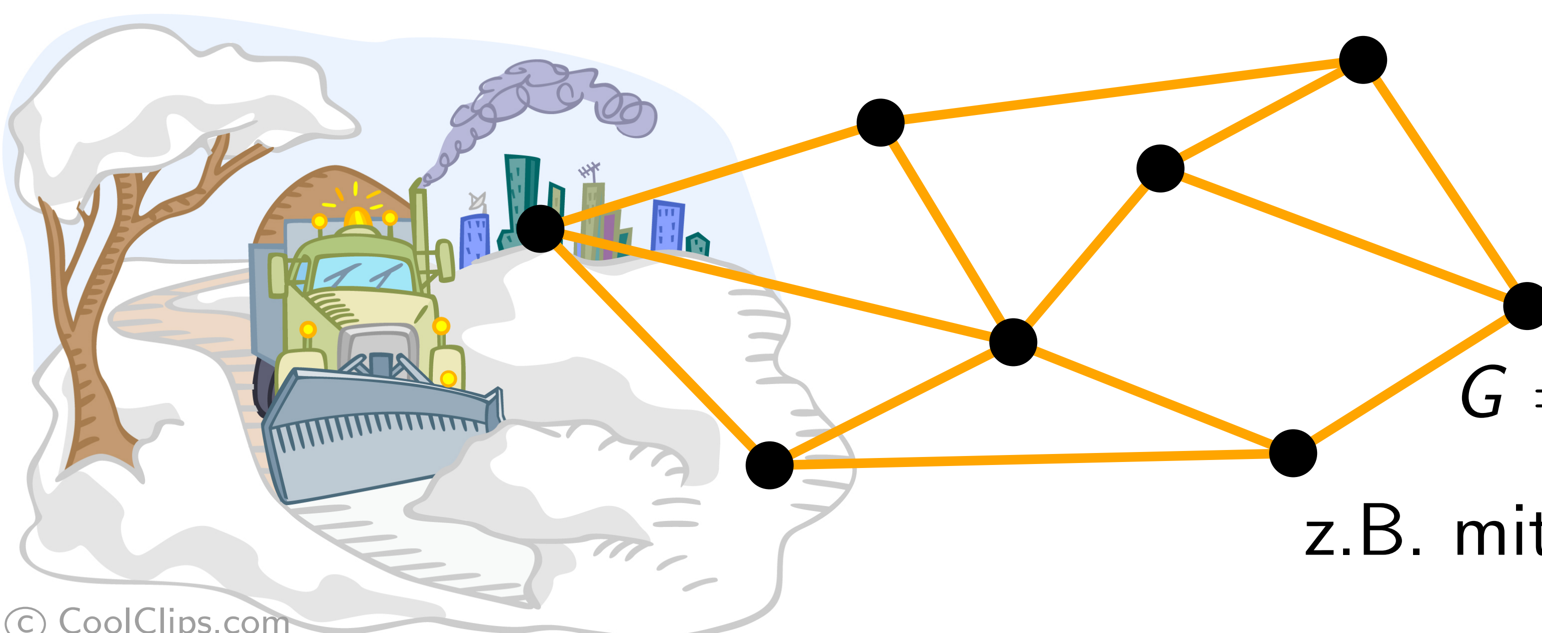
theoretische informatik

Das Problem des minimalen Spannbaums

Dr. Philipp Kindermann
LG Theoretische Informatik
FernUniversität in Hagen

Motivation

Gegeben: Zusammenhängendes Straßennetz $G = (V, E; w^{\star})$, das eine Menge V von n Städten verbindet



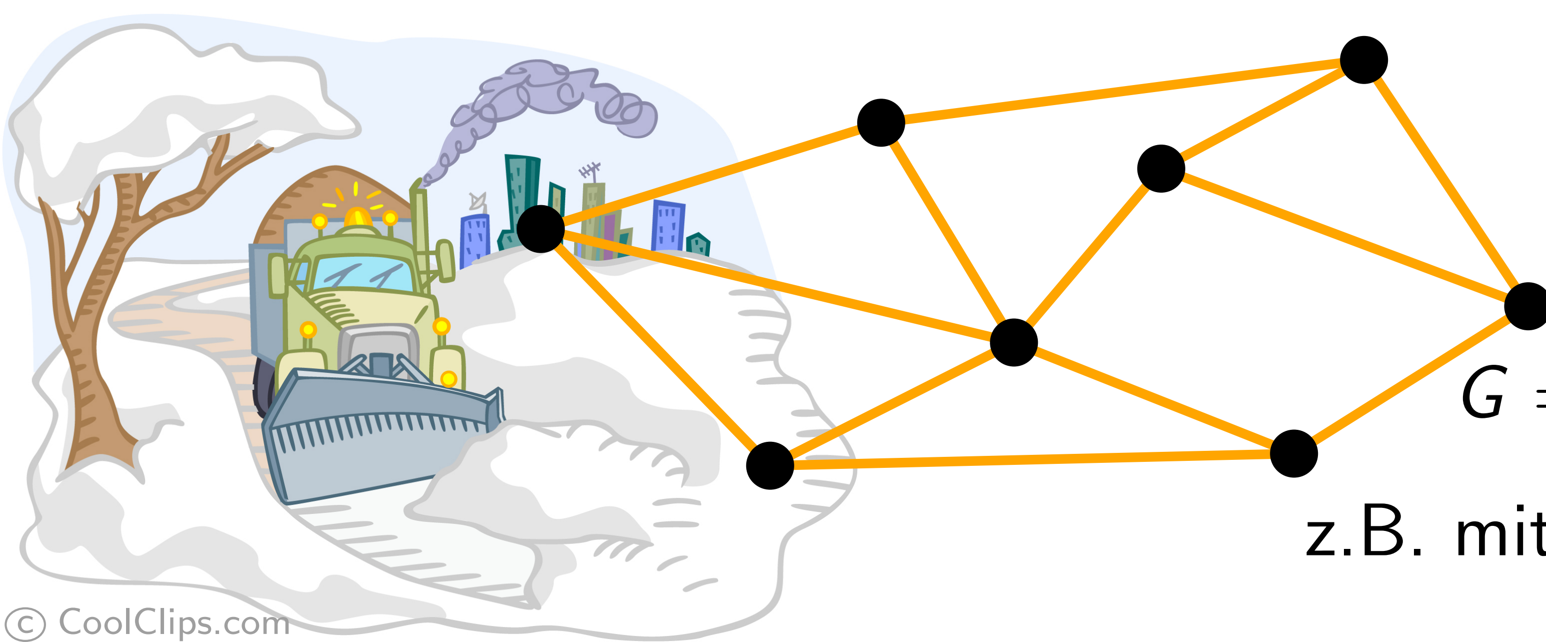
$G = (V, E; w)$

z.B. mit $w \equiv \begin{cases} \text{euklid.} \\ \text{Abstände} \end{cases}$

Motivation

*) Kantengewichte $w \colon E \to \mathbb{R}_{>0}$

Gegeben: Zusammenhängendes Straßennetz $G = (V, E; w^{\star})$, das eine Menge V von n Städten verbindet



$G = (V, E; w)$

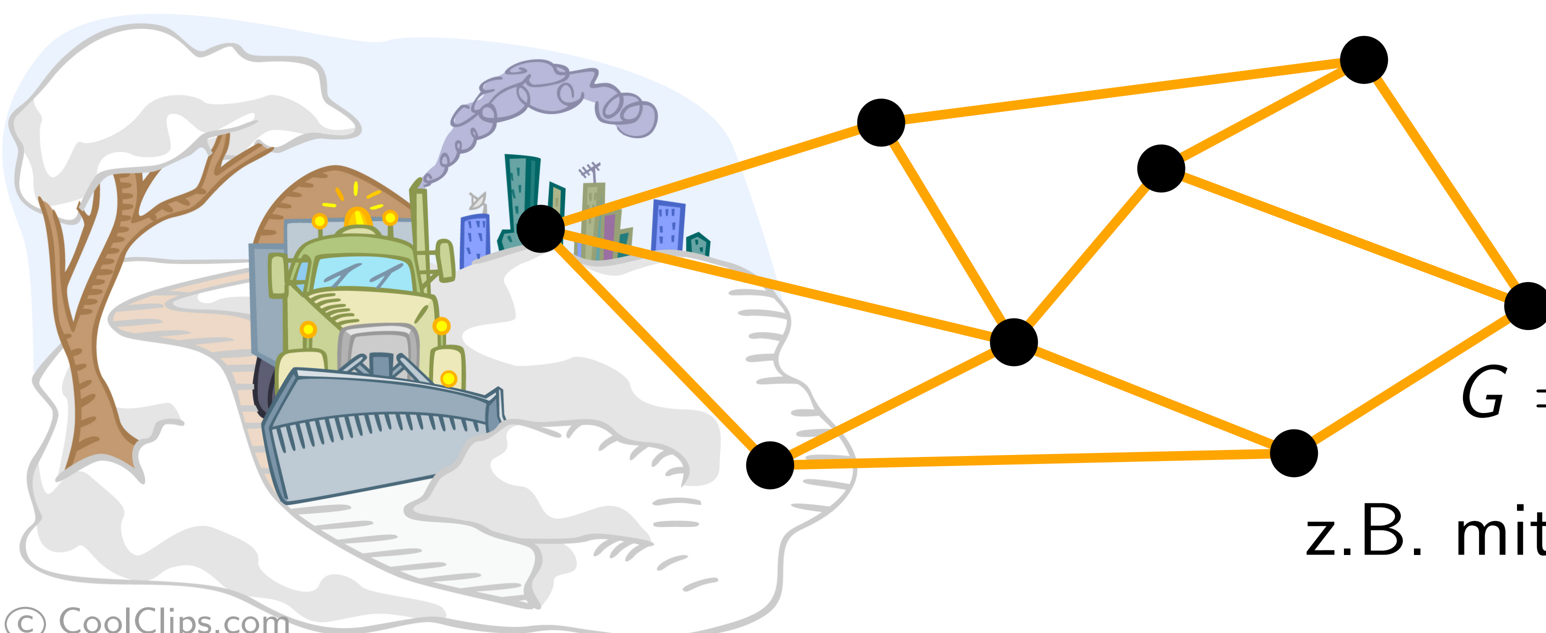
z.B. mit $w \equiv \begin{cases} \text{euklid.} \\ \text{Abstände} \end{cases}$

Motivation

$^{\star})$ Kantengewichte $w\colon E \to \mathbb{R}_{>0}$

Gegeben: Zusammenhängendes Straßennetz $G = (V, E; w^{\star})$, das eine Menge V von n Städten verbindet

Gesucht: Teilnetz $T = (V, E')$ mit $E' \subseteq E$, so dass



$G = (V, E; w)$

z.B. mit $w \equiv \begin{cases} \text{euklid.} \\ \text{Abstände} \end{cases}$

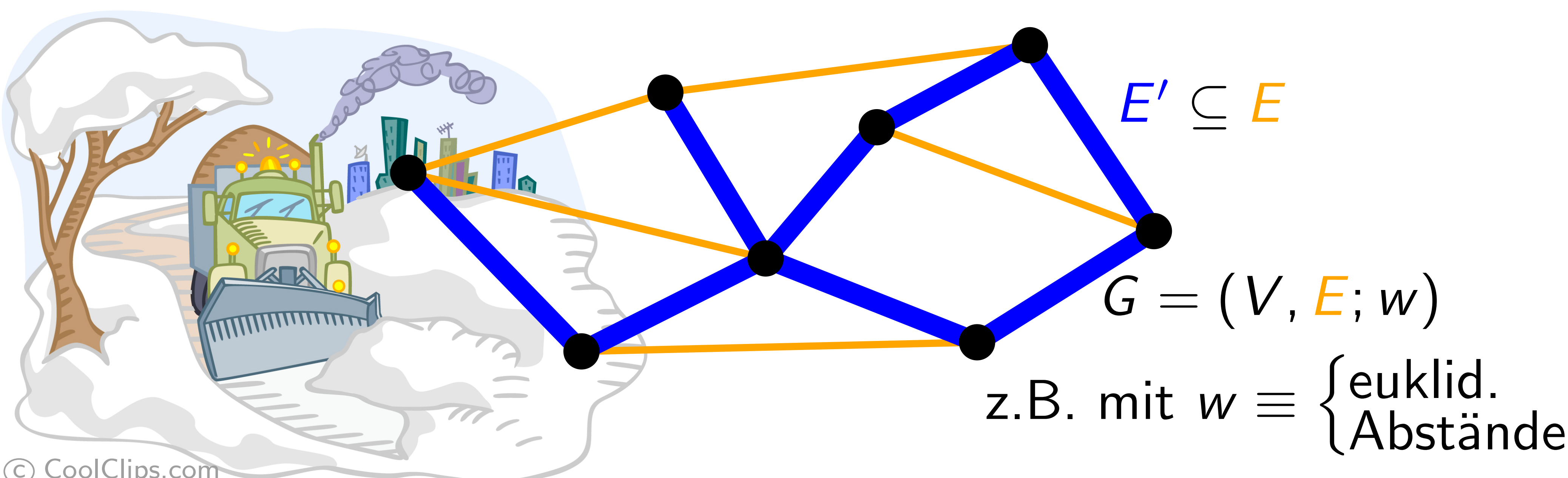
*) Kantengewichte $w\colon E \to \mathbb{R}_{>0}$

Motivation

Gegeben: Zusammenhängendes Straßennetz $G = (V, E; w^\star)$, das eine Menge V von n Städten verbindet

Gesucht: Teilnetz $T = (V, E')$ mit $E' \subseteq E$, so dass

(1) alle Städte in T erreichbar sind



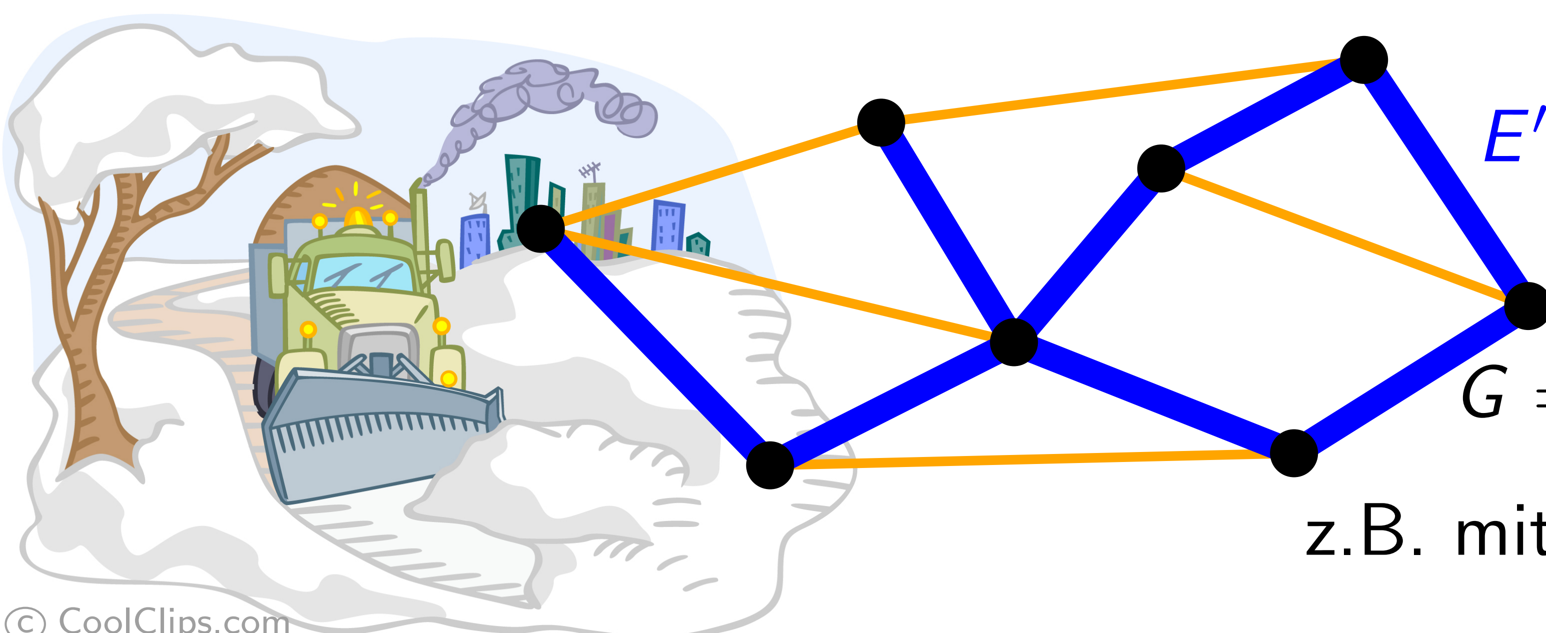
Motivation

*) Kantengewichte $w\colon E \to \mathbb{R}_{>0}$

Gegeben: Zusammenhängendes Straßennetz $G = (V, E; w^{\star})$, das eine Menge V von n Städten verbindet

Gesucht: Teilnetz $T = (V, E')$ mit $E' \subseteq E$, so dass

(1) alle Städte in T erreichbar sind
(T spannt G auf) und



$E' \subseteq E$

$G = (V, E; w)$

z.B. mit $w \equiv \begin{cases} \text{euklid.} \\ \text{Abstände} \end{cases}$

*) Kantengewichte $w\colon E \to \mathbb{R}_{>0}$

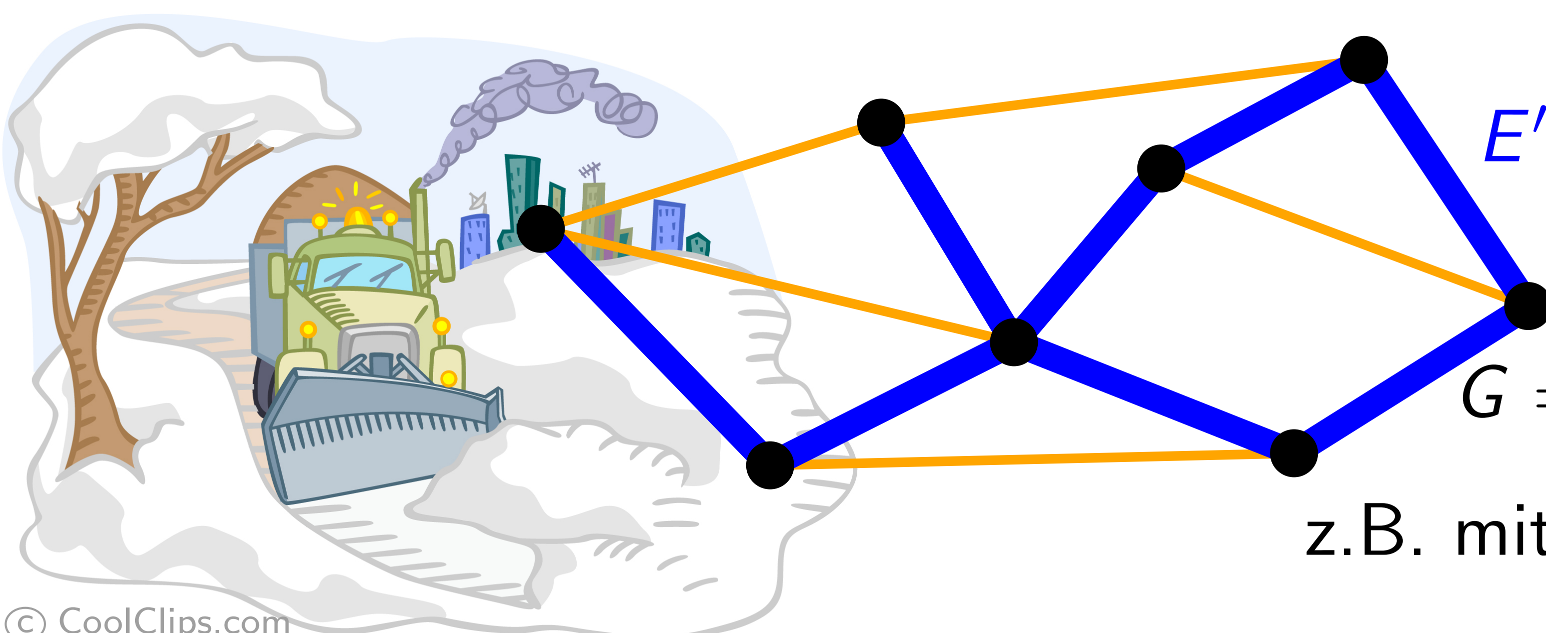
Motivation

Gegeben: Zusammenhängendes Straßennetz $G = (V, E; w^\star)$, das eine Menge V von n Städten verbindet

Gesucht: Teilnetz $T = (V, E')$ mit $E' \subseteq E$, so dass

(1) alle Städte in T erreichbar sind
  (T spannt G auf) und

(2) die „Schneeräumkosten“ $w(E')^{\star\star}$ minimal
  sind unter allen Teilnetzen, die (1) erfüllen.



$E' \subseteq E$

$G = (V, E; w)$

z.B. mit $w \equiv \begin{cases} \text{euklid.} \\ \text{Abstände} \end{cases}$

Motivation

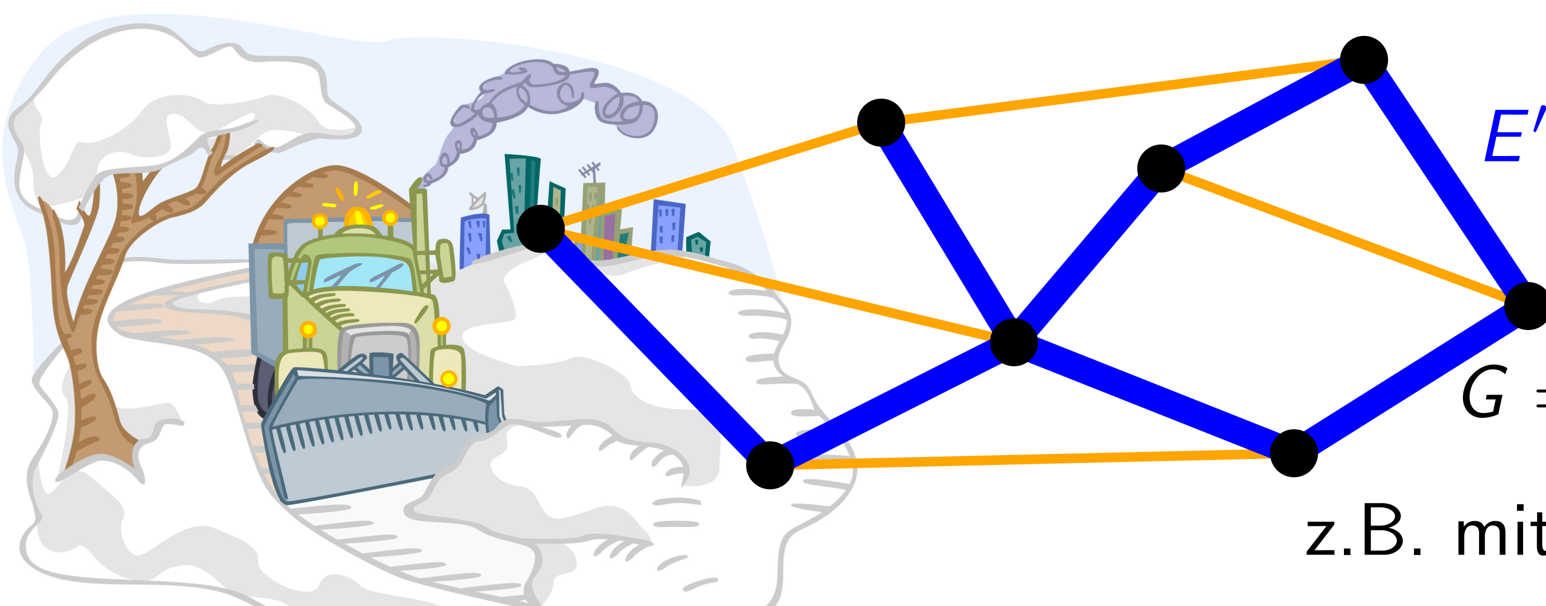
*) Kantengewichte $w \colon E \to \mathbb{R}_{>0}$
**) $w(E') := \sum_{e \in E'} w(e)$

Gegeben: Zusammenhängendes Straßennetz $G = (V, E; w^{\star})$, das eine Menge V von n Städten verbindet

Gesucht: Teilnetz $T = (V, E')$ mit $E' \subseteq E$, so dass

(1) alle Städte in T erreichbar sind
 (T spannt G auf) und

(2) die „Schneeräumkosten“ $w(E')^{\star\star}$ minimal sind unter allen Teilnetzen, die (1) erfüllen.



$E' \subseteq E$

$G = (V, E; w)$

z.B. mit $w \equiv \begin{cases} \text{euklid.} \\ \text{Abstände} \end{cases}$

Motivation

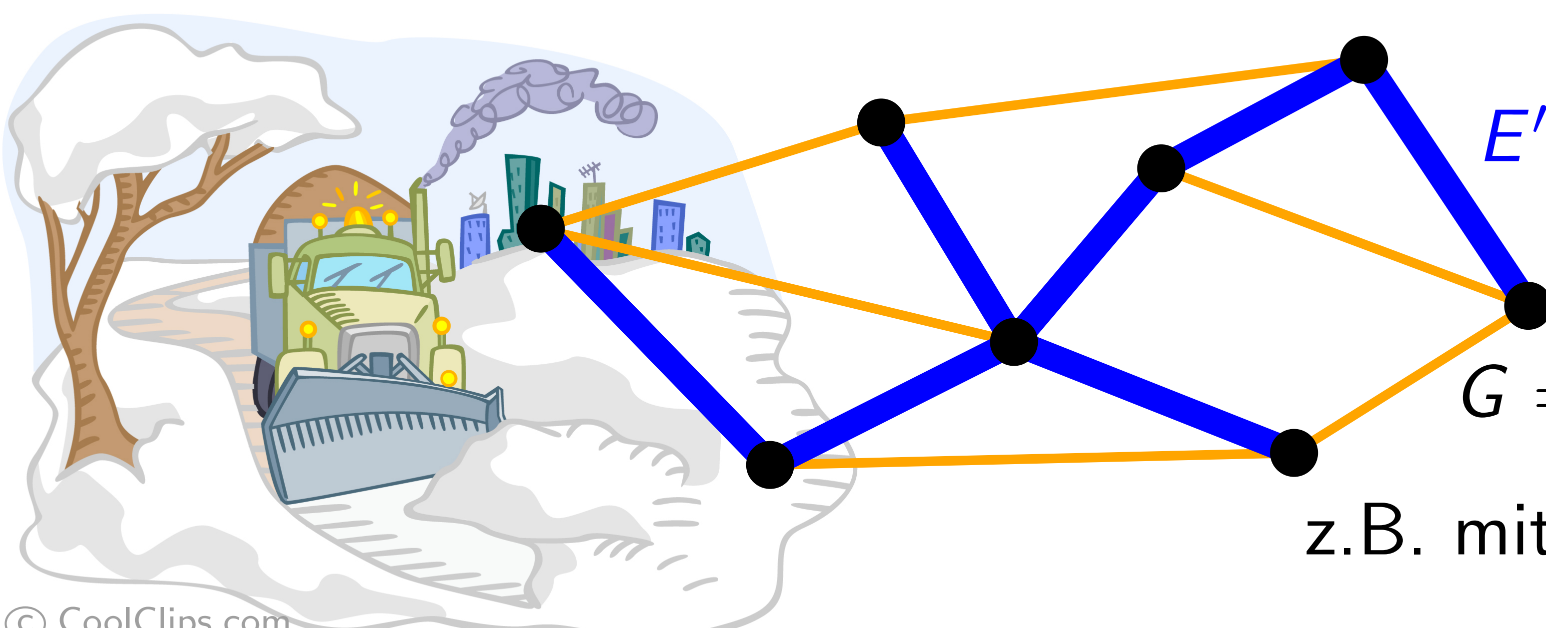
*) Kantengewichte $w \colon E \to \mathbb{R}_{>0}$
**) $w(E') := \sum_{e \in E'} w(e)$

Gegeben: Zusammenhängendes Straßennetz $G = (V, E; w^{\star})$, das eine Menge V von n Städten verbindet

Gesucht: Teilnetz $T = (V, E')$ mit $E' \subseteq E$, so dass

(1) alle Städte in T erreichbar sind
(T spannt G auf) und

(2) die „Schneeräumkosten“ $w(E')^{\star\star}$ minimal sind unter allen Teilnetzen, die (1) erfüllen.



$E' \subseteq E$

$G = (V, E; w)$

z.B. mit $w \equiv \begin{cases} \text{euklid.} \\ \text{Abstände} \end{cases}$

Motivation

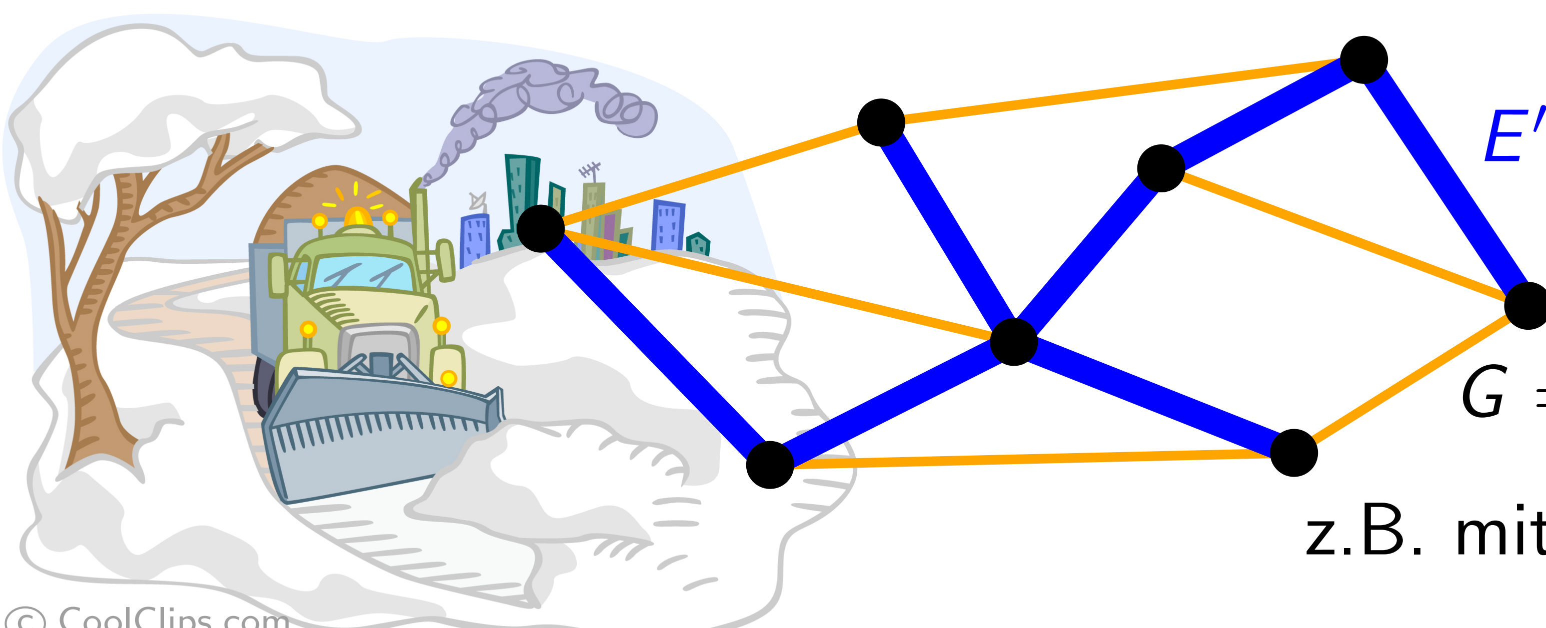
*) Kantengewichte $w\colon E \to \mathbb{R}_{>0}$
**) $w(E') := \sum_{e \in E'} w(e)$

Gegeben: Zusammenhängendes Straßennetz $G = (V, E; w^{\star})$, das eine Menge V von n Städten verbindet

Gesucht: Teilnetz $T = (V, E')$ mit $E' \subseteq E$, so dass

(1) alle Städte in T erreichbar sind
(T spannt G auf) und

(2) die „Schneeräumkosten“ $w(E')^{\star\star}$ minimal sind unter allen Teilnetzen, die (1) erfüllen.



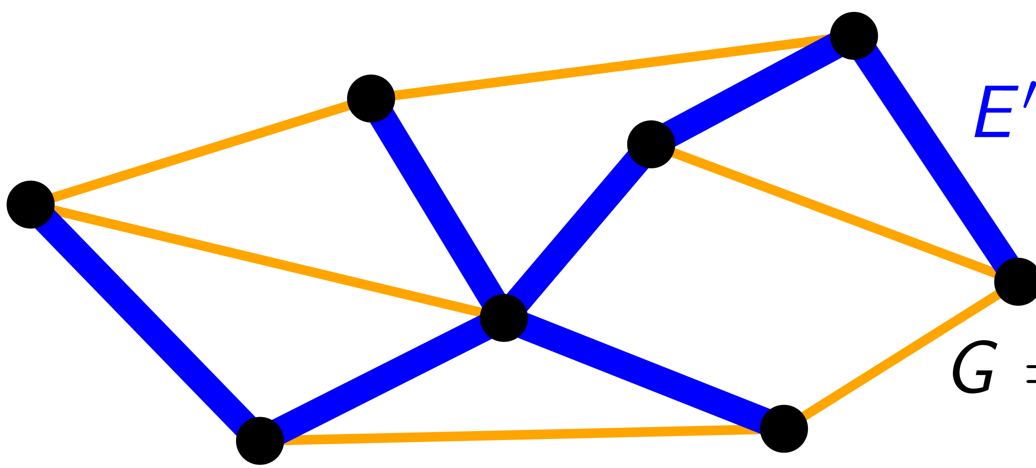
$E' \subseteq E$

$G = (V, E; w)$

z.B. mit $w \equiv \begin{cases}\text{euklid.}\\ \text{Abstände}\end{cases}$

Minimaler Spannbaum

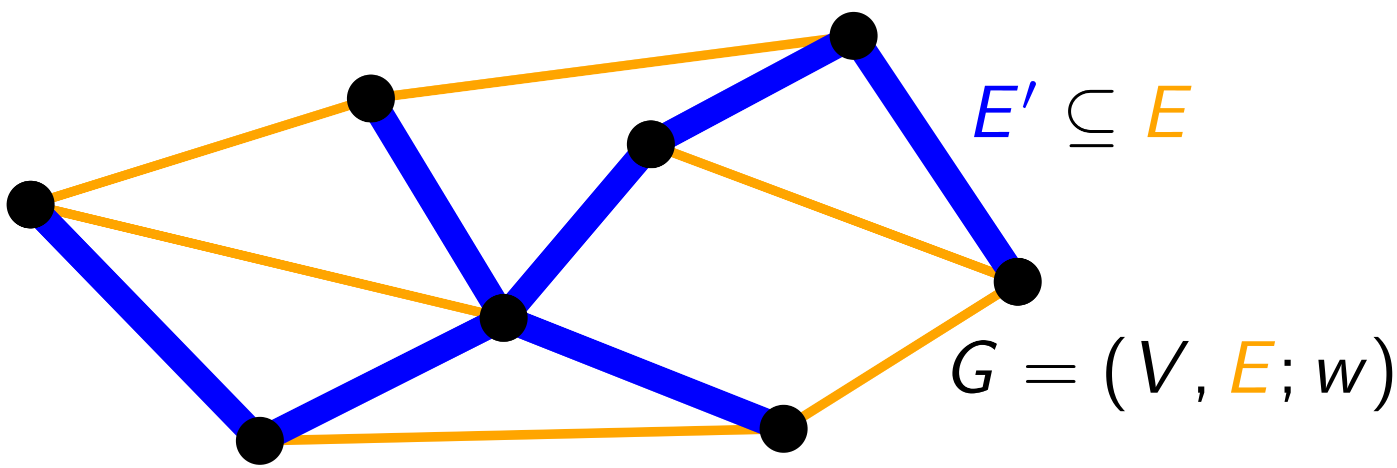
Wegen der Minimalität von $w(E')$ gilt:



Minimaler Spannbaum

Wegen der Minimalität von $w(E')$ gilt:

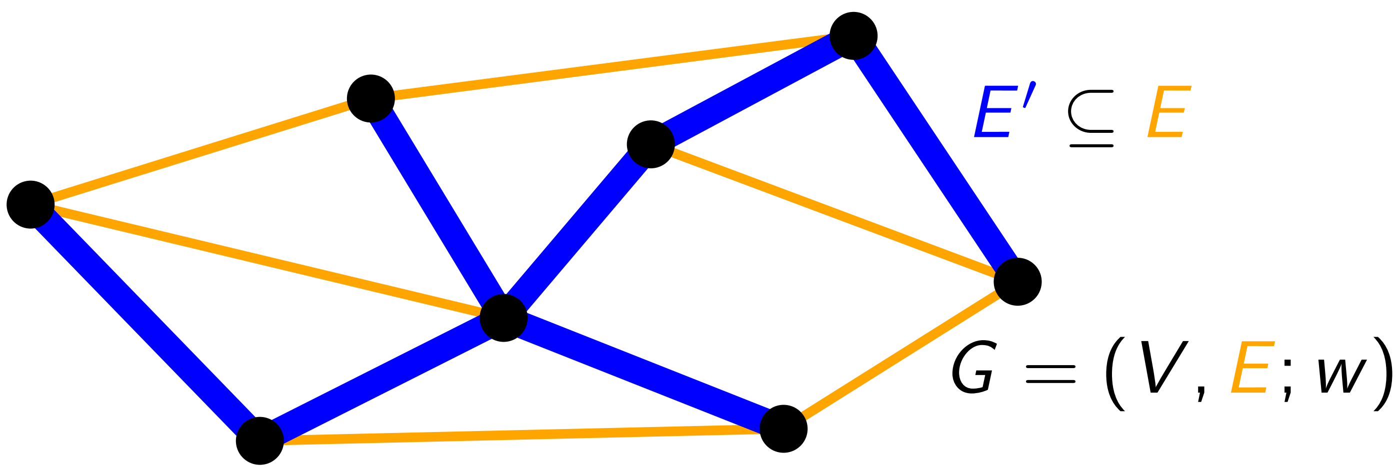
T hat keine Kreise



Minimaler Spannbaum

Wegen der Minimalität von $w(E')$ gilt:

T hat keine Kreise $\Rightarrow$ T ist ein Wald

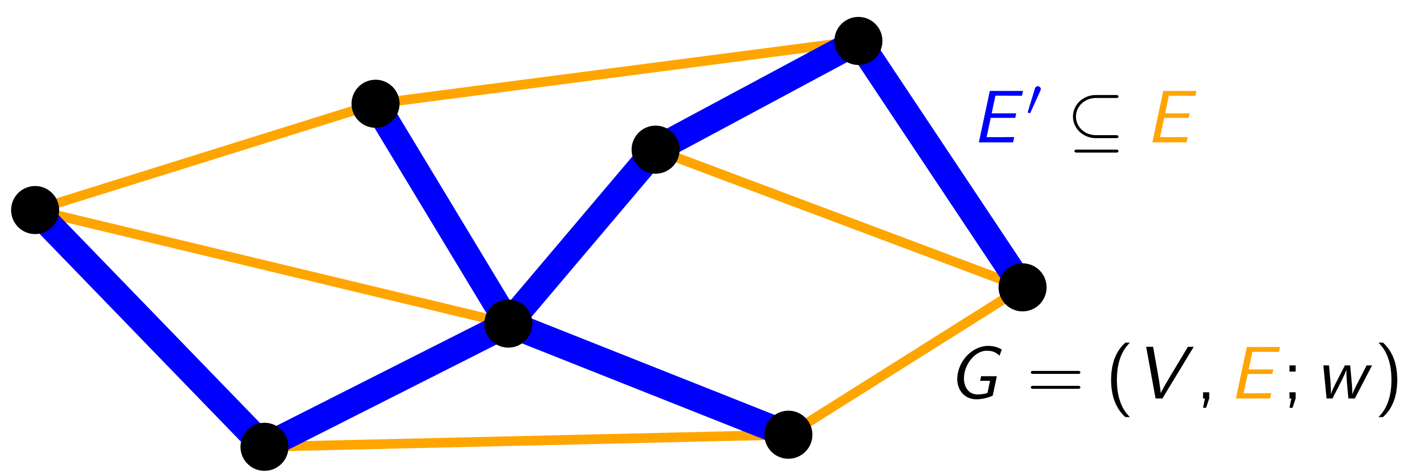


Minimaler Spannbaum

Wegen der Minimalität von $w(E')$ gilt:

T hat keine Kreise $\Rightarrow$ T ist ein Wald

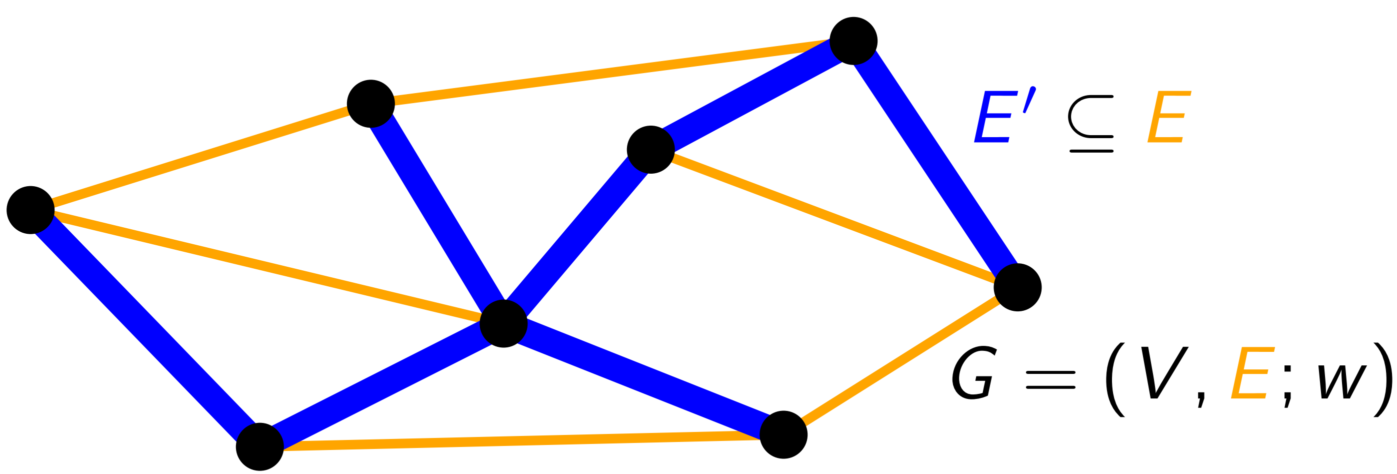
T „erbt“ Zusammenhang von G $\Rightarrow$ T ist ein Baum



Minimaler Spannbaum

Wegen der Minimalität von $w(E')$ gilt:

T hat keine Kreise	$\Rightarrow$	T ist ein Wald
T „erbt“ Zusammenhang von G	$\Rightarrow$	T ist ein Baum
T spannt G auf	$\Rightarrow$	T ist Spannbaum von G

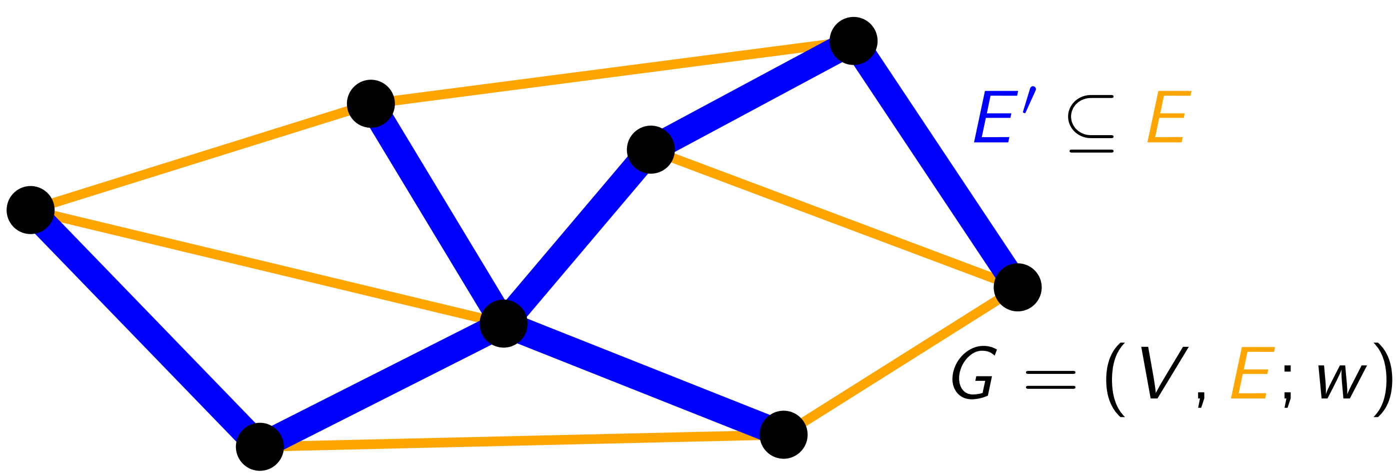


Minimaler Spannbaum

Wegen der Minimalität von $w(E')$ gilt:

T hat keine Kreise	$\Rightarrow$	T ist ein Wald
T „erbt“ Zusammenhang von G	$\Rightarrow$	T ist ein Baum
T spannt G auf	$\Rightarrow$	T ist Spannbaum von G

T hat minimales Gewicht unter *allen* Spannbäumen von G.

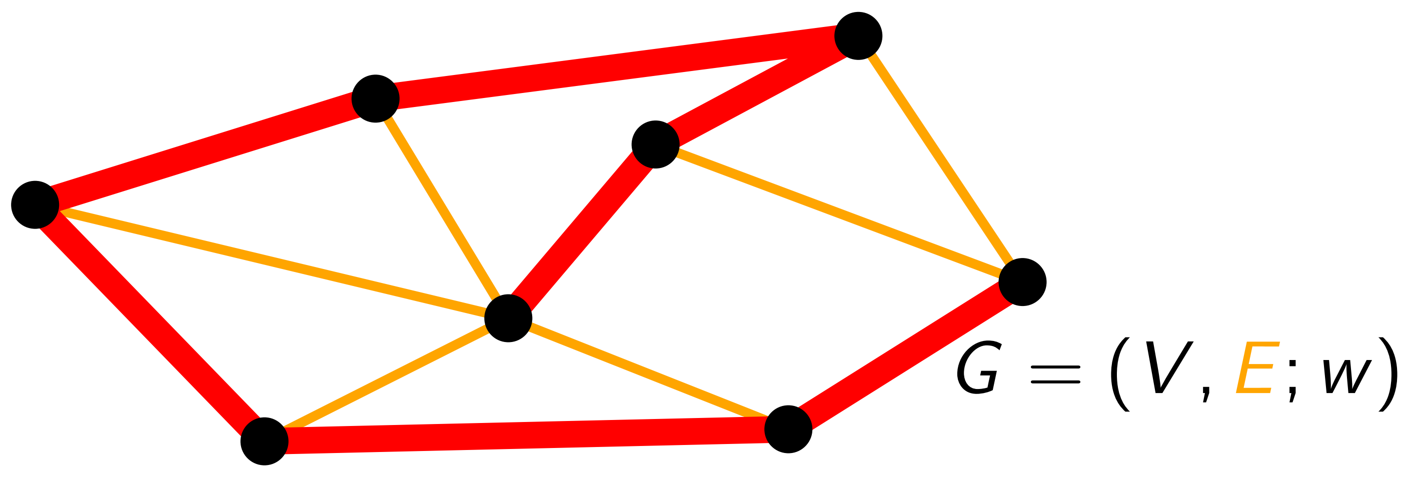


Minimaler Spannbaum

Wegen der Minimalität von $w(E')$ gilt:

T hat keine Kreise	$\Rightarrow$	T ist ein Wald
T „erbt“ Zusammenhang von G	$\Rightarrow$	T ist ein Baum
T spannt G auf	$\Rightarrow$	T ist Spannbaum von G

T hat minimales Gewicht unter *allen* Spannbäumen von G.

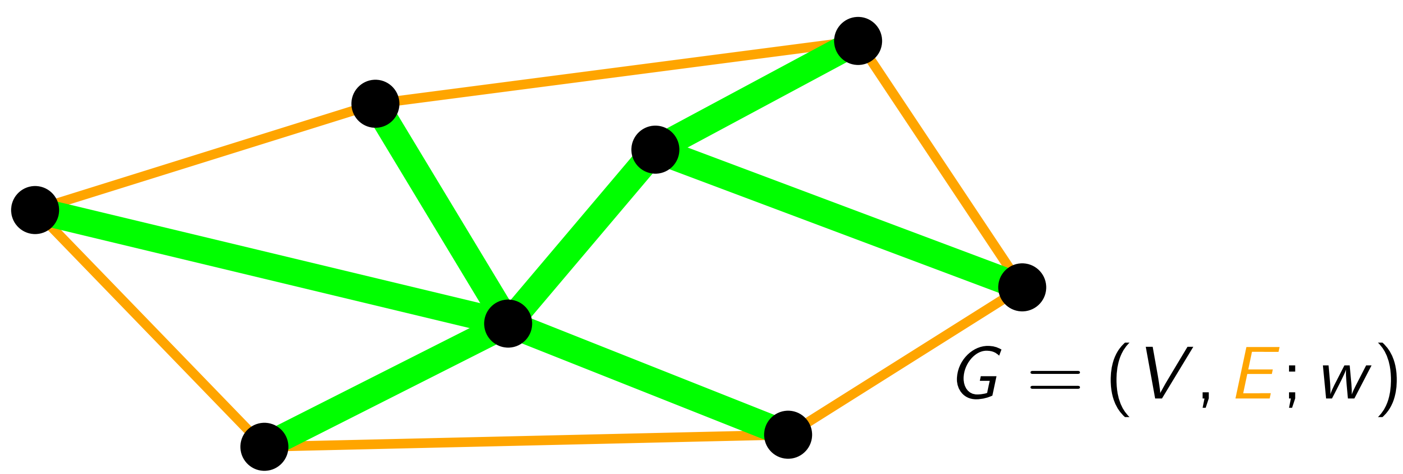


Minimaler Spannbaum

Wegen der Minimalität von $w(E')$ gilt:

T hat keine Kreise	$\Rightarrow$	T ist ein Wald
T „erbt" Zusammenhang von G	$\Rightarrow$	T ist ein Baum
T spannt G auf	$\Rightarrow$	T ist Spannbaum von G

T hat minimales Gewicht unter *allen* Spannbäumen von G.



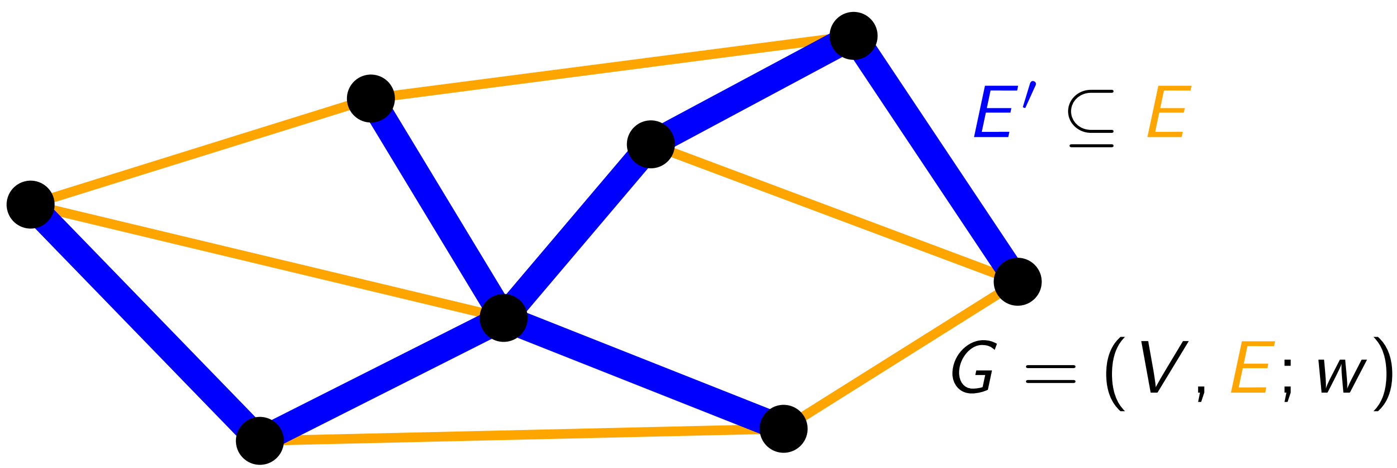
Minimaler Spannbaum

Wegen der Minimalität von $w(E')$ gilt:

T hat keine Kreise	$\Rightarrow$	T ist ein Wald
T „erbt“ Zusammenhang von G	$\Rightarrow$	T ist ein Baum
T spannt G auf	$\Rightarrow$	T ist Spannbaum von G

T hat minimales Gewicht unter *allen* Spannbäumen von G.

Wir nennen T kurz *minimalen Spannbaum* (MSB) von G.



Minimaler Spannbaum

Borůvka 1926

Wegen der Minimalität von $w(E')$ gilt:

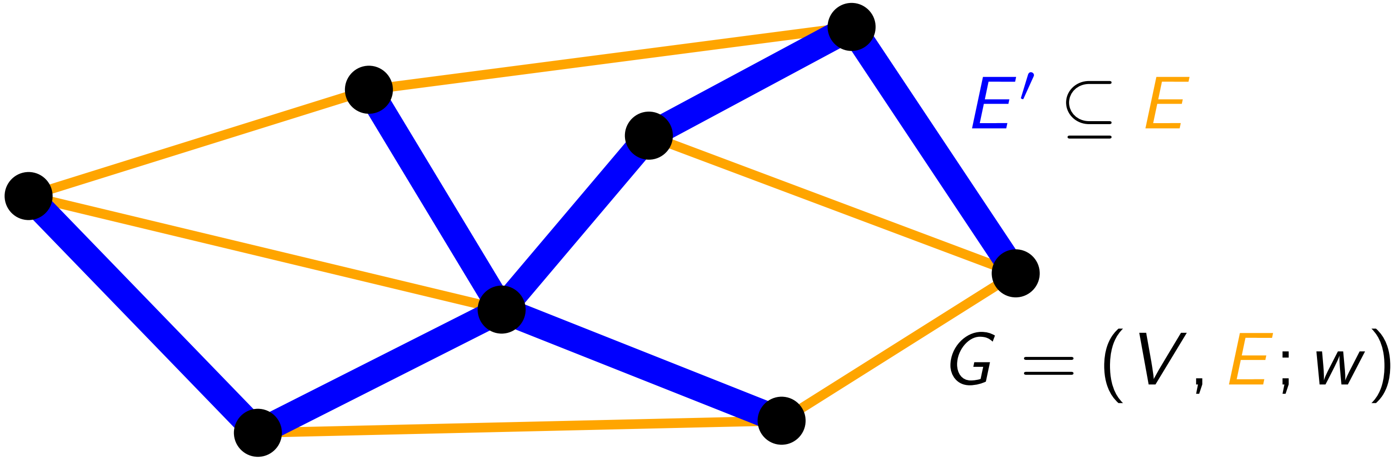
T hat keine Kreise $\Rightarrow$ T ist ein Wald

T „erbt“ Zusammenhang von G $\Rightarrow$ T ist ein Baum

T spannt G auf $\Rightarrow$ T ist Spannbaum von G

T hat minimales Gewicht unter *allen* Spannbäumen von G.

Wir nennen T kurz *minimalen Spannbaum* (MSB) von G.



Minimaler Spannbaum

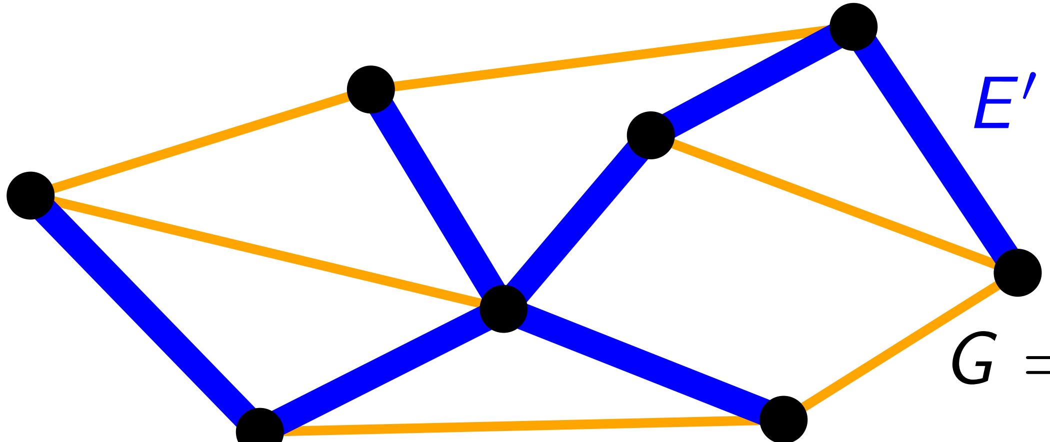
Borůvka 1926

Wegen der Minimalität von $w(E')$ gilt:

T hat keine Kreise	$\Rightarrow$	T ist ein Wald
T „erbt" Zusammenhang von G	$\Rightarrow$	T ist ein Baum
T spannt G auf	$\Rightarrow$	T ist Spannbaum von G

T hat minimales Gewicht unter *allen* Spannbäumen von G.

Wir nennen T kurz *minimalen Spannbaum* (MSB) von G.



Beob. $|E'| =$ **?**

Minimaler Spannbaum

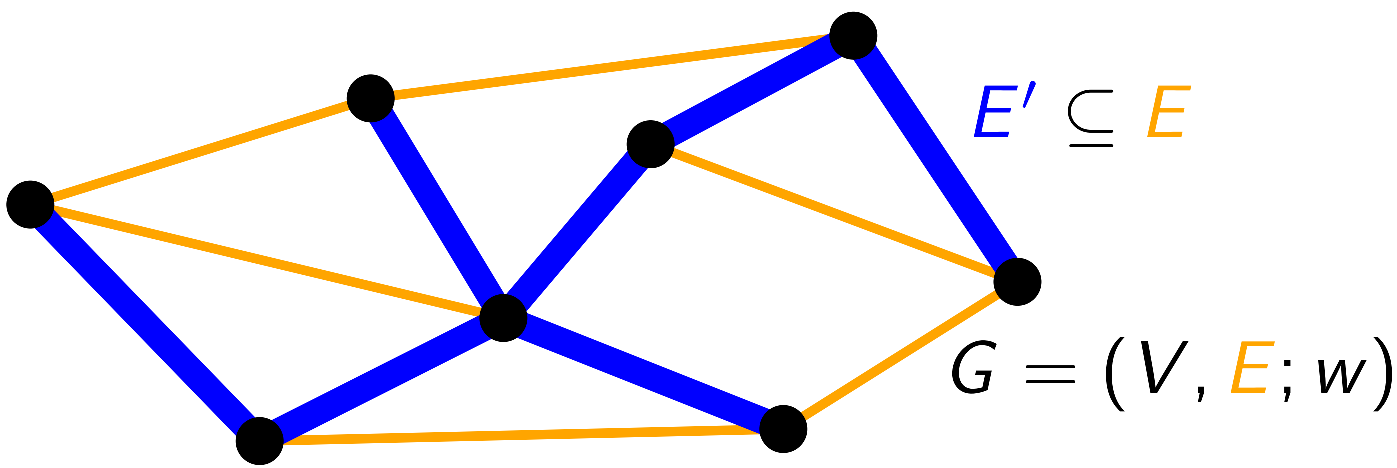
Borůvka 1926

Wegen der Minimalität von $w(E')$ gilt:

T hat keine Kreise	$\Rightarrow$	T ist ein Wald
T „erbt“ Zusammenhang von G	$\Rightarrow$	T ist ein Baum
T spannt G auf	$\Rightarrow$	T ist Spannbaum von G

T hat minimales Gewicht unter *allen* Spannbäumen von G.

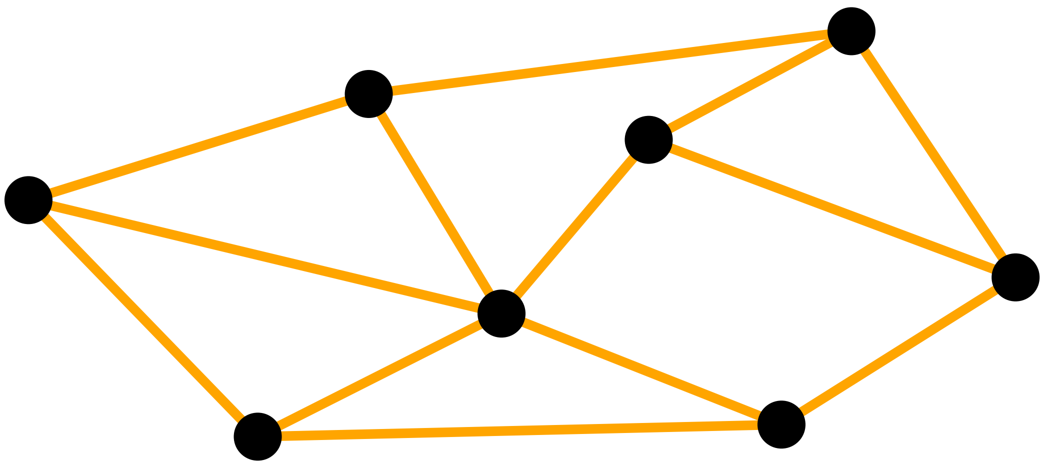
Wir nennen T kurz *minimalen Spannbaum* (MSB) von G.



Beob. $|E'| = |V| - 1$

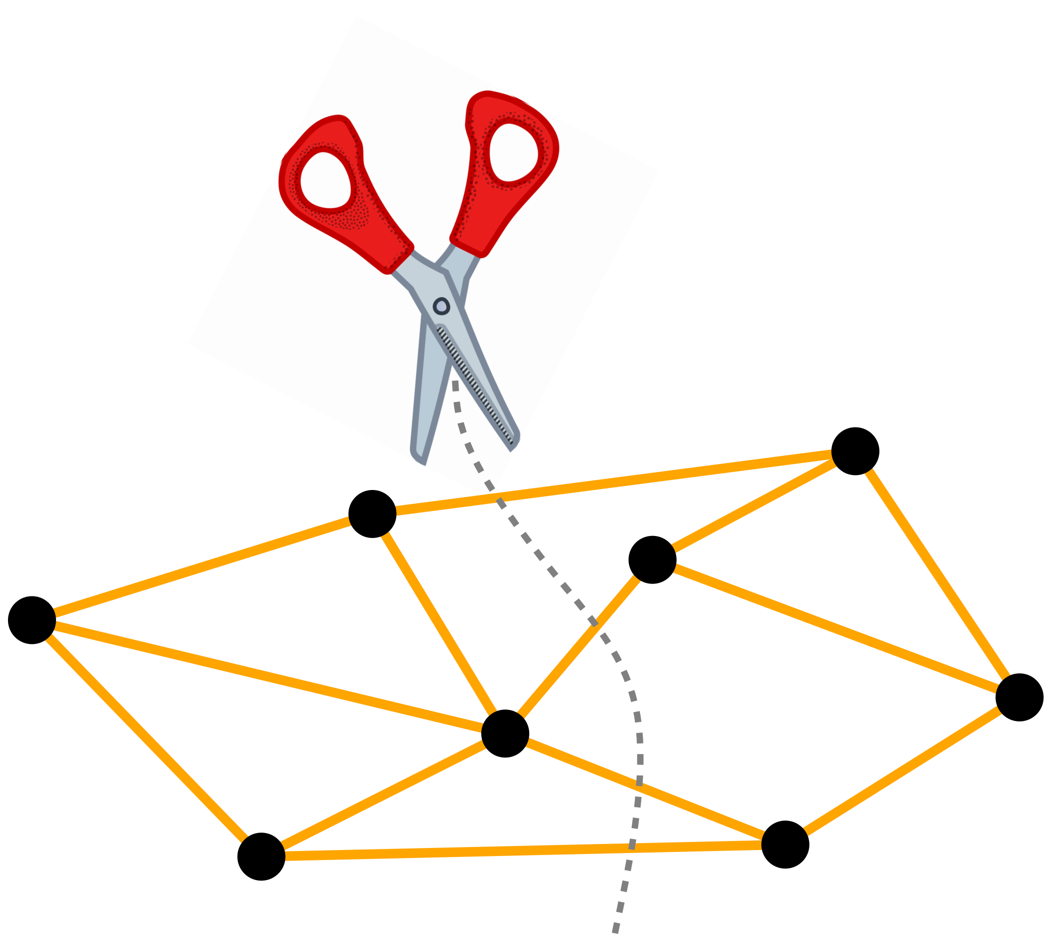
Schnitte

Def. Ein *Schnitt* $(S, V \setminus S)$ eines Graphen $G = (V, E)$ ist eine Zerlegung von V in 2 Teilmengen.



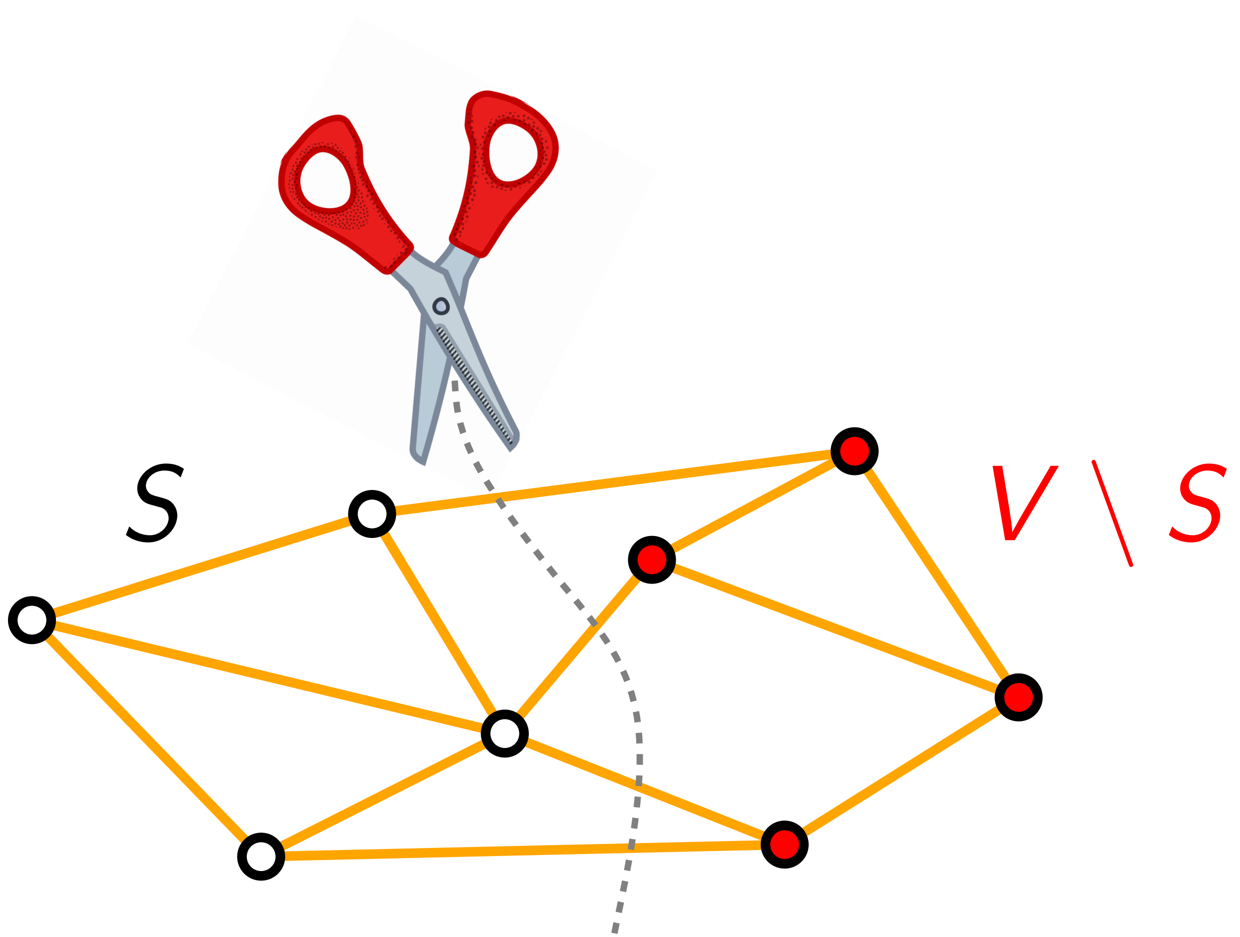
Schnitte

Def. Ein *Schnitt* $(S, V \setminus S)$ eines Graphen $G = (V, E)$ ist eine Zerlegung von V in 2 Teilmengen.



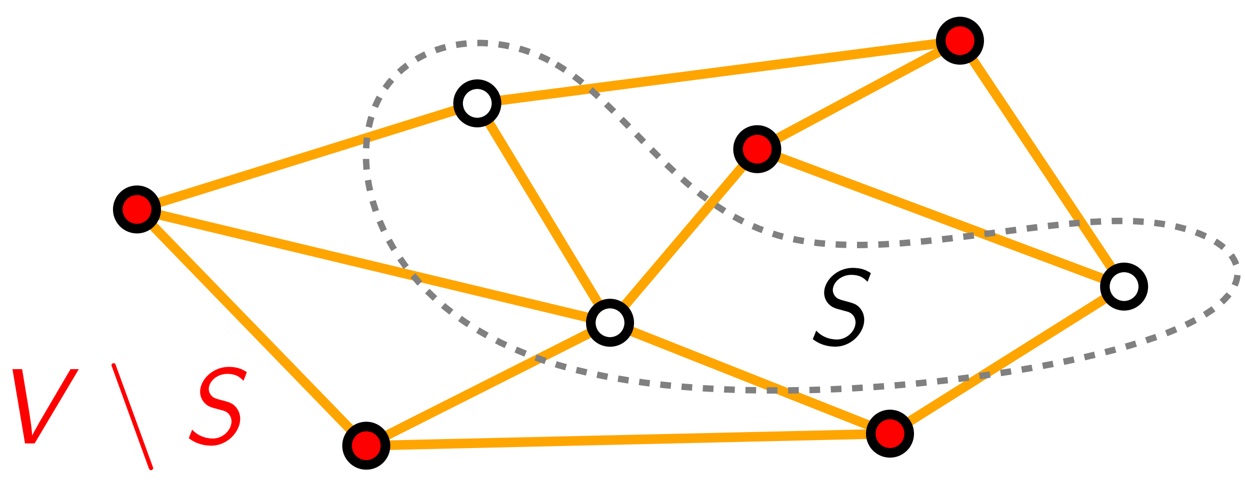
Schnitte

Def. Ein *Schnitt* $(S, V \setminus S)$ eines Graphen $G = (V, E)$ ist eine Zerlegung von V in 2 Teilmengen.



Schnitte

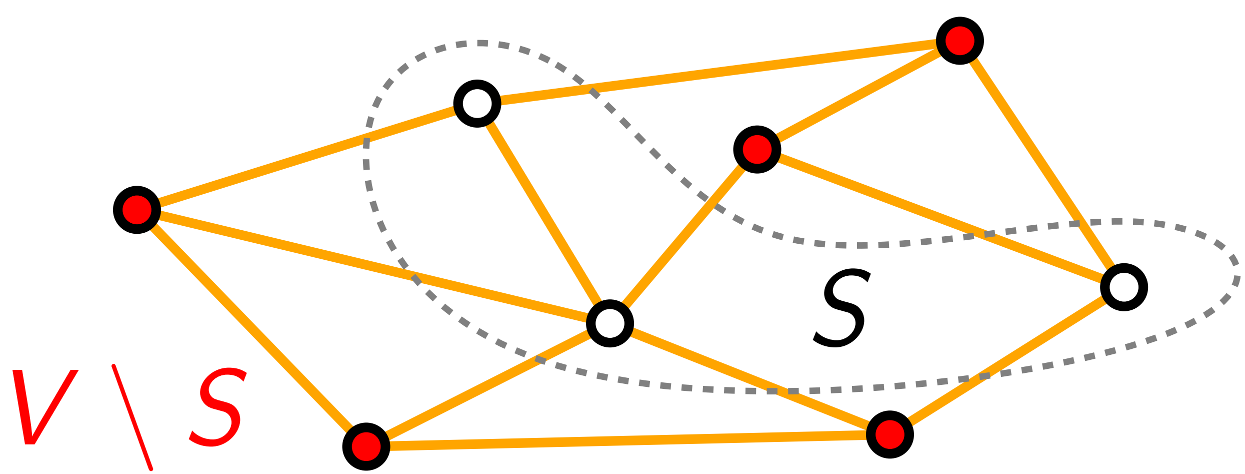
Def. Ein *Schnitt* $(S, V \setminus S)$ eines Graphen $G = (V, E)$ ist eine Zerlegung von V in 2 Teilmengen.



Schnitte

Def. Ein *Schnitt* $(S, V \setminus S)$ eines Graphen $G = (V, E)$ ist eine Zerlegung von V in 2 Teilmengen.

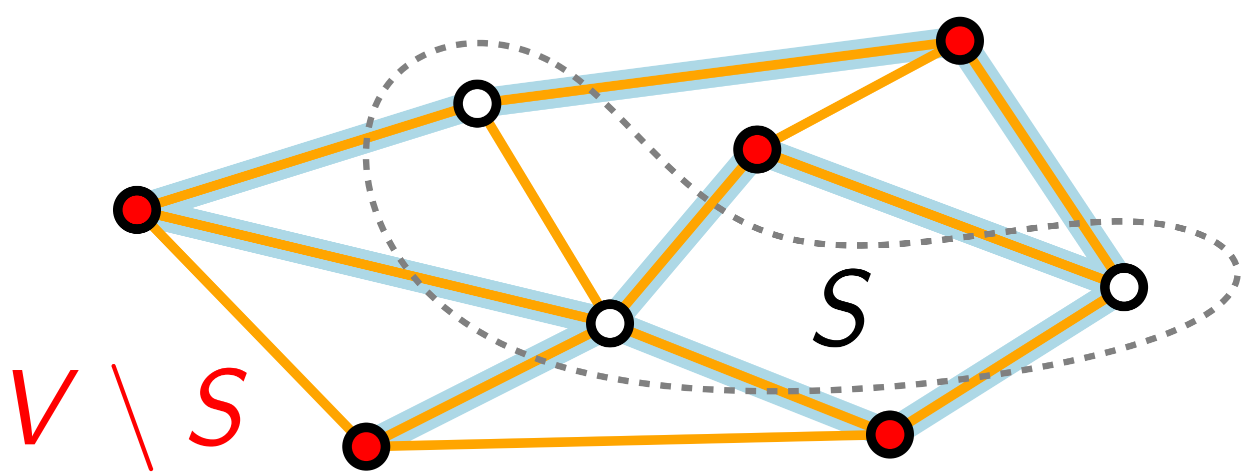
Eine Kante uv *kreuzt* $(S, V \setminus S)$, wenn $u \in S$ und $v \in V \setminus S$ (oder andersherum).



Schnitte

Def. Ein *Schnitt* $(S, V \setminus S)$ eines Graphen $G = (V, E)$ ist eine Zerlegung von V in 2 Teilmengen.

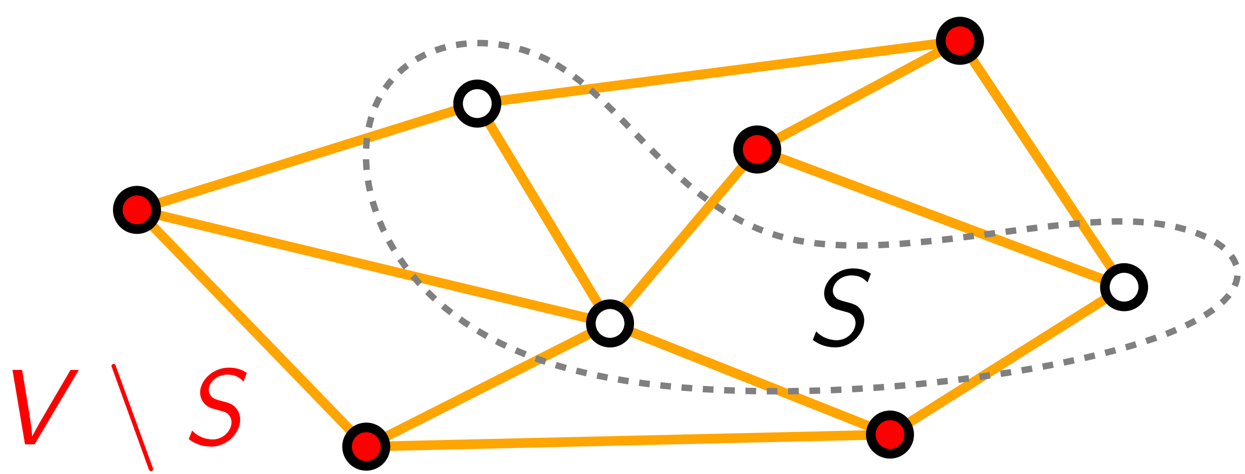
Eine Kante uv *kreuzt* $(S, V \setminus S)$, wenn $u \in S$ und $v \in V \setminus S$ (oder andersherum).



Schnitte

Def. Ein *Schnitt* $(S, V \setminus S)$ eines Graphen $G = (V, E)$ ist eine Zerlegung von V in 2 Teilmengen.

Eine Kante uv *kreuzt* $(S, V \setminus S)$, wenn $u \in S$ und $v \in V \setminus S$ (oder andersherum).

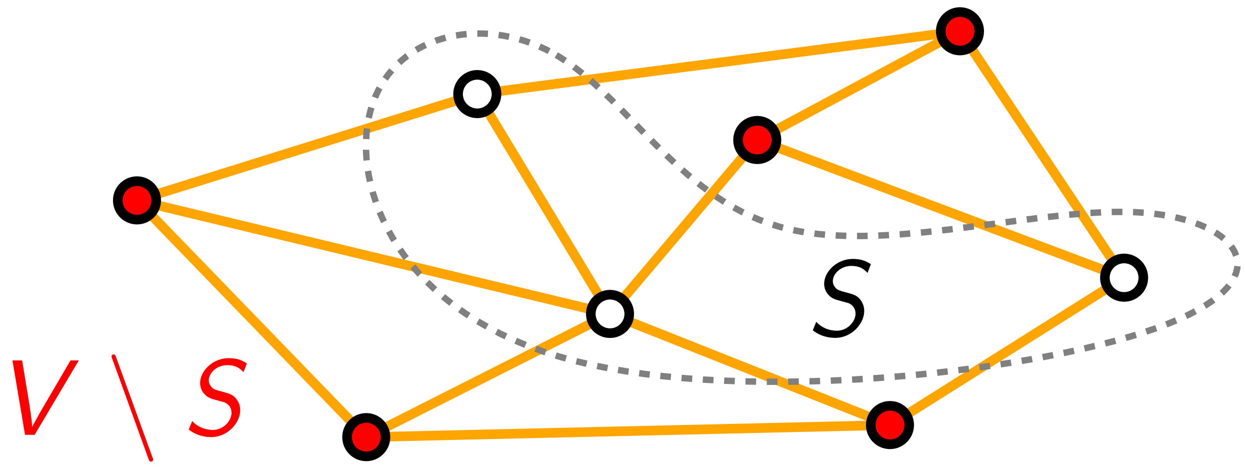


Schnitte

Def. Ein *Schnitt* $(S, V \setminus S)$ eines Graphen $G = (V, E)$ ist eine Zerlegung von V in 2 Teilmengen.

Eine Kante uv *kreuzt* $(S, V \setminus S)$, wenn $u \in S$ und $v \in V \setminus S$ (oder andersherum).

Eine Kante uv, die einen Schnitt kreuzt, ist *leicht*, wenn alle Kanten, die den Schnitt kreuzen, mindestens $w(uv)$ wiegen.

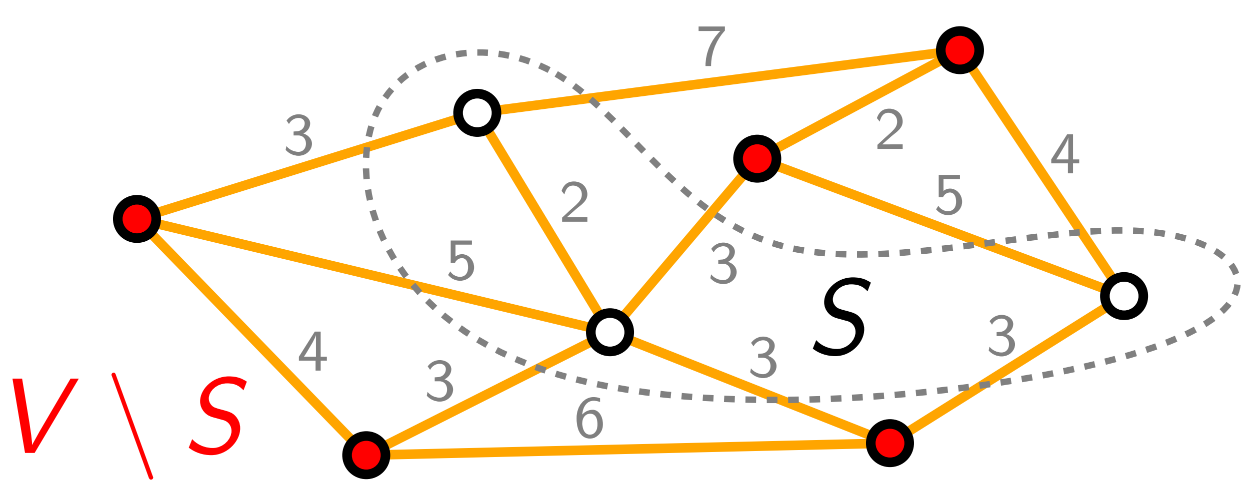


Schnitte

Def. Ein *Schnitt* $(S, V \setminus S)$ eines Graphen $G = (V, E)$ ist eine Zerlegung von V in 2 Teilmengen.

Eine Kante uv *kreuzt* $(S, V \setminus S)$, wenn $u \in S$ und $v \in V \setminus S$ (oder andersherum).

Eine Kante uv, die einen Schnitt kreuzt, ist *leicht*, wenn alle Kanten, die den Schnitt kreuzen, mindestens $w(uv)$ wiegen.

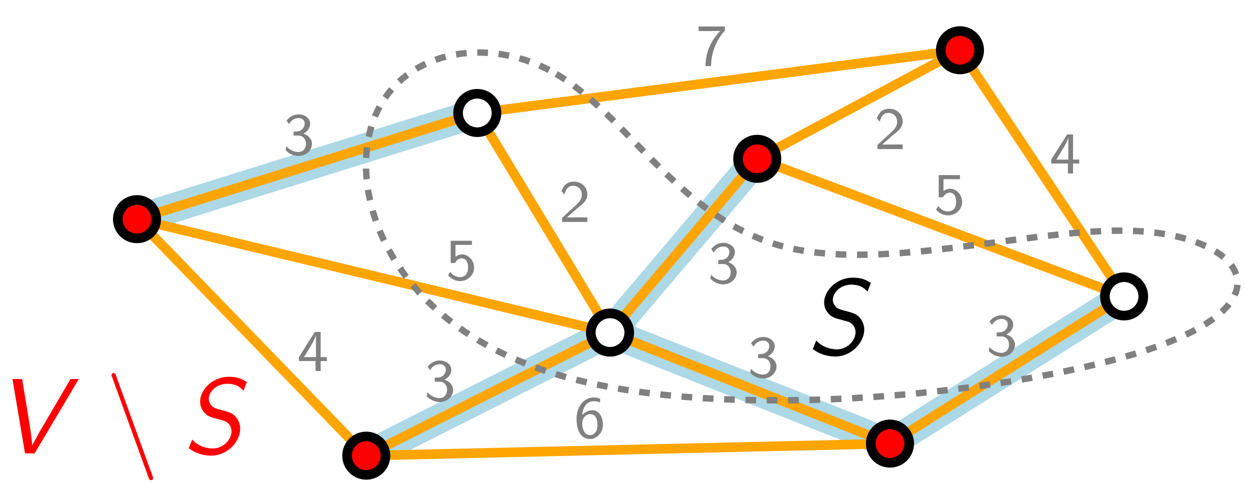


Schnitte

Def. Ein *Schnitt* $(S, V \setminus S)$ eines Graphen $G = (V, E)$ ist eine Zerlegung von V in 2 Teilmengen.

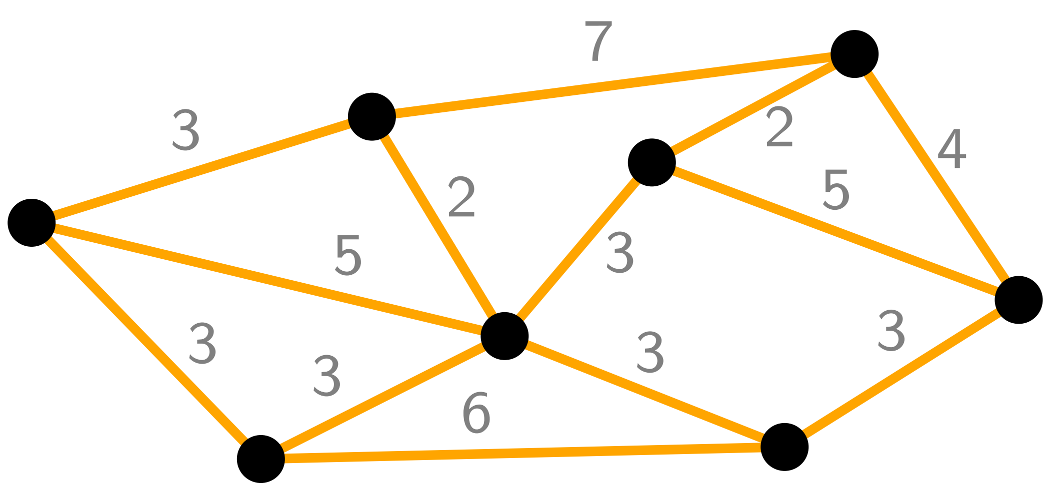
Eine Kante uv *kreuzt* $(S, V \setminus S)$, wenn $u \in S$ und $v \in V \setminus S$ (oder andersherum).

Eine Kante uv, die einen Schnitt kreuzt, ist *leicht*, wenn alle Kanten, die den Schnitt kreuzen, mindestens $w(uv)$ wiegen.



Allgemeiner Greedy Algorithmus

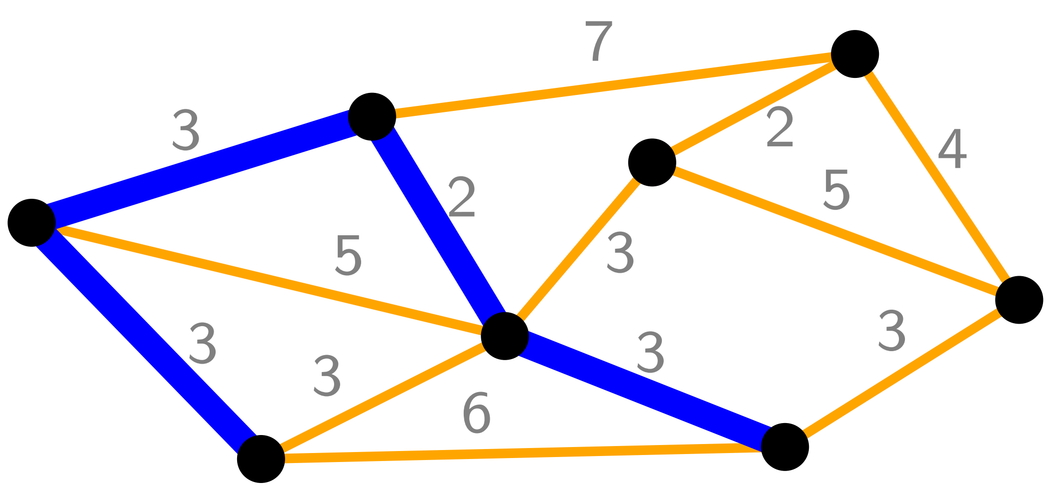
Färbe alle Kanten des Graphen:



Allgemeiner Greedy Algorithmus

Färbe alle Kanten des Graphen:

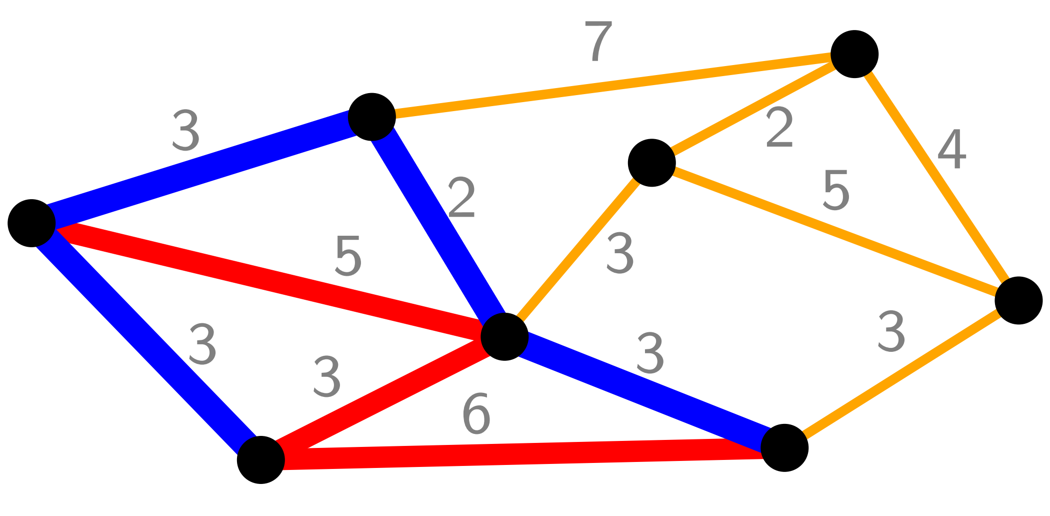
- blau: Kante aus MSB



Allgemeiner Greedy Algorithmus

Färbe alle Kanten des Graphen:

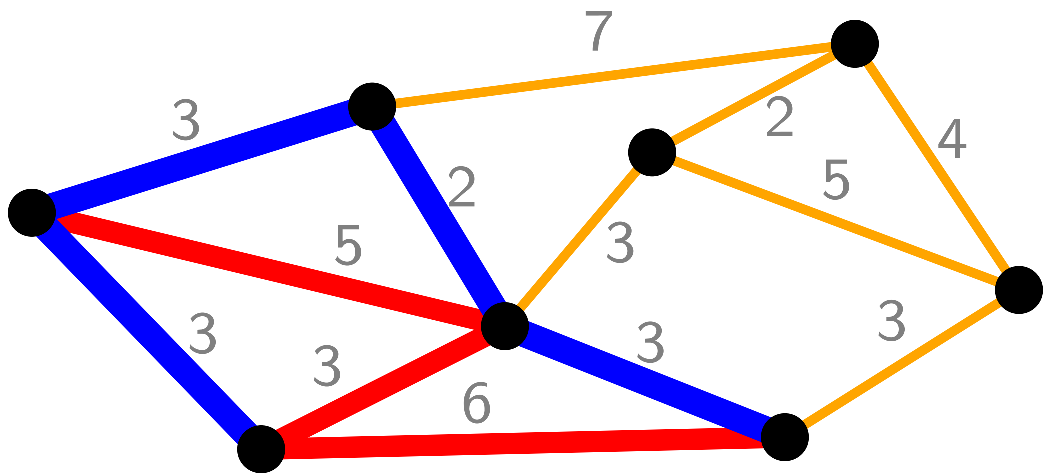
- blau: Kante aus MSB
- rot: Kante nicht aus MSB



Allgemeiner Greedy Algorithmus

Färbe alle Kanten des Graphen:

- blau: Kante aus MSB
- rot: Kante nicht aus MSB
- ungefärbt: Noch nicht entschieden

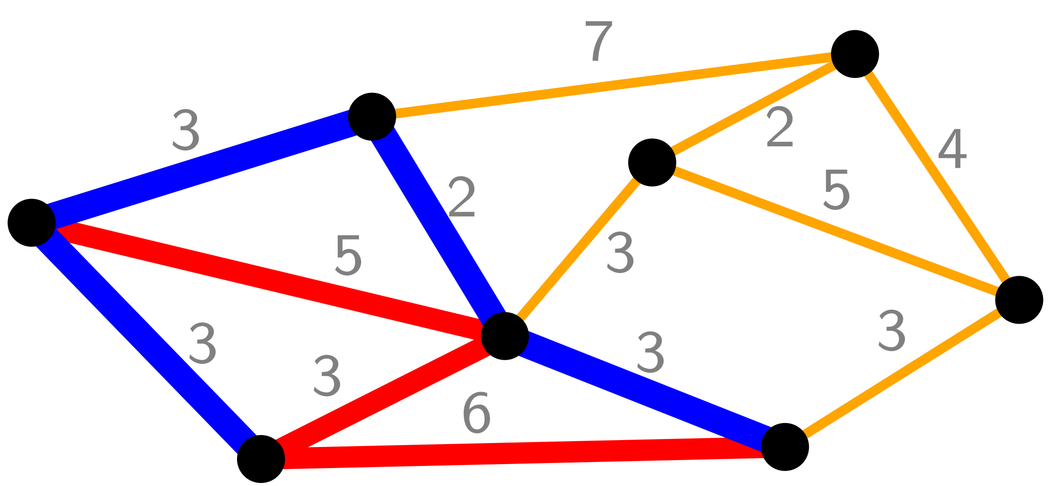


Allgemeiner Greedy Algorithmus

Färbe alle Kanten des Graphen:

- blau: Kante aus MSB
- rot: Kante nicht aus MSB
- ungefärbt: Noch nicht entschieden

Verwende 2 Regeln:



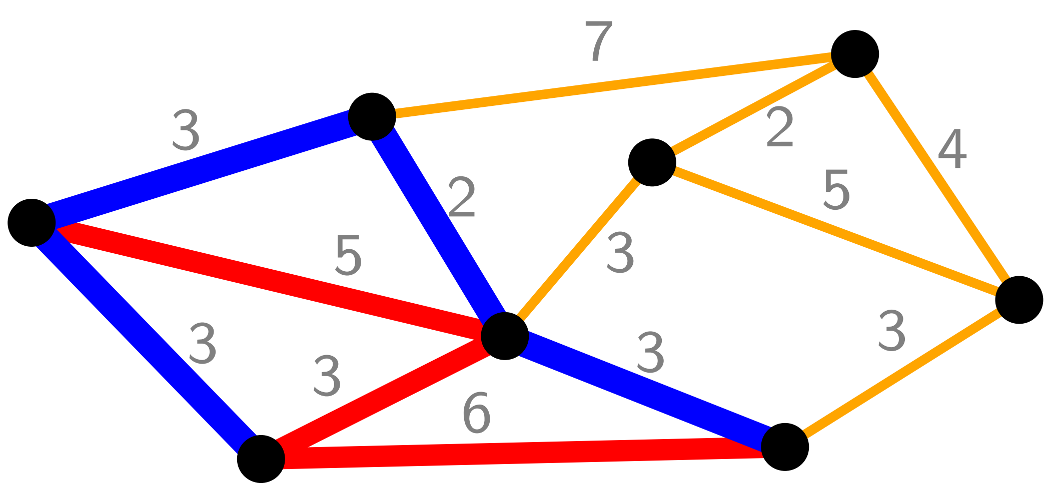
Allgemeiner Greedy Algorithmus

Färbe alle Kanten des Graphen:

- blau: Kante aus MSB
- rot: Kante nicht aus MSB
- ungefärbt: Noch nicht entschieden

Verwende 2 Regeln:

Blaue Regel:



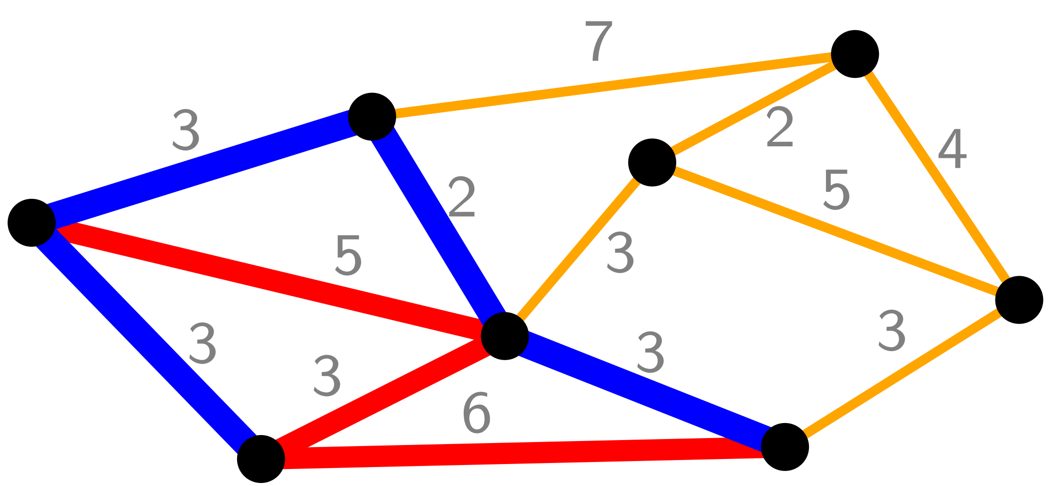
Allgemeiner Greedy Algorithmus

Färbe alle Kanten des Graphen:

- blau: Kante aus MSB
- rot: Kante nicht aus MSB
- ungefärbt: Noch nicht entschieden

Verwende 2 Regeln:

Blaue Regel:
Wähle Schnitt, den keine blaue Kante kreuzt



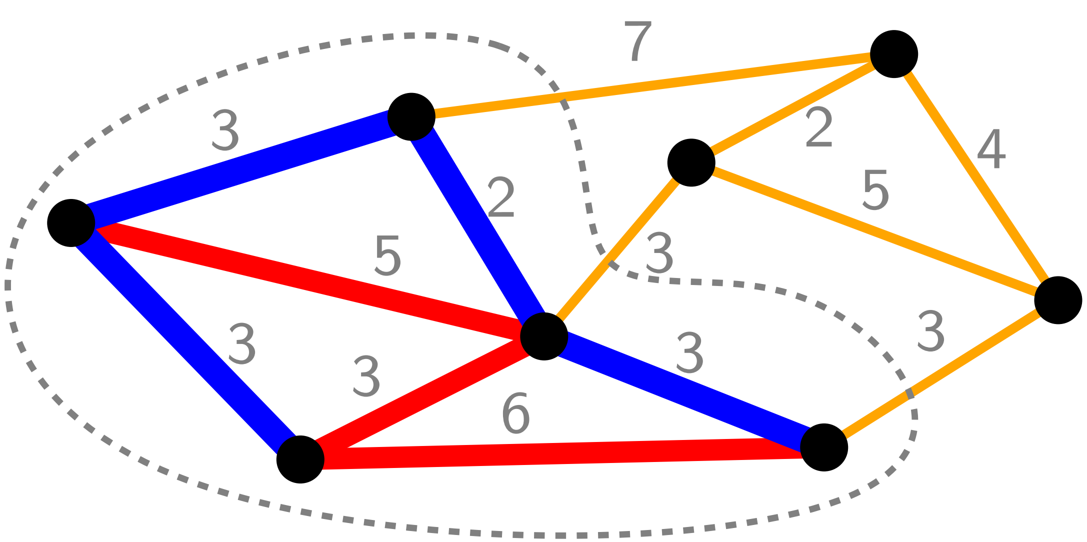
Allgemeiner Greedy Algorithmus

Färbe alle Kanten des Graphen:

- blau: Kante aus MSB
- rot: Kante nicht aus MSB
- ungefärbt: Noch nicht entschieden

Verwende 2 Regeln:

Blaue Regel:
Wähle Schnitt, den keine blaue Kante kreuzt



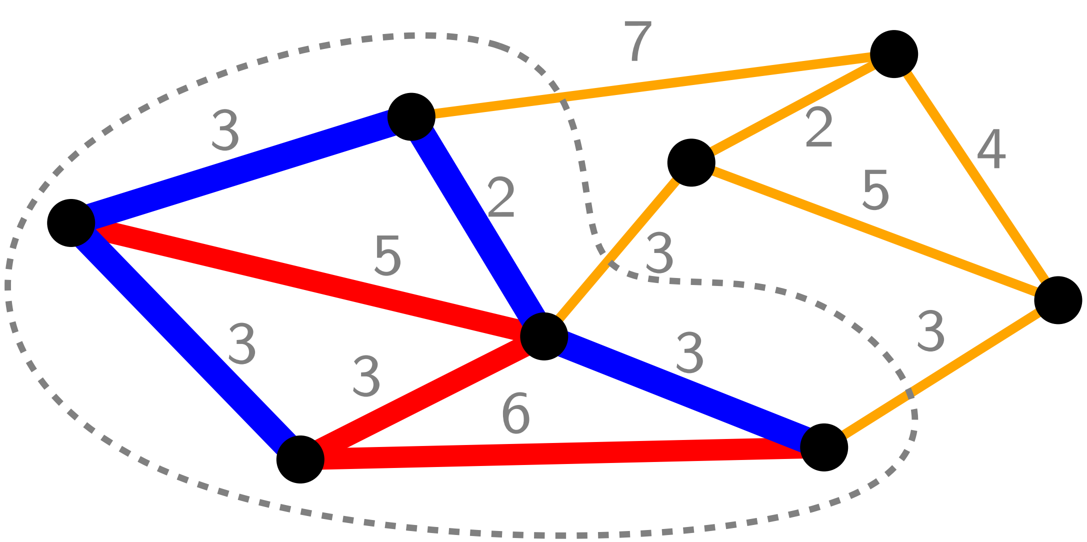
Allgemeiner Greedy Algorithmus

Färbe alle Kanten des Graphen:

- blau: Kante aus MSB
- rot: Kante nicht aus MSB
- ungefärbt: Noch nicht entschieden

Verwende 2 Regeln:

Blaue Regel:
Wähle Schnitt, den keine blaue Kante kreuzt
Färbe leichte Kante blau



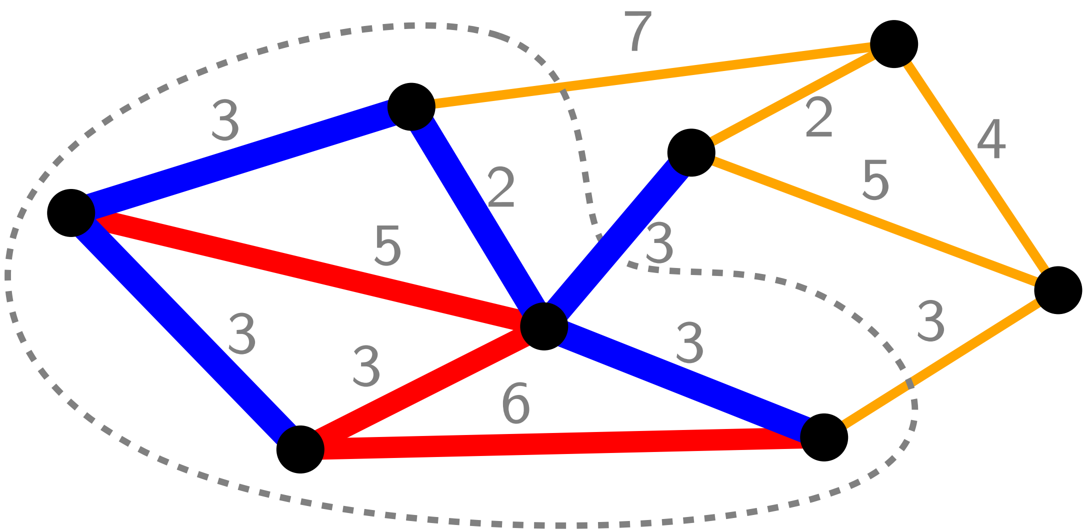
Allgemeiner Greedy Algorithmus

Färbe alle Kanten des Graphen:

- blau: Kante aus MSB
- rot: Kante nicht aus MSB
- ungefärbt: Noch nicht entschieden

Verwende 2 Regeln:

Blaue Regel:
Wähle Schnitt, den keine blaue Kante kreuzt
Färbe leichte Kante blau



Allgemeiner Greedy Algorithmus

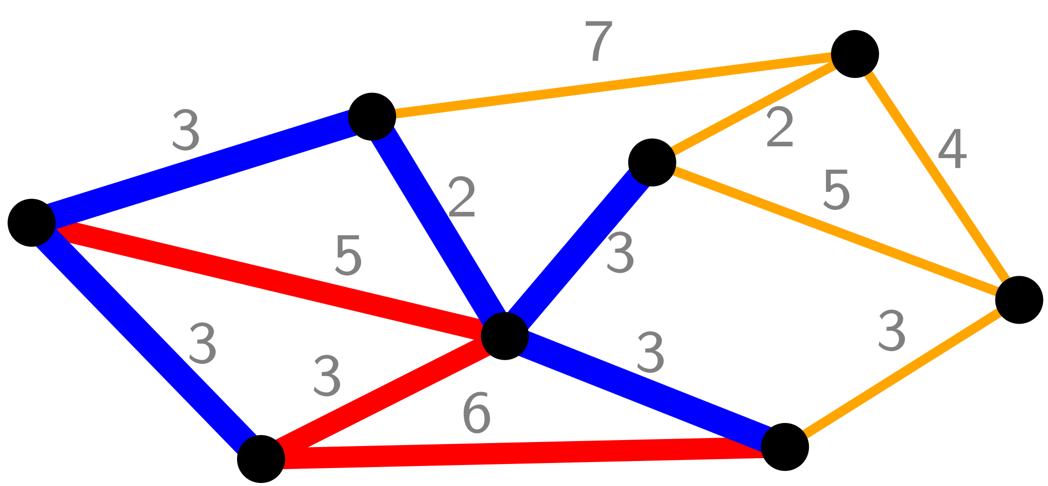
Färbe alle Kanten des Graphen:

- blau: Kante aus MSB
- rot: Kante nicht aus MSB
- ungefärbt: Noch nicht entschieden

Verwende 2 Regeln:

Blaue Regel:
Wähle Schnitt, den keine blaue Kante kreuzt
Färbe leichte Kante blau

Rote Regel:



Allgemeiner Greedy Algorithmus

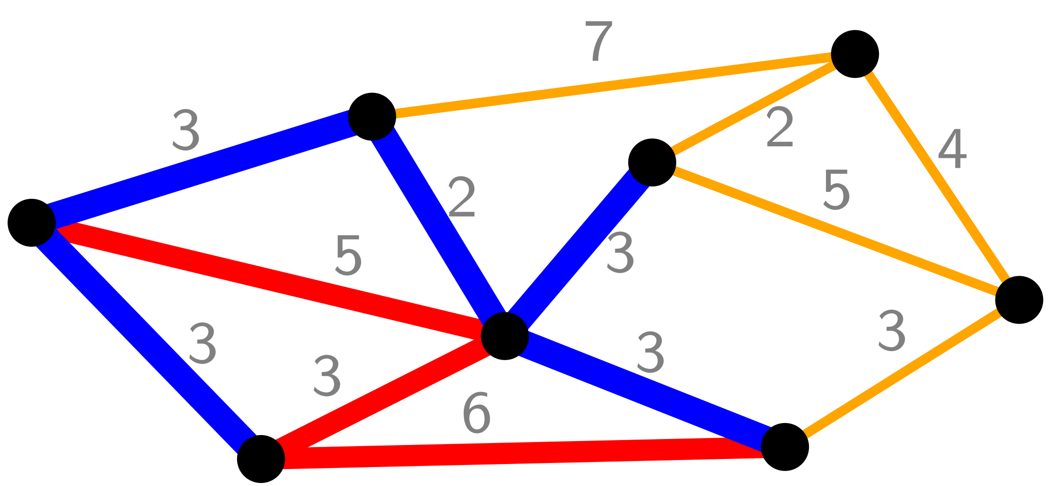
Färbe alle Kanten des Graphen:

- blau: Kante aus MSB
- rot: Kante nicht aus MSB
- ungefärbt: Noch nicht entschieden

Verwende 2 Regeln:

Blaue Regel:
Wähle Schnitt, den keine blaue Kante kreuzt
Färbe leichte Kante blau

Rote Regel:
Wähle Kreis ohne rote Kante



Allgemeiner Greedy Algorithmus

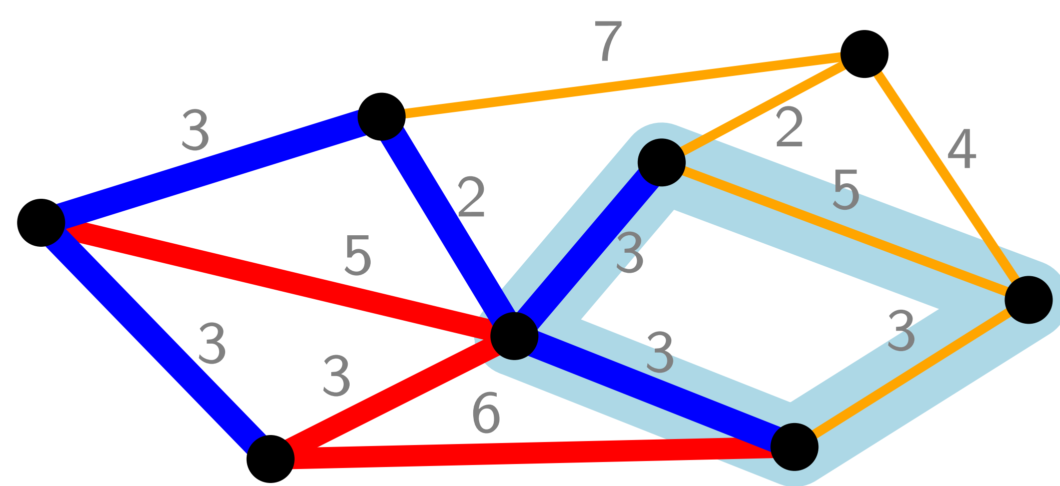
Färbe alle Kanten des Graphen:

- blau: Kante aus MSB
- rot: Kante nicht aus MSB
- ungefärbt: Noch nicht entschieden

Verwende 2 Regeln:

Blaue Regel:
Wähle Schnitt, den keine blaue Kante kreuzt
Färbe leichte Kante blau

Rote Regel:
Wähle Kreis ohne rote Kante



Allgemeiner Greedy Algorithmus

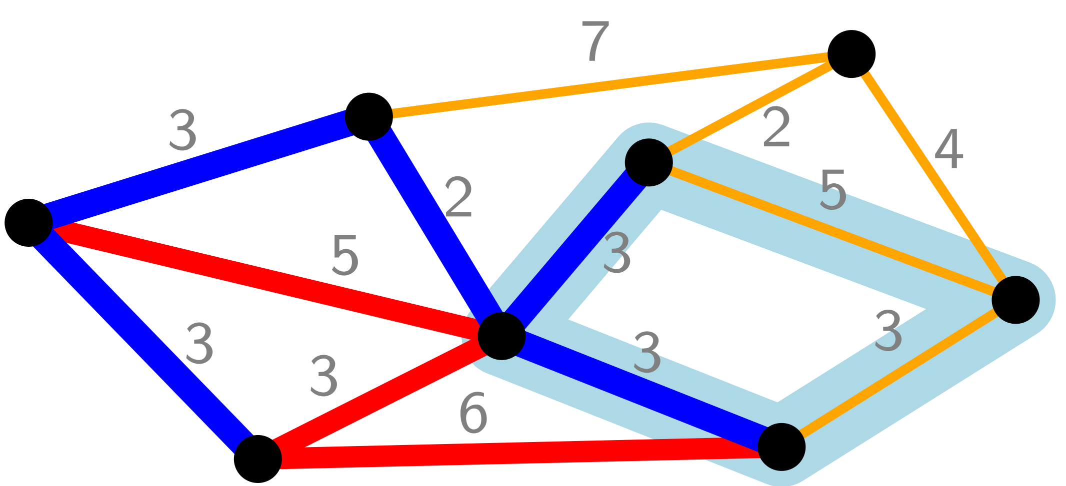
Färbe alle Kanten des Graphen:

- blau: Kante aus MSB
- rot: Kante nicht aus MSB
- ungefärbt: Noch nicht entschieden

Verwende 2 Regeln:

Blaue Regel:
Wähle Schnitt, den keine blaue Kante kreuzt
Färbe leichte Kante blau

Rote Regel:
Wähle Kreis ohne rote Kante
Färbe größte ungefärbte rot



Allgemeiner Greedy Algorithmus

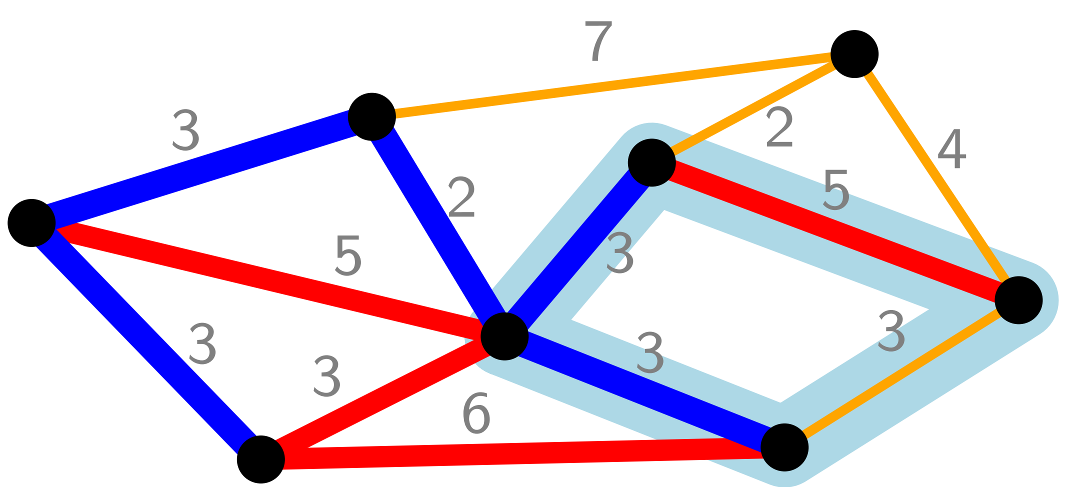
Färbe alle Kanten des Graphen:

- blau: Kante aus MSB
- rot: Kante nicht aus MSB
- ungefärbt: Noch nicht entschieden

Verwende 2 Regeln:

Blaue Regel:
Wähle Schnitt, den keine blaue Kante kreuzt
Färbe leichte Kante blau

Rote Regel:
Wähle Kreis ohne rote Kante
Färbe größte ungefärbte rot



Allgemeiner Greedy Algorithmus

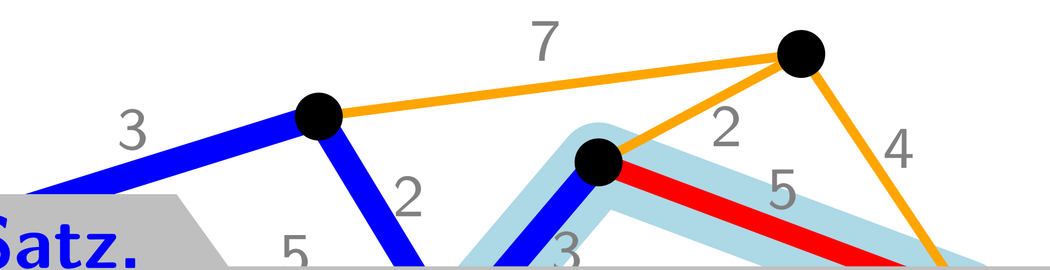
Färbe alle Kanten des Graphen:

- blau: Kante aus MSB
- rot: Kante nicht aus MSB
- ungefärbt: Noch nicht entschieden

Verwende 2 Regeln:

Blaue Regel:
Wähle Schnitt, den keine blaue Kante kreuzt
Färbe leichte Kante blau

Rote Regel:
Wähle Kreis ohne rote Kante
Färbe größte ungefärbte rot



Satz.

1. Die Regeln färben alle Kanten.
2. Die blauen Kanten ergeben einen minimalen Spannbaum

Allgemeiner Greedy Algorithmus

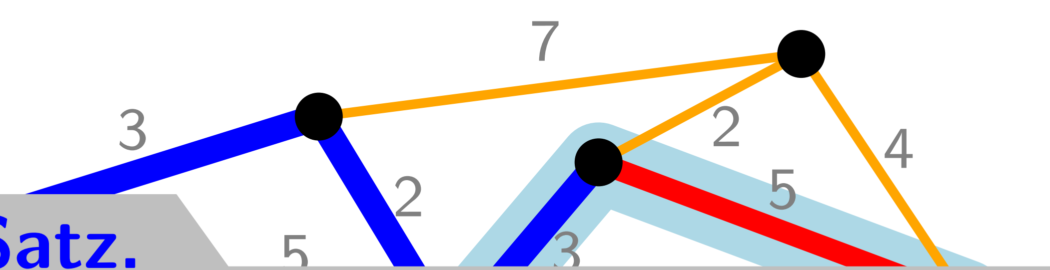
Färbe alle Kanten des Graphen:

- blau: Kante aus MSB
- rot: Kante nicht aus MSB
- ungefärbt: Noch nicht entschieden

Verwende 2 Regeln:

Blaue Regel:
Wähle Schnitt, den keine blaue Kante kreuzt
Färbe leichte Kante blau

Rote Regel:
Wähle Kreis ohne rote Kante
Färbe größte ungefärbte rot



Satz.

1. Die Regeln färben alle Kanten.
2. Die blauen Kanten ergeben einen minimalen Spannbaum

Beweis der blauen Regel

Farbinvariante (FI): Es gibt einen MSB $T = (V, E')$:

Beweis der blauen Regel

Farbinvariante (FI): Es gibt einen MSB $T = (V, E')$:

- T enthält alle blauen Kanten

Beweis der blauen Regel

Farbinvariante (FI): Es gibt einen MSB $T = (V, E')$:

- T enthält alle blauen Kanten
- T enthält keine rote Kante

Beweis der blauen Regel

Farbinvariante (FI): Es gibt einen MSB $T = (V, E')$:

- T enthält alle blauen Kanten
- T enthält keine rote Kante

Satz. Die blaue Regel erhält die Farbinvariante

Beweis der blauen Regel

Farbinvariante (FI): Es gibt einen MSB $T = (V, E')$:

- T enthält alle blauen Kanten
- T enthält keine rote Kante

Satz. Die blaue Regel erhält die Farbinvariante

Beweis.

Beweis der blauen Regel

Farbinvariante (FI): Es gibt einen MSB $T = (V, E')$:

- T enthält alle blauen Kanten
- T enthält keine rote Kante

Satz. Die blaue Regel erhält die Farbinvariante

Beweis. Alle Kanten ungefärbt $\Rightarrow$ jeder MSB *bezeugt* FI

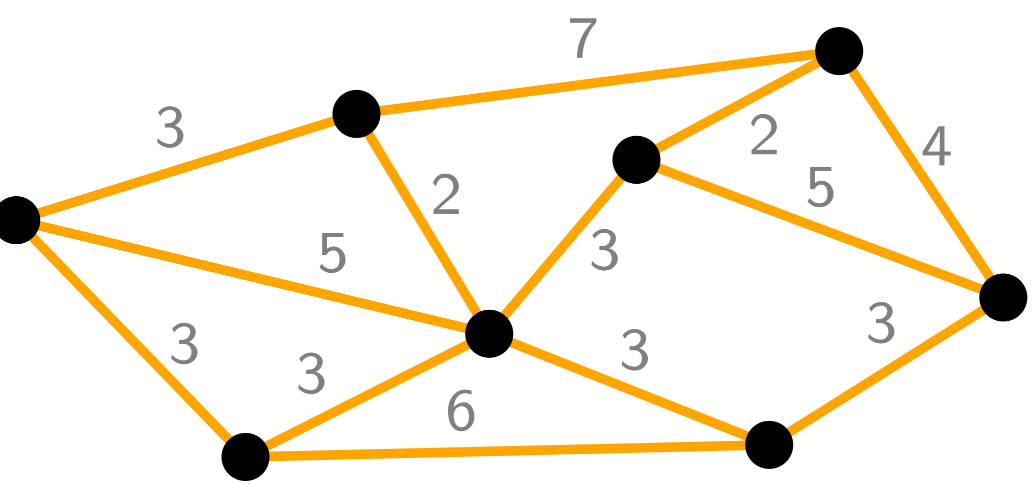
Beweis der blauen Regel

Farbinvariante (FI): Es gibt einen MSB $T = (V, E')$:

- T enthält alle blauen Kanten
- T enthält keine rote Kante

Satz. Die blaue Regel erhält die Farbinvariante

Beweis. Alle Kanten ungefärbt $\Rightarrow$ jeder MSB *bezeugt* FI



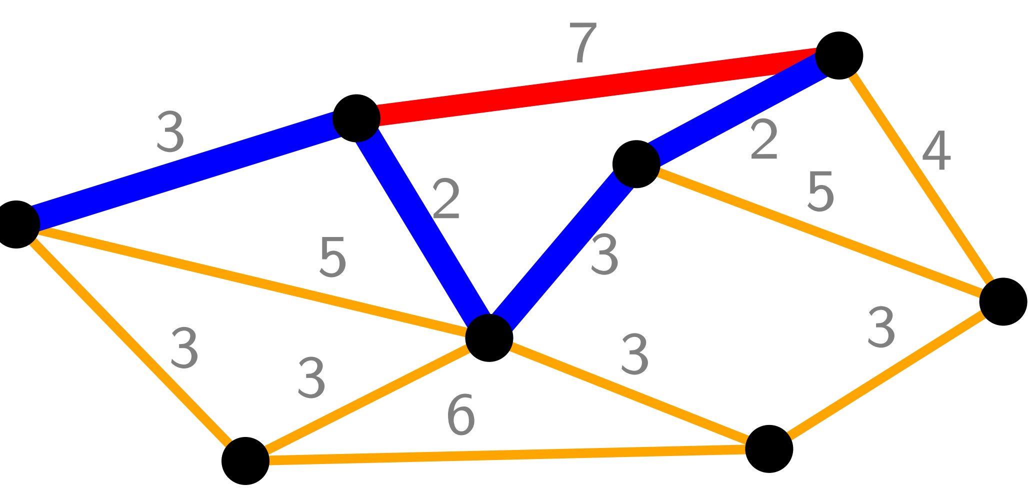
Beweis der blauen Regel

Farbinvariante (FI): Es gibt einen MSB $T = (V, E')$:

- T enthält alle blauen Kanten
- T enthält keine rote Kante

Satz. Die blaue Regel erhält die Farbinvariante

Beweis. Alle Kanten ungefärbt $\Rightarrow$ jeder MSB *bezeugt* FI



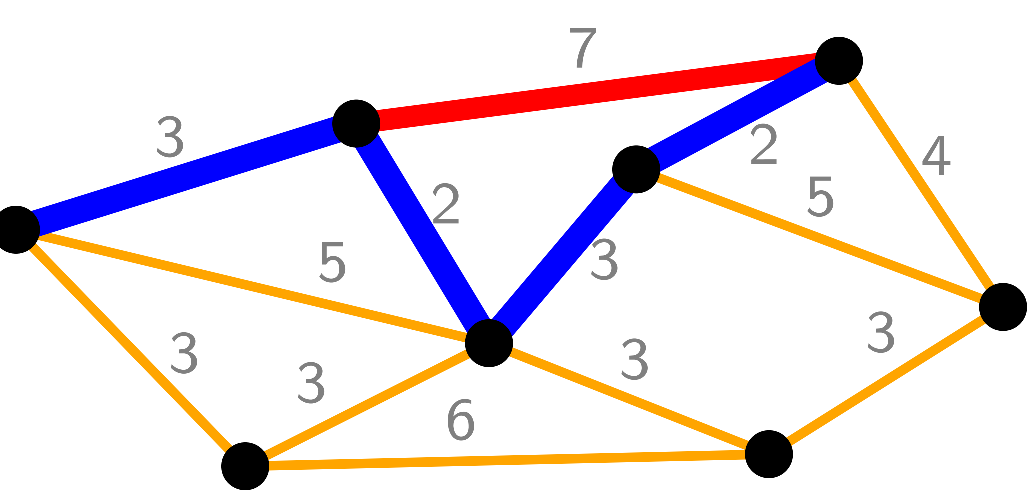
Beweis der blauen Regel

Farbinvariante (FI): Es gibt einen MSB $T = (V, E')$:

- T enthält alle blauen Kanten
- T enthält keine rote Kante

Satz. Die blaue Regel erhält die Farbinvariante

Beweis. Alle Kanten ungefärbt $\Rightarrow$ jeder MSB *bezeugt* FI
Sei T min. Spannbaum, der FI bezeugt



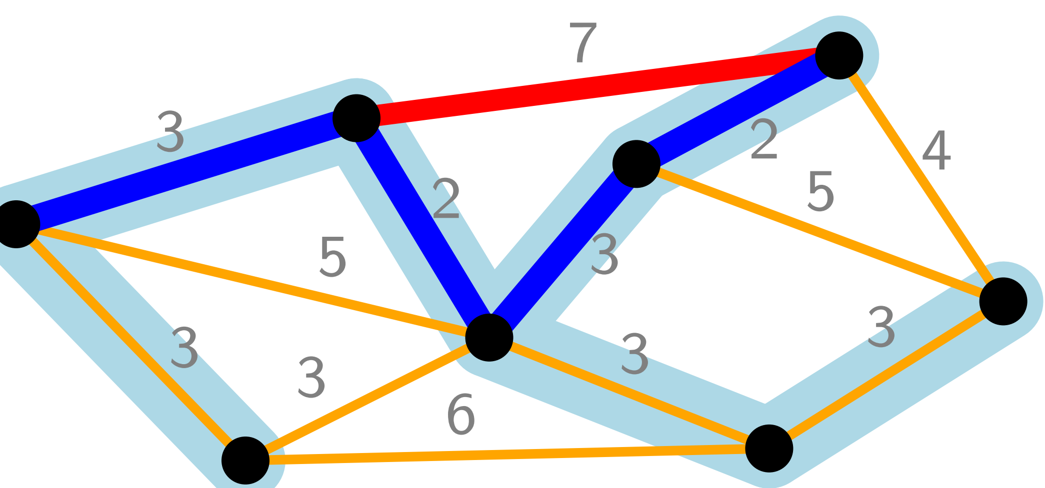
Beweis der blauen Regel

Farbinvariante (FI): Es gibt einen MSB $T = (V, E')$:

- T enthält alle blauen Kanten
- T enthält keine rote Kante

Satz. Die blaue Regel erhält die Farbinvariante

Beweis. Alle Kanten ungefärbt $\Rightarrow$ jeder MSB *bezeugt* FI
Sei T min. Spannbaum, der FI bezeugt



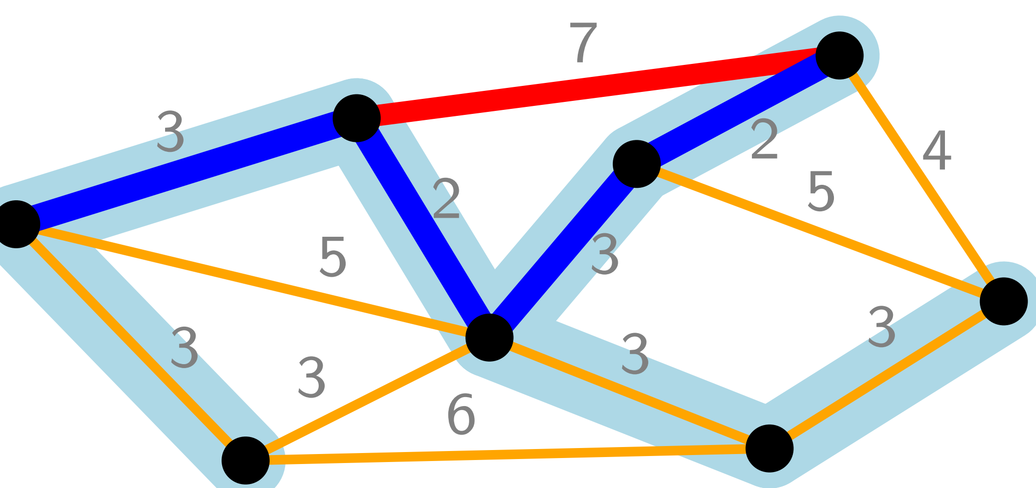
Beweis der blauen Regel

Farbinvariante (FI): Es gibt einen MSB $T = (V, E')$:

- T enthält alle blauen Kanten
- T enthält keine rote Kante

Satz. Die blaue Regel erhält die Farbinvariante

Beweis. Alle Kanten ungefärbt $\Rightarrow$ jeder MSB *bezeugt* FI
Sei T min. Spannbaum, der FI bezeugt
Sei $uv \in E$ von blauer Regel ausgewählte Kante



Beweis der blauen Regel

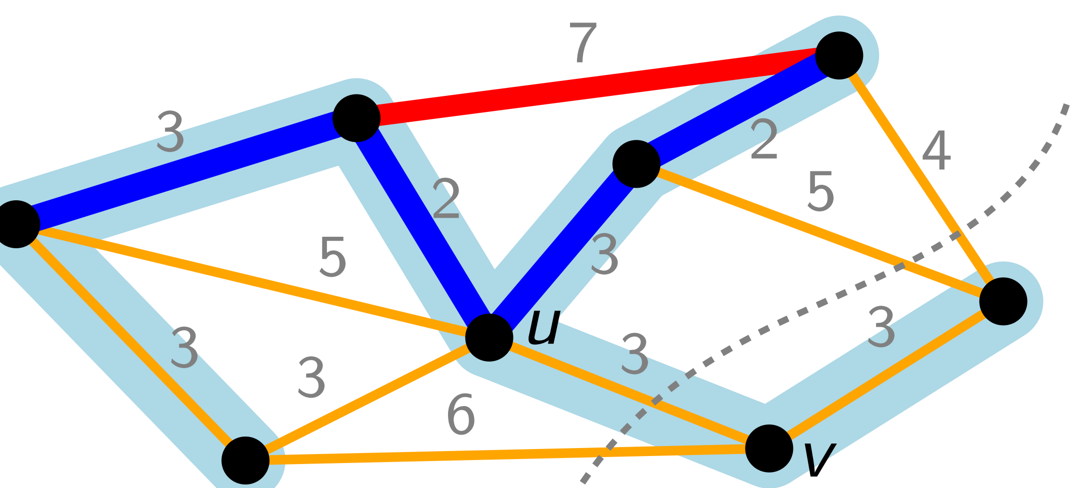
Farbinvariante (FI): Es gibt einen MSB $T = (V, E')$:

- T enthält alle blauen Kanten
- T enthält keine rote Kante

Satz. Die blaue Regel erhält die Farbinvariante

Beweis. Alle Kanten ungefärbt $\Rightarrow$ jeder MSB *bezeugt* FI
Sei T min. Spannbaum, der FI bezeugt
Sei $uv \in E$ von blauer Regel ausgewählte Kante

1. Fall: $uv \in E'$



Beweis der blauen Regel

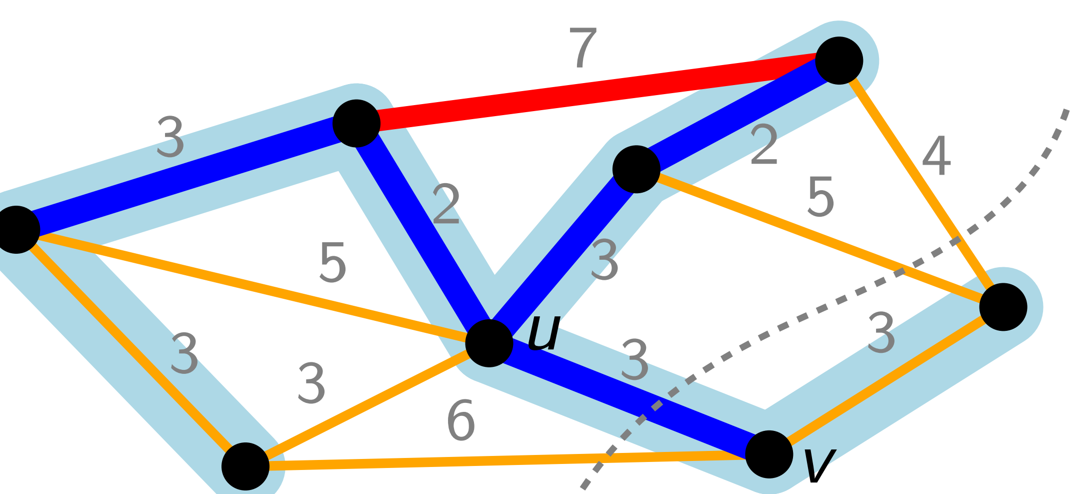
Farbinvariante (FI): Es gibt einen MSB $T = (V, E')$:

- T enthält alle blauen Kanten
- T enthält keine rote Kante

Satz. Die blaue Regel erhält die Farbinvariante

Beweis. Alle Kanten ungefärbt $\Rightarrow$ jeder MSB *bezeugt* FI
Sei T min. Spannbaum, der FI bezeugt
Sei $uv \in E$ von blauer Regel ausgewählte Kante

1. Fall: $uv \in E'$



Beweis der blauen Regel

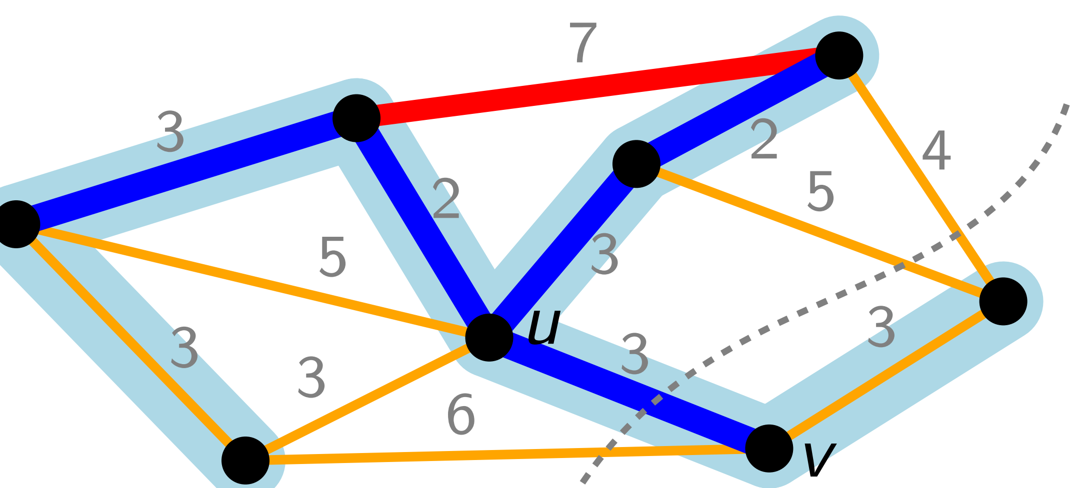
Farbinvariante (FI): Es gibt einen MSB $T = (V, E')$:

- T enthält alle blauen Kanten
- T enthält keine rote Kante

Satz. Die blaue Regel erhält die Farbinvariante

Beweis. Alle Kanten ungefärbt $\Rightarrow$ jeder MSB *bezeugt* FI
Sei T min. Spannbaum, der FI bezeugt
Sei $uv \in E$ von blauer Regel ausgewählte Kante

1. Fall: $uv \in E' \Rightarrow$ FI bleibt erhalten



Beweis der blauen Regel

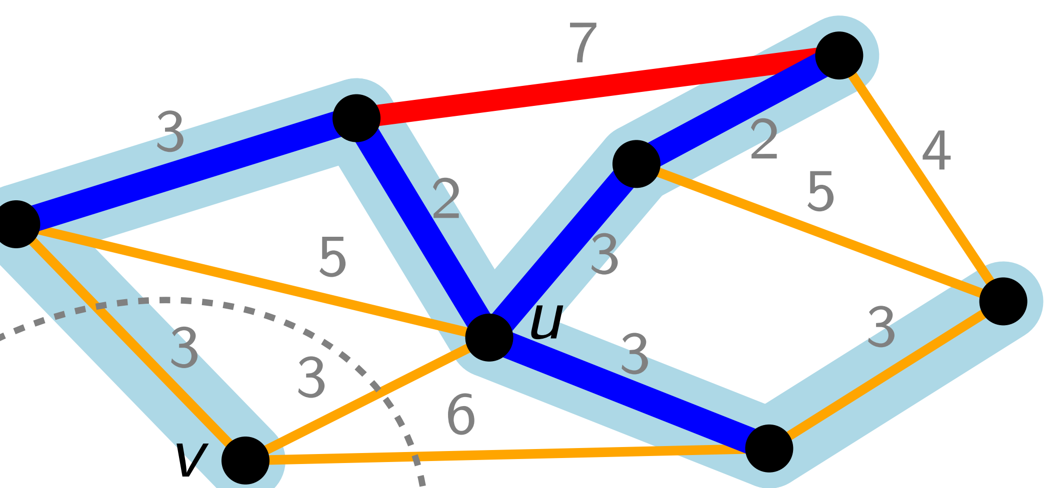
Farbinvariante (FI): Es gibt einen MSB $T = (V, E')$:

- T enthält alle blauen Kanten
- T enthält keine rote Kante

Satz. Die blaue Regel erhält die Farbinvariante

Beweis. Alle Kanten ungefärbt $\Rightarrow$ jeder MSB *bezeugt* FI
Sei T min. Spannbaum, der FI bezeugt
Sei $uv \in E$ von blauer Regel ausgewählte Kante

1. Fall: $uv \in E' \Rightarrow$ FI bleibt erhalten
2. Fall: $uv \notin E'$



Beweis der blauen Regel

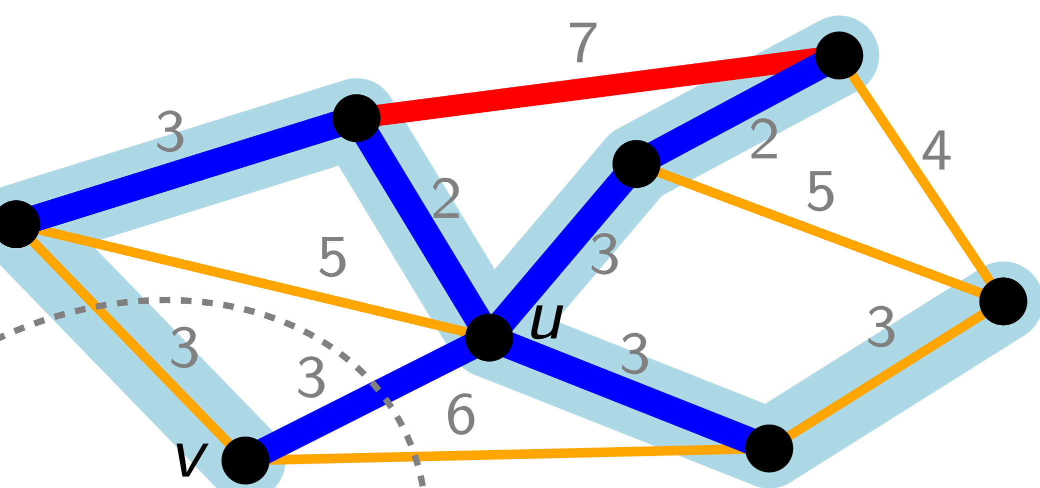
Farbinvariante (FI): Es gibt einen MSB $T = (V, E')$:

- T enthält alle blauen Kanten
- T enthält keine rote Kante

Satz. Die blaue Regel erhält die Farbinvariante

Beweis. Alle Kanten ungefärbt $\Rightarrow$ jeder MSB *bezeugt* FI
Sei T min. Spannbaum, der FI bezeugt
Sei $uv \in E$ von blauer Regel ausgewählte Kante

1. Fall: $uv \in E' \Rightarrow$ FI bleibt erhalten
2. Fall: $uv \notin E'$



Beweis der blauen Regel

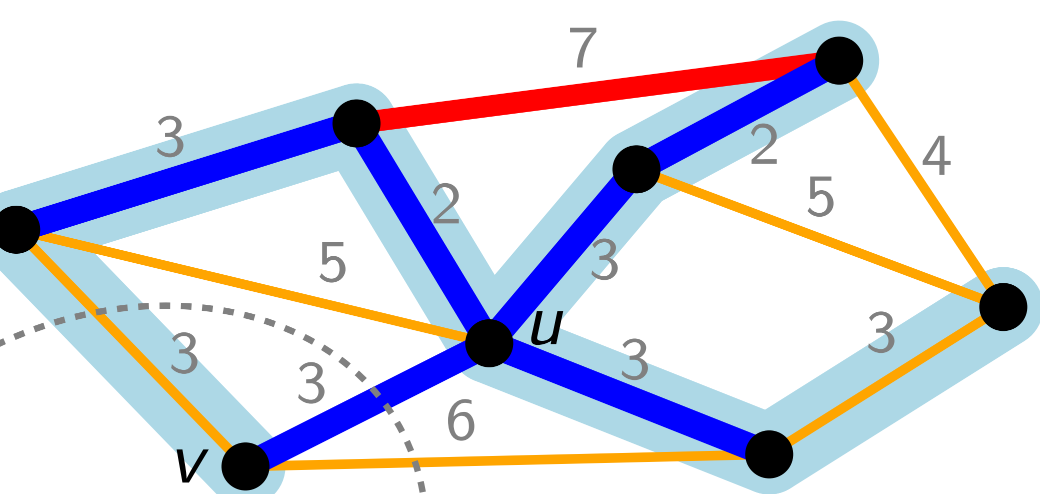
Farbinvariante (FI): Es gibt einen MSB $T = (V, E')$:

- T enthält alle blauen Kanten
- T enthält keine rote Kante

Satz. Die blaue Regel erhält die Farbinvariante

Beweis. Alle Kanten ungefärbt $\Rightarrow$ jeder MSB *bezeugt* FI
Sei T min. Spannbaum, der FI bezeugt
Sei $uv \in E$ von blauer Regel ausgewählte Kante

1. Fall: $uv \in E' \Rightarrow$ FI bleibt erhalten
2. Fall: $uv \notin E' \Rightarrow$ es gibt Pfad p von u nach v in T



Beweis der blauen Regel

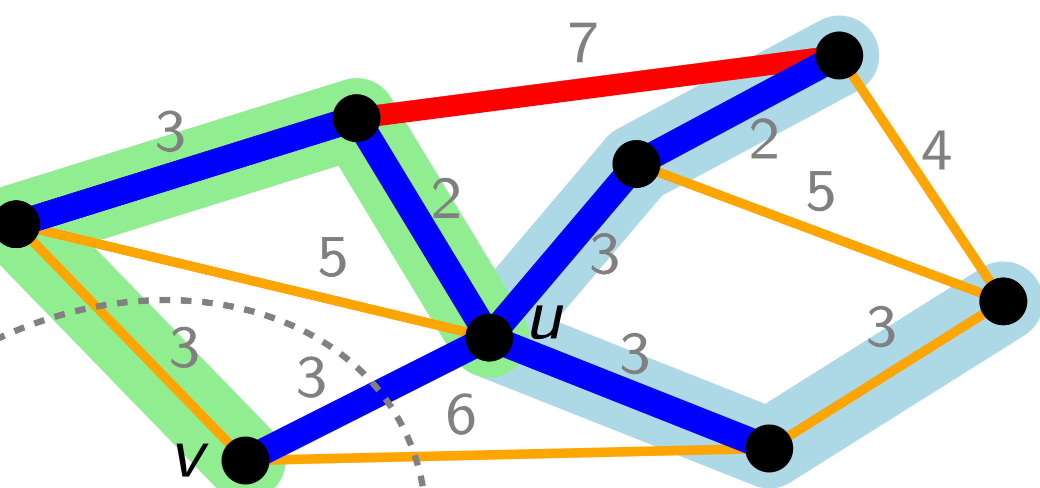
Farbinvariante (FI): Es gibt einen MSB $T = (V, E')$:

- T enthält alle blauen Kanten
- T enthält keine rote Kante

Satz. Die blaue Regel erhält die Farbinvariante

Beweis. Alle Kanten ungefärbt $\Rightarrow$ jeder MSB *bezeugt* FI
Sei T min. Spannbaum, der FI bezeugt
Sei $uv \in E$ von blauer Regel ausgewählte Kante

1. Fall: $uv \in E' \Rightarrow$ FI bleibt erhalten
2. Fall: $uv \notin E' \Rightarrow$ es gibt Pfad p von u nach v in T



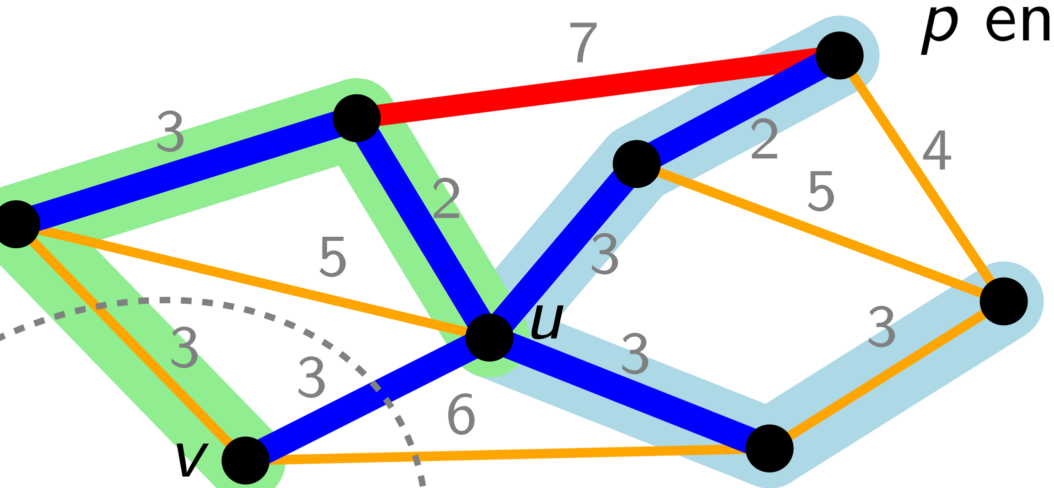
Beweis der blauen Regel

Farbinvariante (FI): Es gibt einen MSB $T = (V, E')$:

- T enthält alle blauen Kanten
- T enthält keine rote Kante

Satz. Die blaue Regel erhält die Farbinvariante

Beweis. Alle Kanten ungefärbt $\Rightarrow$ jeder MSB *bezeugt* FI
Sei T min. Spannbaum, der FI bezeugt
Sei $uv \in E$ von blauer Regel ausgewählte Kante
1. Fall: $uv \in E' \Rightarrow$ FI bleibt erhalten
2. Fall: $uv \notin E' \Rightarrow$ es gibt Pfad p von u nach v in T
p enthält Kante xy, die Schnitt kreuzt



Beweis der blauen Regel

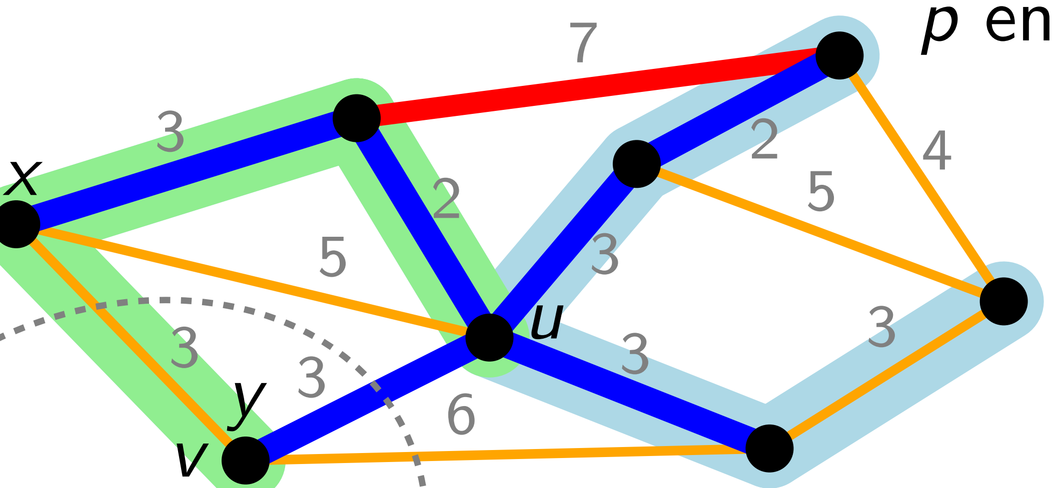
Farbinvariante (FI): Es gibt einen MSB $T = (V, E')$:

- T enthält alle blauen Kanten
- T enthält keine rote Kante

Satz. Die blaue Regel erhält die Farbinvariante

Beweis. Alle Kanten ungefärbt $\Rightarrow$ jeder MSB *bezeugt* FI
Sei T min. Spannbaum, der FI bezeugt
Sei $uv \in E$ von blauer Regel ausgewählte Kante

1. Fall: $uv \in E' \Rightarrow$ FI bleibt erhalten
2. Fall: $uv \notin E' \Rightarrow$ es gibt Pfad p von u nach v in T
p enthält Kante xy, die Schnitt kreuzt



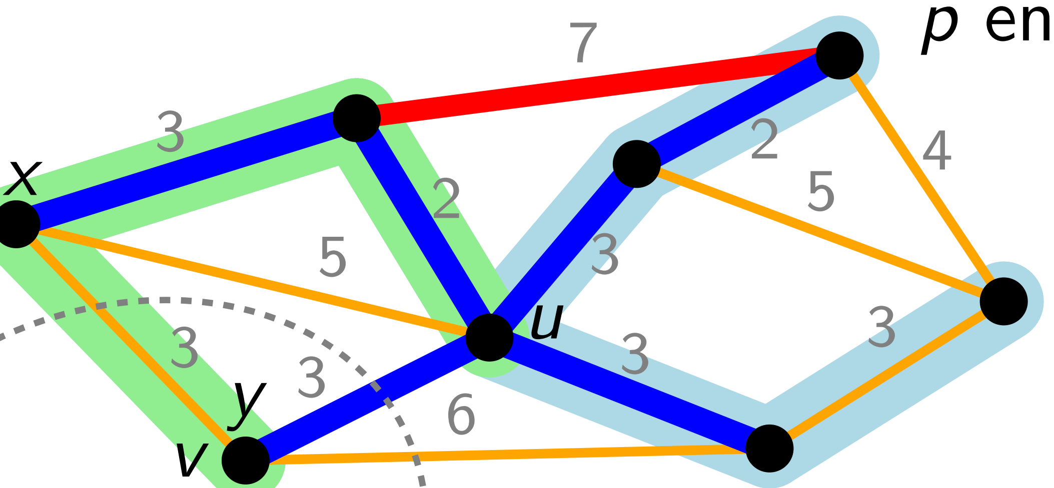
Beweis der blauen Regel

Farbinvariante (FI): Es gibt einen MSB $T = (V, E')$:
- T enthält alle blauen Kanten
- T enthält keine rote Kante

Satz. Die blaue Regel erhält die Farbinvariante

Beweis. Blaue Regel:
Wähle Schnitt, den keine blaue Kante kreuzt
Färbe leichte Kante blau

1. Fall: $uv \in E' \Rightarrow$ FI bleibt erhalten
2. Fall: $uv \notin E' \Rightarrow$ es gibt Pfad p von u nach v in T
   p enthält Kante xy, die Schnitt kreuzt



Beweis der blauen Regel

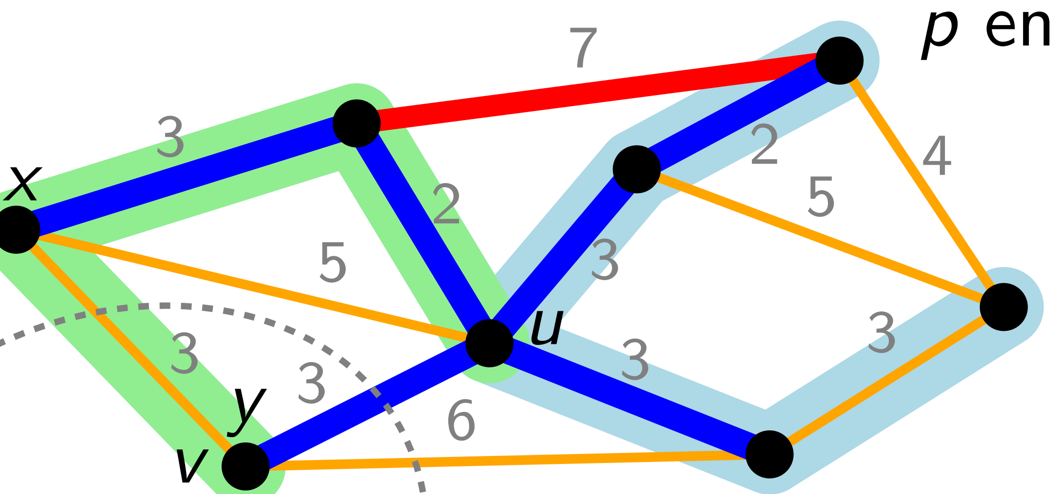
Farbinvariante (FI): Es gibt einen MSB $T = (V, E')$:

- T enthält alle blauen Kanten
- T enthält keine rote Kante

Satz. Die blaue Regel erhält die Farbinvariante

Beweis. Blaue Regel:
Wähle Schnitt, den keine blaue Kante kreuzt
Färbe leichte Kante blau

1. Fall: $uv \in E' \Rightarrow$ FI bleibt erhalten
2. Fall: $uv \notin E' \Rightarrow$ es gibt Pfad p von u nach v in T
p enthält Kante xy, die Schnitt kreuzt



Beweis der blauen Regel

Farbinvariante (FI): Es gibt einen MSB $T = (V, E')$:

- T enthält alle blauen Kanten
- T enthält keine rote Kante

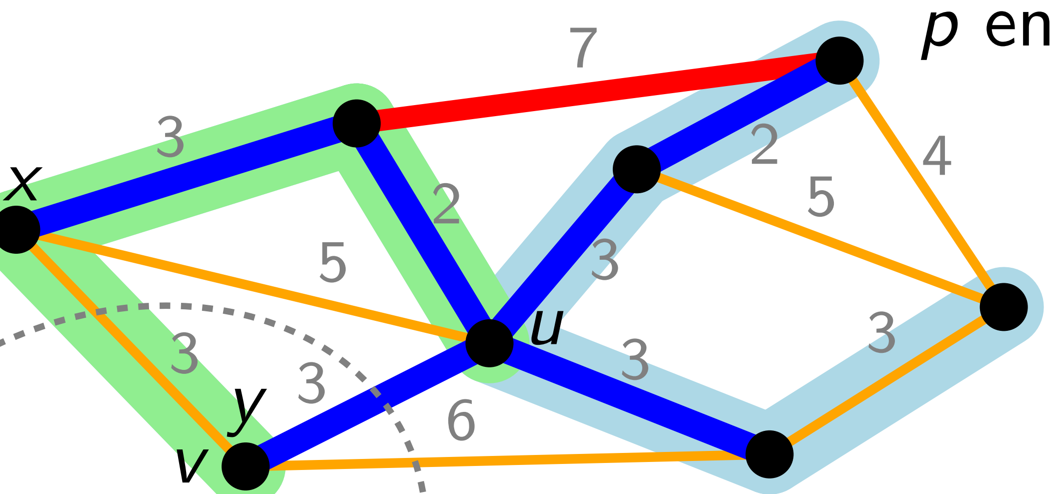
Satz. Die blaue Regel erhält die Farbinvariante

Beweis. Blaue Regel:
Wähle Schnitt, den keine blaue Kante kreuzt
Färbe leichte Kante blau

1. Fall: $uv \in E' \Rightarrow$ FI bleibt erhalten
2. Fall: $uv \notin E' \Rightarrow$ es gibt Pfad p von u nach v in T

p enthält Kante xy, die Schnitt kreuzt

xy ist ungefärbt



Beweis der blauen Regel

Farbinvariante (FI): Es gibt einen MSB $T = (V, E')$:

- T enthält alle blauen Kanten
- T enthält keine rote Kante

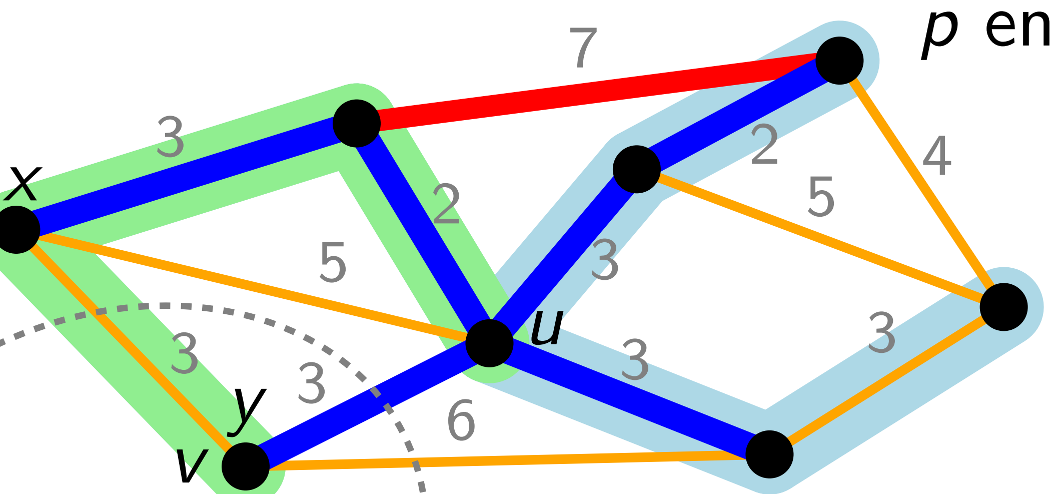
Satz. Die blaue Regel erhält die Farbinvariante

Beweis. Blaue Regel:
Wähle Schnitt, den keine blaue Kante kreuzt
Färbe leichte Kante blau

1. Fall: $uv \in E' \Rightarrow$ FI bleibt erhalten
2. Fall: $uv \notin E' \Rightarrow$ es gibt Pfad p von u nach v in T

p enthält Kante xy, die Schnitt kreuzt

xy ist ungefärbt



Beweis der blauen Regel

Farbinvariante (FI): Es gibt einen MSB $T = (V, E')$:

- T enthält alle blauen Kanten
- T enthält keine rote Kante

Satz. Die blaue Regel erhält die Farbinvariante

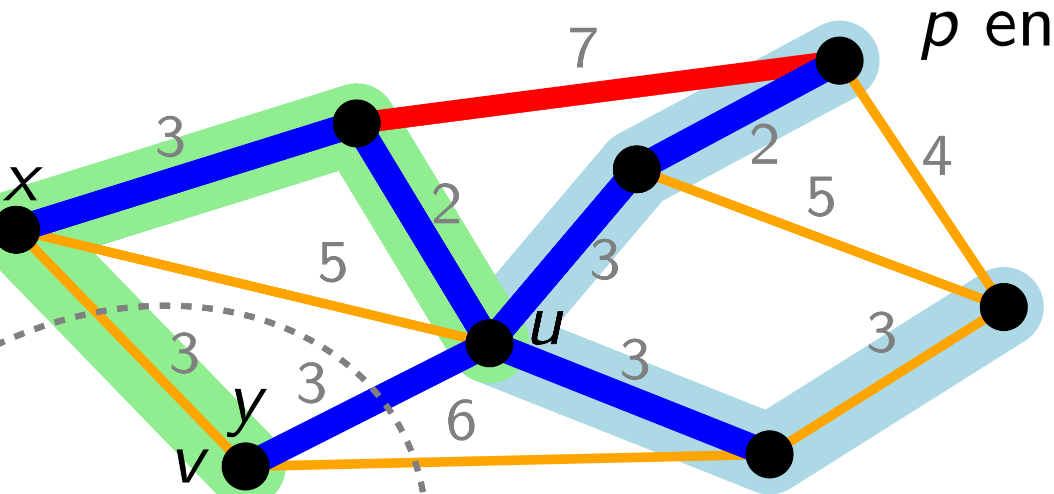
Beweis. Blaue Regel:
Wähle Schnitt, den keine blaue Kante kreuzt
Färbe leichte Kante blau

1. Fall: $uv \in E' \Rightarrow$ FI bleibt erhalten
2. Fall: $uv \notin E' \Rightarrow$ es gibt Pfad p von u nach v in T

p enthält Kante xy, die Schnitt kreuzt

xy ist ungefärbt

$w(xy) \geq w(uv)$



Beweis der blauen Regel

Farbinvariante (FI): Es gibt einen MSB $T = (V, E')$:

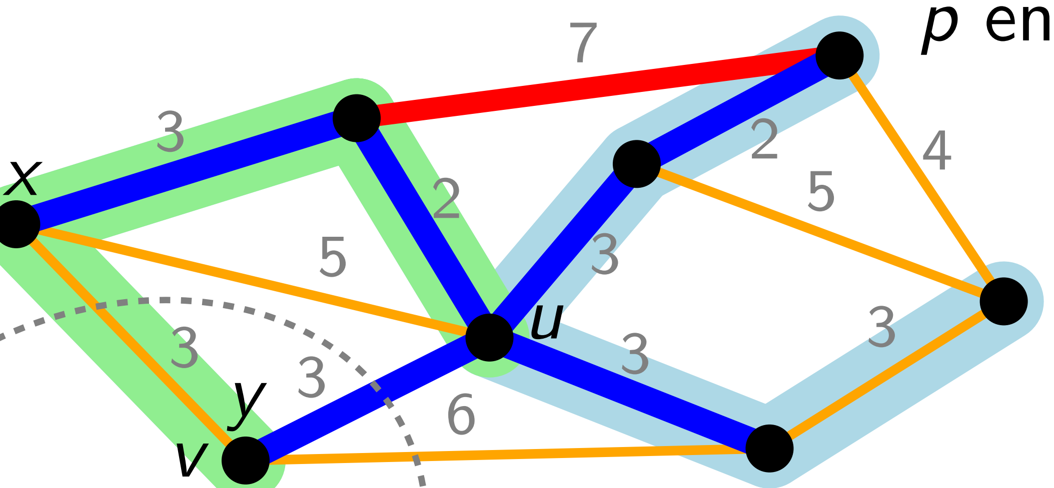
- T enthält alle blauen Kanten
- T enthält keine rote Kante

Satz. Die blaue Regel erhält die Farbinvariante

Beweis. Alle Kanten ungefärbt $\Rightarrow$ jeder MSB *bezeugt* FI
Sei T min. Spannbaum, der FI bezeugt
Sei $uv \in E$ von blauer Regel ausgewählte Kante

1. Fall: $uv \in E' \Rightarrow$ FI bleibt erhalten
2. Fall: $uv \notin E' \Rightarrow$ es gibt Pfad p von u nach v in T

p enthält Kante xy, die Schnitt kreuzt



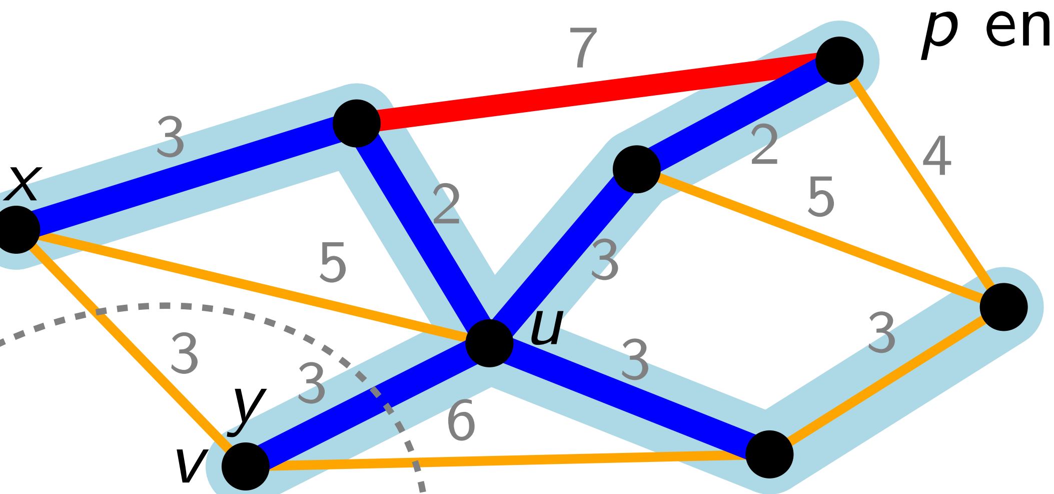
xy ist ungefärbt
$w(xy) \geq w(uv)$
Wähle $E'' = E' \cup \{uv\} \setminus \{xy\}$

Beweis der blauen Regel

Farbinvariante (FI): Es gibt einen MSB $T = (V, E')$:
- T enthält alle blauen Kanten
- T enthält keine rote Kante

Satz. Die blaue Regel erhält die Farbinvariante

Beweis. Alle Kanten ungefärbt $\Rightarrow$ jeder MSB *bezeugt* FI
Sei T min. Spannbaum, der FI bezeugt
Sei $uv \in E$ von blauer Regel ausgewählte Kante
1. Fall: $uv \in E' \Rightarrow$ FI bleibt erhalten
2. Fall: $uv \notin E' \Rightarrow$ es gibt Pfad p von u nach v in T



p enthält Kante xy, die Schnitt kreuzt
xy ist ungefärbt
$w(xy) \geq w(uv)$
Wähle $E'' = E' \cup \{uv\} \setminus \{xy\}$

Beweis der blauen Regel

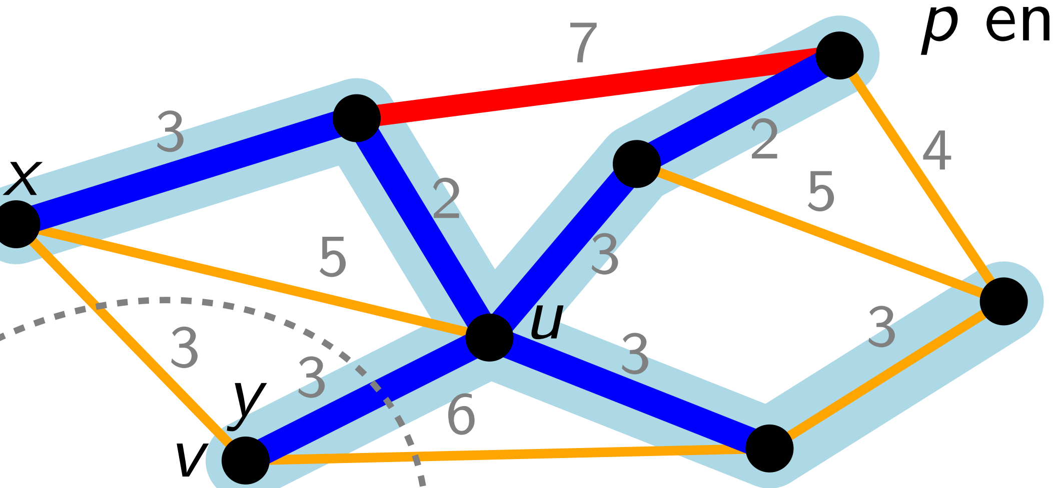
Farbinvariante (FI): Es gibt einen MSB $T = (V, E')$:

- T enthält alle blauen Kanten
- T enthält keine rote Kante

Satz. Die blaue Regel erhält die Farbinvariante

Beweis. Alle Kanten ungefärbt $\Rightarrow$ jeder MSB *bezeugt* FI
Sei T min. Spannbaum, der FI bezeugt
Sei $uv \in E$ von blauer Regel ausgewählte Kante

1. Fall: $uv \in E' \Rightarrow$ FI bleibt erhalten
2. Fall: $uv \notin E' \Rightarrow$ es gibt Pfad p von u nach v in T



p enthält Kante xy, die Schnitt kreuzt
xy ist ungefärbt
$w(xy) \geq w(uv)$
Wähle $E'' = E' \cup \{uv\} \setminus \{xy\}$
$\Rightarrow$ $T' = (V, E'')$ ist MSB,
der FI bezeugt

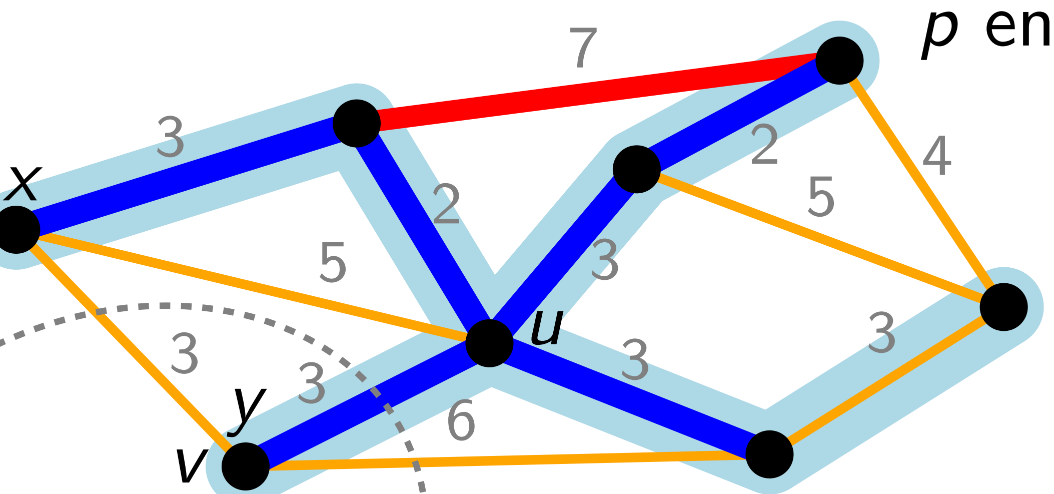
Beweis der blauen Regel

Farbinvariante (FI): Es gibt einen MSB $T = (V, E')$:
- T enthält alle blauen Kanten
- T enthält keine rote Kante

Satz. Die blaue Regel erhält die Farbinvariante

Beweis. Alle Kanten ungefärbt $\Rightarrow$ jeder MSB *bezeugt* FI
Sei T min. Spannbaum, der FI bezeugt
Sei $uv \in E$ von blauer Regel ausgewählte Kante

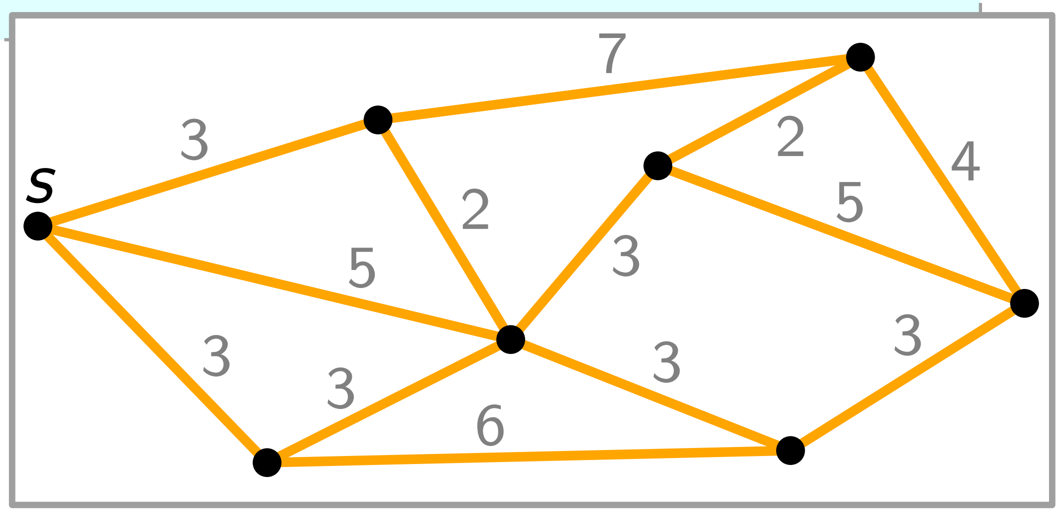
1. Fall: $uv \in E' \Rightarrow$ FI bleibt erhalten
2. Fall: $uv \notin E' \Rightarrow$ es gibt Pfad p von u nach v in T



p enthält Kante xy, die Schnitt kreuzt
xy ist ungefärbt
$w(xy) \geq w(uv)$
Wähle $E'' = E' \cup \{uv\} \setminus \{xy\}$
$\Rightarrow T' = (V, E'')$ ist MSB,
der FI bezeugt $\square$

Der Algorithmus von Jarník-Prim (1930/1957)

Jarník-Prim(WeightedGraph $G = (V, E; w)$, Vertex s)

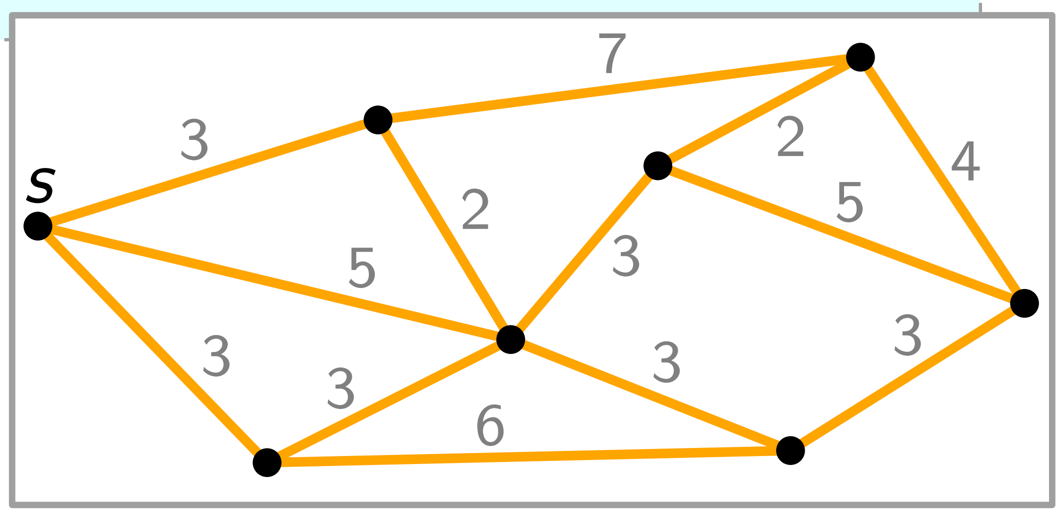


Der Algorithmus von Jarník-Prim (1930/1957)

Jarník-Prim(WeightedGraph $G = (V, E; w)$, Vertex s)

$S = \{s\}$

$E' = \emptyset$



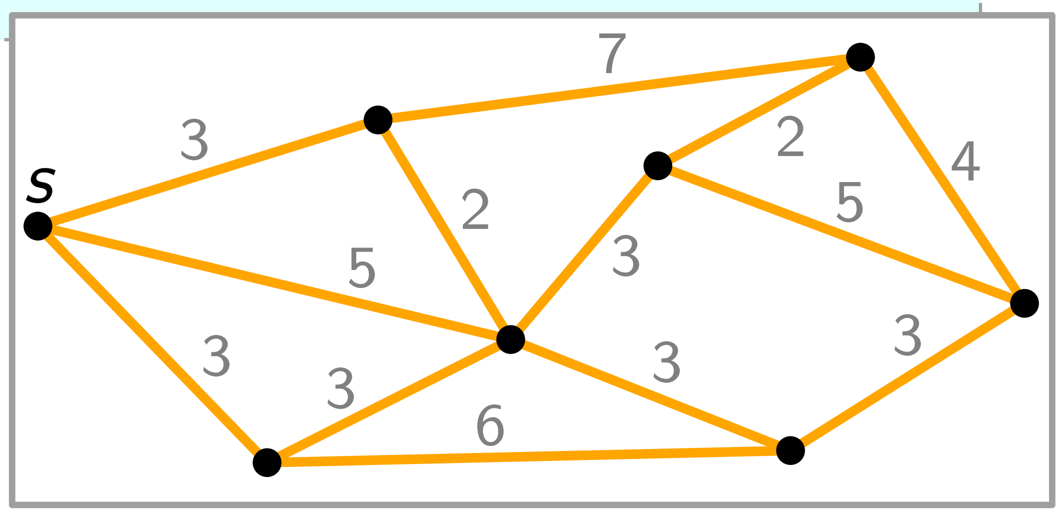
Der Algorithmus von Jarník-Prim (1930/1957)

Jarník-Prim(WeightedGraph $G = (V, E; w)$, Vertex s)

$S = \{s\}$

$E' = \emptyset$

while not $S == V$ **do**



Der Algorithmus von Jarník-Prim (1930/1957)

Jarník-Prim(WeightedGraph $G = (V, E; w)$, Vertex s)

$S = \{s\}$

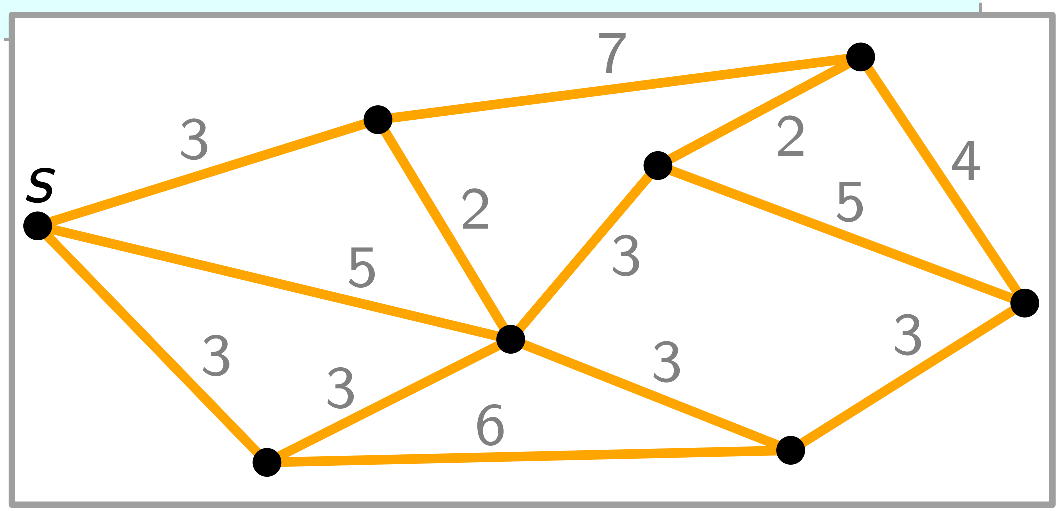
$E' = \emptyset$

while not $S == V$ **do**

Wähle Schnitt $(S, V \setminus S)$

Färbe leichte Kante uv blau $(u \in S, v \in V \setminus S)$

Blaue Regel



Der Algorithmus von Jarník-Prim (1930/1957)

Jarník-Prim(WeightedGraph $G = (V, E; w)$, Vertex s)

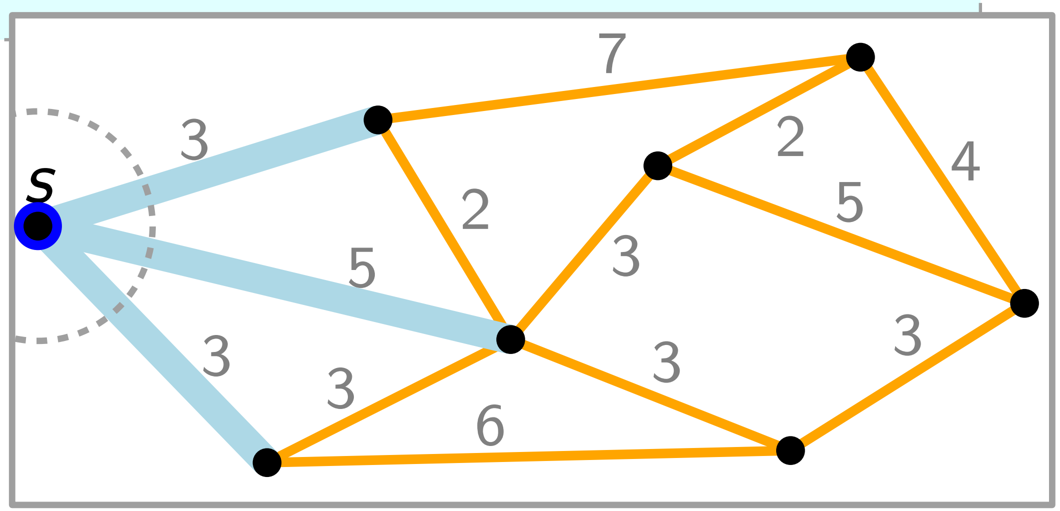
$S = \{s\}$

$E' = \emptyset$

while not $S == V$ **do**

Wähle Schnitt $(S, V \setminus S)$ — Blaue Regel

Färbe leichte Kante uv blau $(u \in S, v \in V \setminus S)$



Der Algorithmus von Jarník-Prim (1930/1957)

Jarník-Prim(WeightedGraph $G = (V, E; w)$, Vertex s)

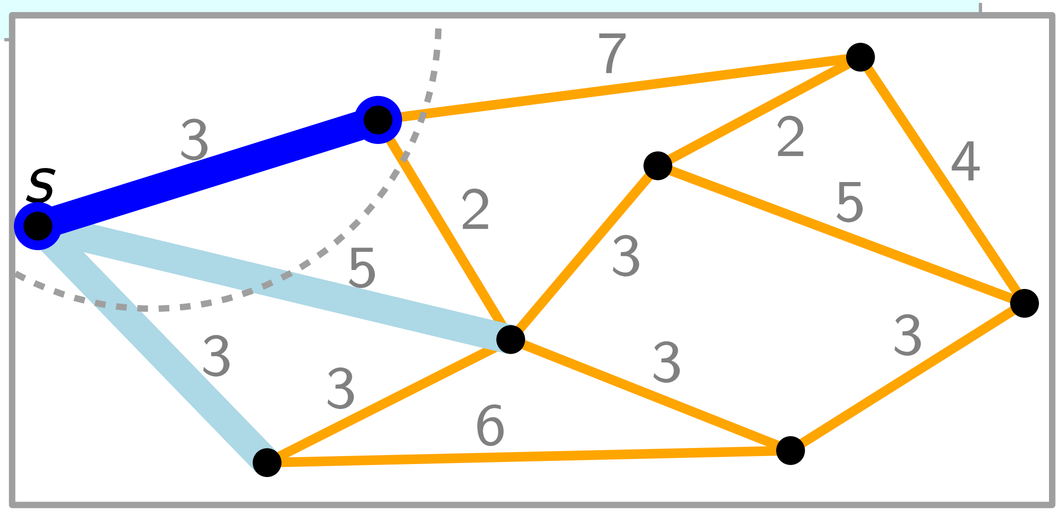
$S = \{s\}$

$E' = \emptyset$

while not $S == V$ **do**

Wähle Schnitt $(S, V \setminus S)$ Blaue Regel

Färbe leichte Kante uv blau $(u \in S, v \in V \setminus S)$



Der Algorithmus von Jarník-Prim (1930/1957)

Jarník-Prim(WeightedGraph $G = (V, E; w)$, Vertex s)

$S = \{s\}$

$E' = \emptyset$

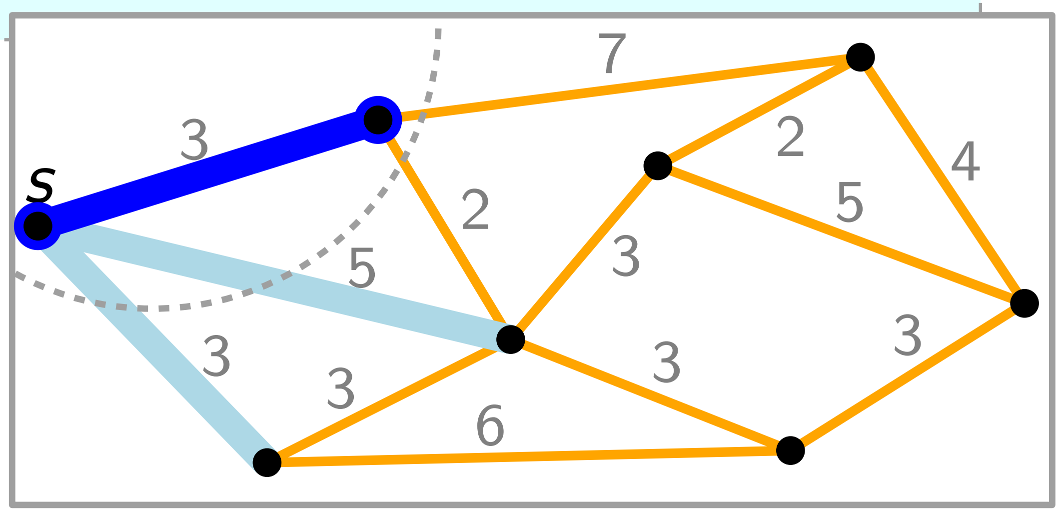
while not $S == V$ **do**

Wähle Schnitt $(S, V \setminus S)$ Blaue Regel

Färbe leichte Kante uv blau $(u \in S, v \in V \setminus S)$

$S = S \cup \{v\}$

$E' = E' \cup \{uv\}$



Der Algorithmus von Jarník-Prim (1930/1957)

Jarník-Prim(WeightedGraph $G = (V, E; w)$, Vertex s)

$S = \{s\}$

$E' = \emptyset$

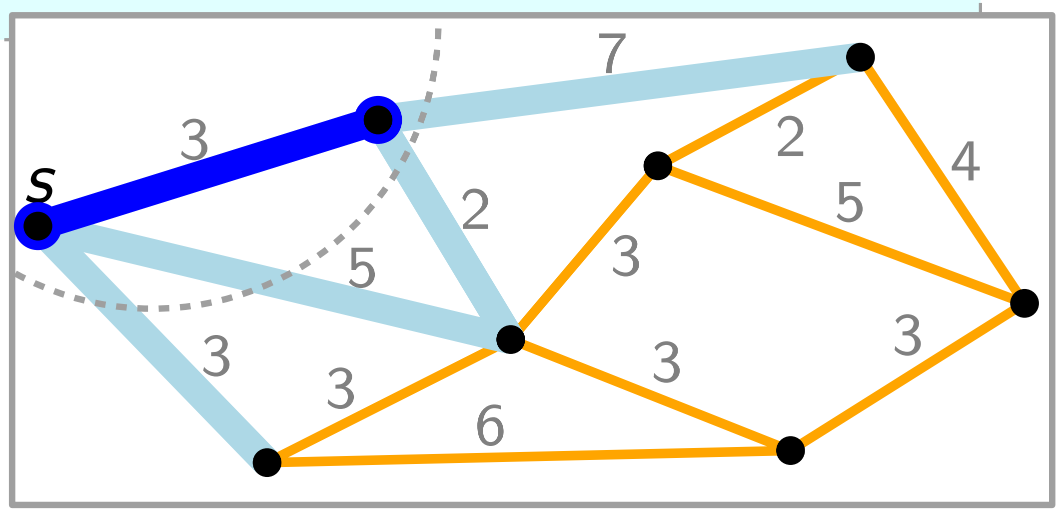
while not $S == V$ **do**

Wähle Schnitt $(S, V \setminus S)$ Blaue Regel

Färbe leichte Kante uv blau $(u \in S, v \in V \setminus S)$

$S = S \cup \{v\}$

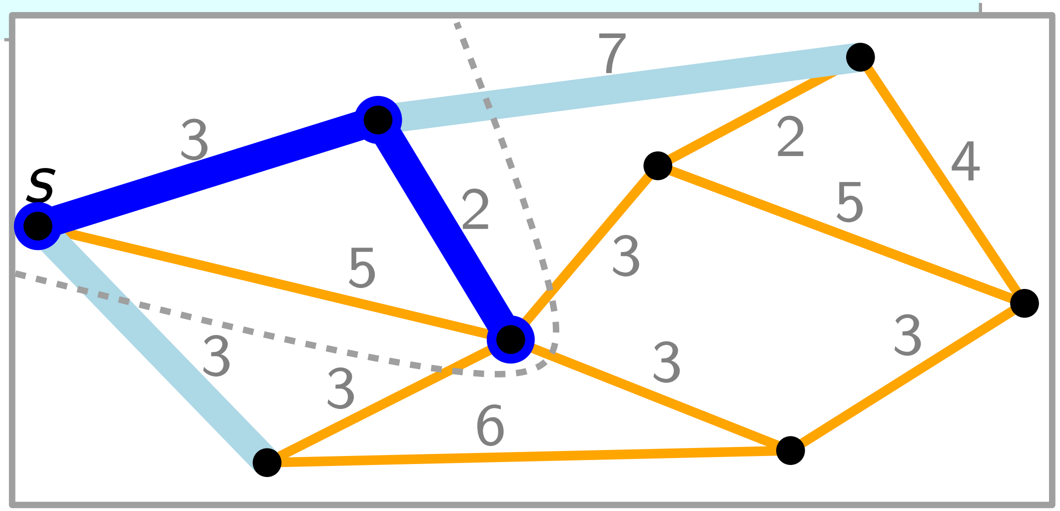
$E' = E' \cup \{uv\}$



Der Algorithmus von Jarník-Prim (1930/1957)

Jarník-Prim(WeightedGraph $G = (V, E; w)$, Vertex s)

$S = \{s\}$
$E' = \emptyset$
while not $S == V$ **do**
    Wähle Schnitt $(S, V \setminus S)$ — Blaue Regel
    Färbe leichte Kante uv blau $(u \in S, v \in V \setminus S)$
    $S = S \cup \{v\}$
    $E' = E' \cup \{uv\}$



Der Algorithmus von Jarník-Prim (1930/1957)

Jarník-Prim(WeightedGraph $G = (V, E; w)$, Vertex s)

$S = \{s\}$

$E' = \emptyset$

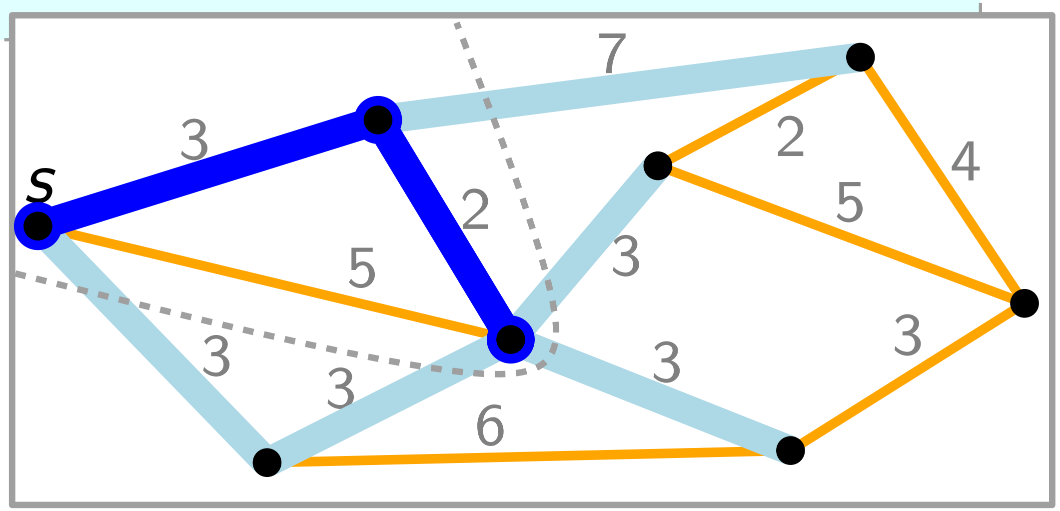
while not $S == V$ **do**

Wähle Schnitt $(S, V \setminus S)$ Blaue Regel

Färbe leichte Kante uv blau $(u \in S, v \in V \setminus S)$

$S = S \cup \{v\}$

$E' = E' \cup \{uv\}$



Der Algorithmus von Jarník-Prim (1930/1957)

Jarník-Prim(WeightedGraph $G = (V, E; w)$, Vertex s)

$S = \{s\}$

$E' = \emptyset$

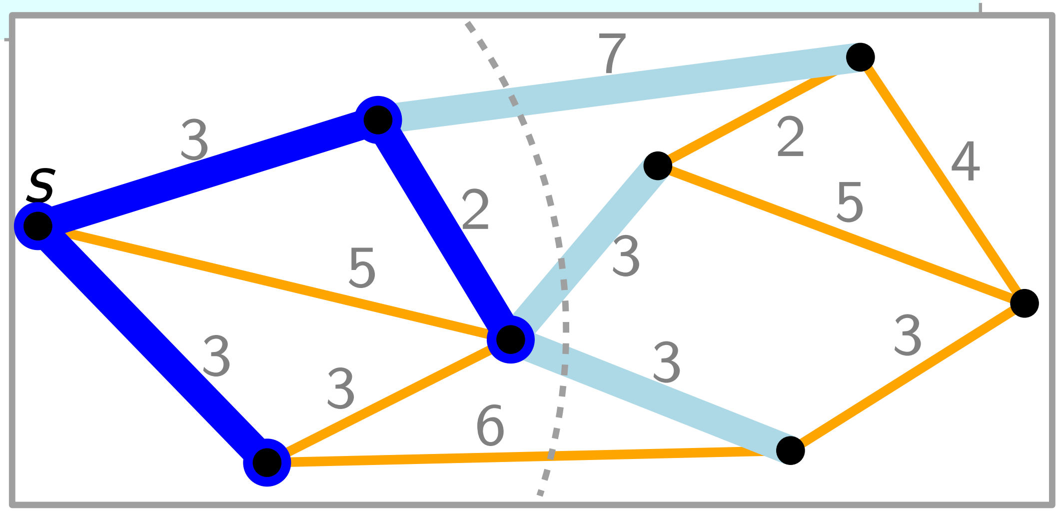
while not $S == V$ **do**

Wähle Schnitt $(S, V \setminus S)$ Blaue Regel

Färbe leichte Kante uv blau ($u \in S, v \in V \setminus S$)

$S = S \cup \{v\}$

$E' = E' \cup \{uv\}$



Der Algorithmus von Jarník-Prim (1930/1957)

Jarník-Prim(WeightedGraph $G = (V, E; w)$, Vertex s)

$S = \{s\}$

$E' = \emptyset$

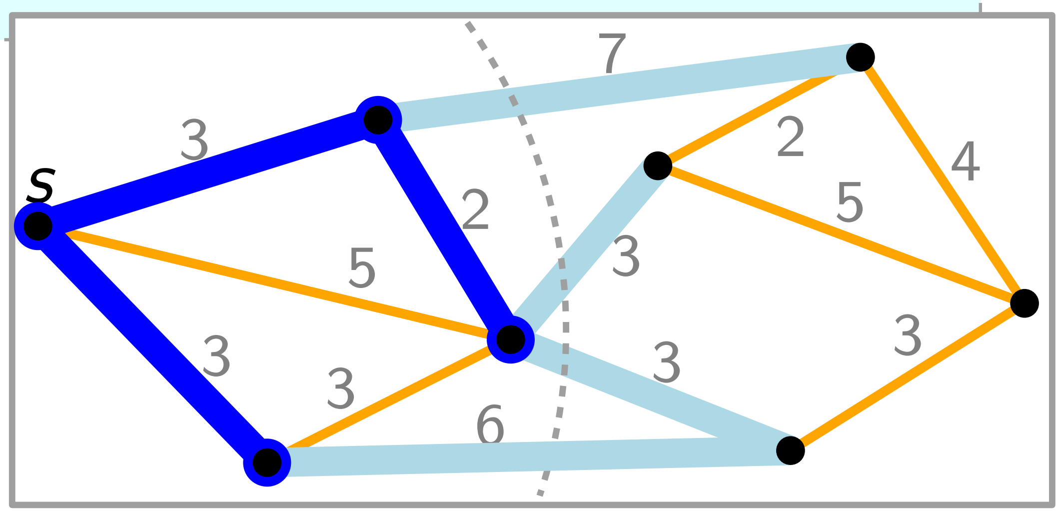
while not $S == V$ **do**

Wähle Schnitt $(S, V \setminus S)$ — Blaue Regel

Färbe leichte Kante uv blau ($u \in S, v \in V \setminus S$)

$S = S \cup \{v\}$

$E' = E' \cup \{uv\}$



Der Algorithmus von Jarník-Prim (1930/1957)

Jarník-Prim(WeightedGraph $G = (V, E; w)$, Vertex s)

$S = \{s\}$

$E' = \emptyset$

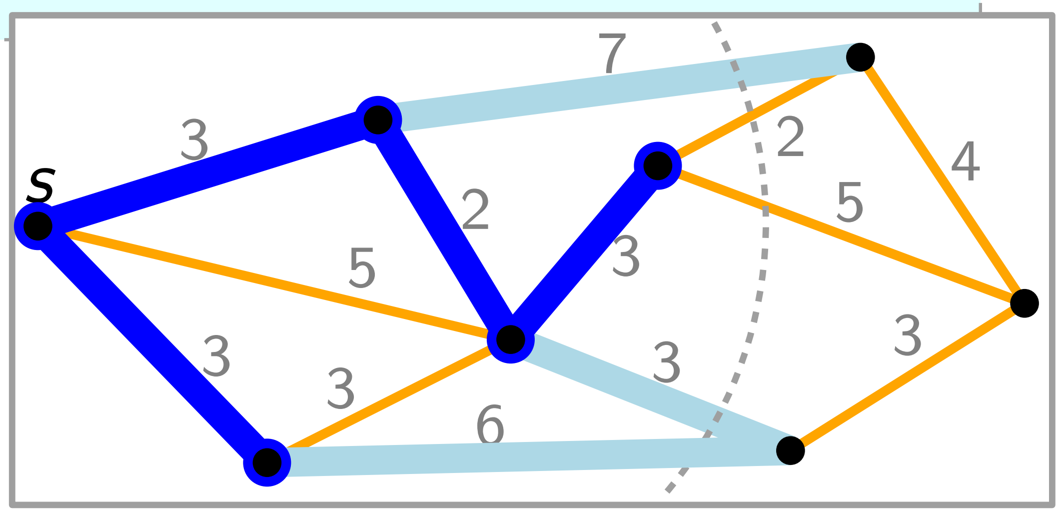
while not $S == V$ **do**

Wähle Schnitt $(S, V \setminus S)$ — Blaue Regel

Färbe leichte Kante uv blau $(u \in S, v \in V \setminus S)$

$S = S \cup \{v\}$

$E' = E' \cup \{uv\}$



Der Algorithmus von Jarník-Prim (1930/1957)

Jarník-Prim(WeightedGraph $G = (V, E; w)$, Vertex s)

$S = \{s\}$

$E' = \emptyset$

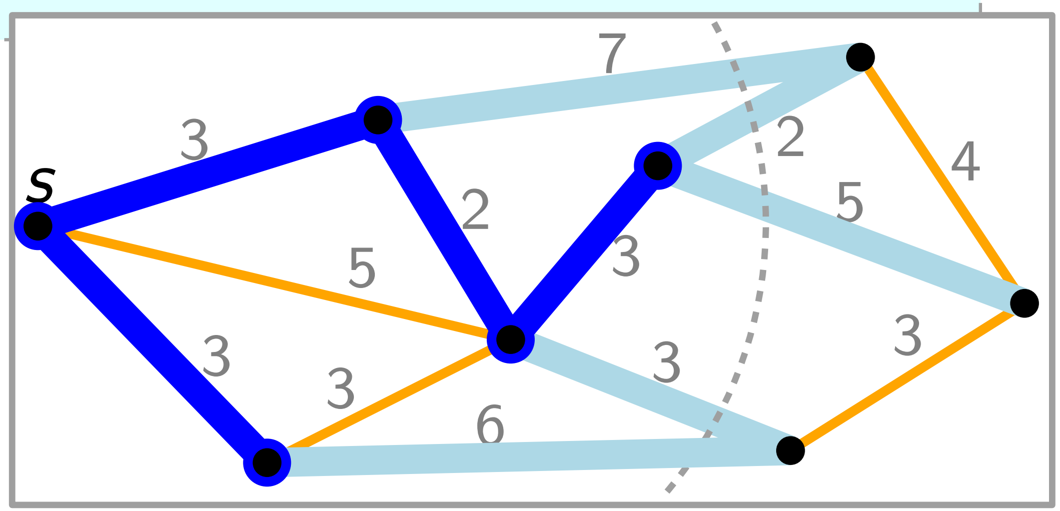
while not $S == V$ **do**

Wähle Schnitt $(S, V \setminus S)$ — Blaue Regel

Färbe leichte Kante uv blau $(u \in S, v \in V \setminus S)$

$S = S \cup \{v\}$

$E' = E' \cup \{uv\}$



Der Algorithmus von Jarník-Prim (1930/1957)

Jarník-Prim(WeightedGraph $G = (V, E; w)$, Vertex s)

$S = \{s\}$

$E' = \emptyset$

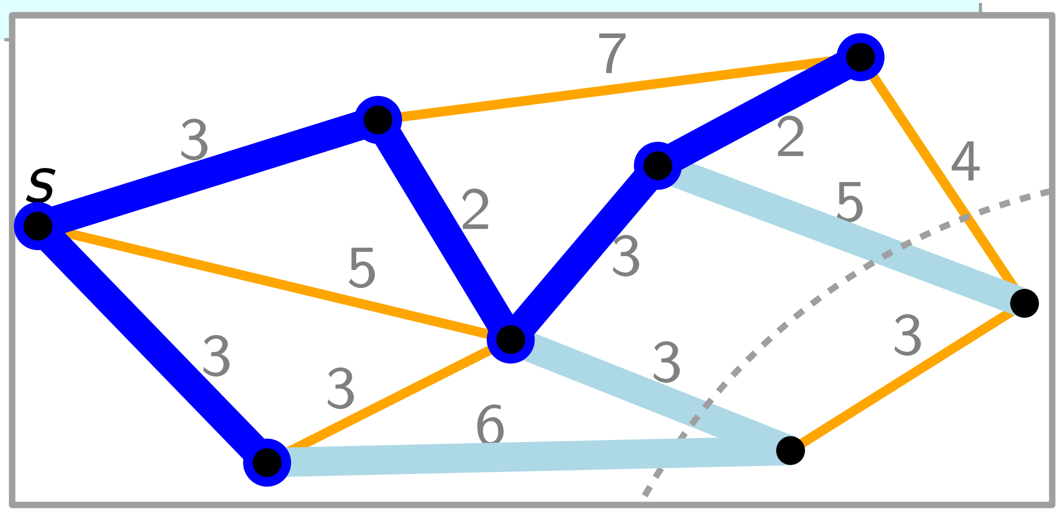
while not $S == V$ **do**

Wähle Schnitt $(S, V \setminus S)$ — Blaue Regel

Färbe leichte Kante uv blau $(u \in S, v \in V \setminus S)$

$S = S \cup \{v\}$

$E' = E' \cup \{uv\}$



Der Algorithmus von Jarník-Prim (1930/1957)

Jarník-Prim(WeightedGraph $G = (V, E; w)$, Vertex s)

$S = \{s\}$

$E' = \emptyset$

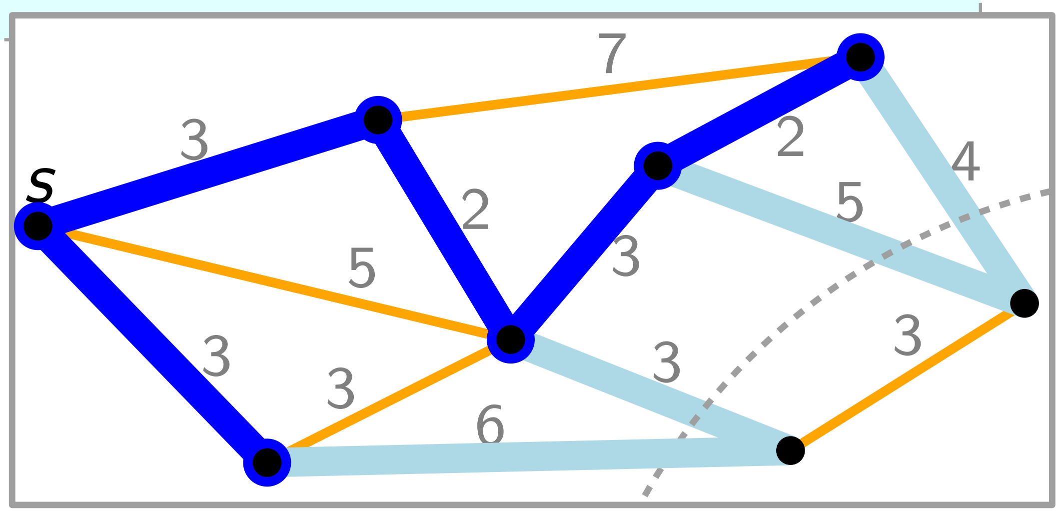
while not $S == V$ **do**

Wähle Schnitt $(S, V \setminus S)$ — Blaue Regel

Färbe leichte Kante uv blau $(u \in S, v \in V \setminus S)$

$S = S \cup \{v\}$

$E' = E' \cup \{uv\}$



Der Algorithmus von Jarník-Prim (1930/1957)

Jarník-Prim(WeightedGraph $G = (V, E; w)$, Vertex s)

$S = \{s\}$

$E' = \emptyset$

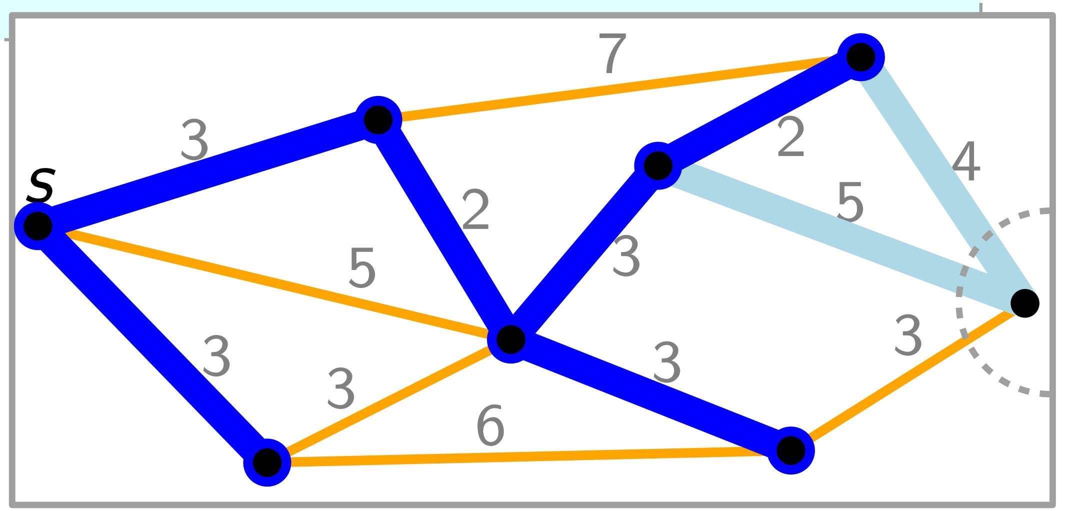
while not $S == V$ **do**

Wähle Schnitt $(S, V \setminus S)$ Blaue Regel

Färbe leichte Kante uv blau ($u \in S, v \in V \setminus S$)

$S = S \cup \{v\}$

$E' = E' \cup \{uv\}$



Der Algorithmus von Jarník-Prim (1930/1957)

Jarník-Prim(WeightedGraph $G = (V, E; w)$, Vertex s)

$S = \{s\}$

$E' = \emptyset$

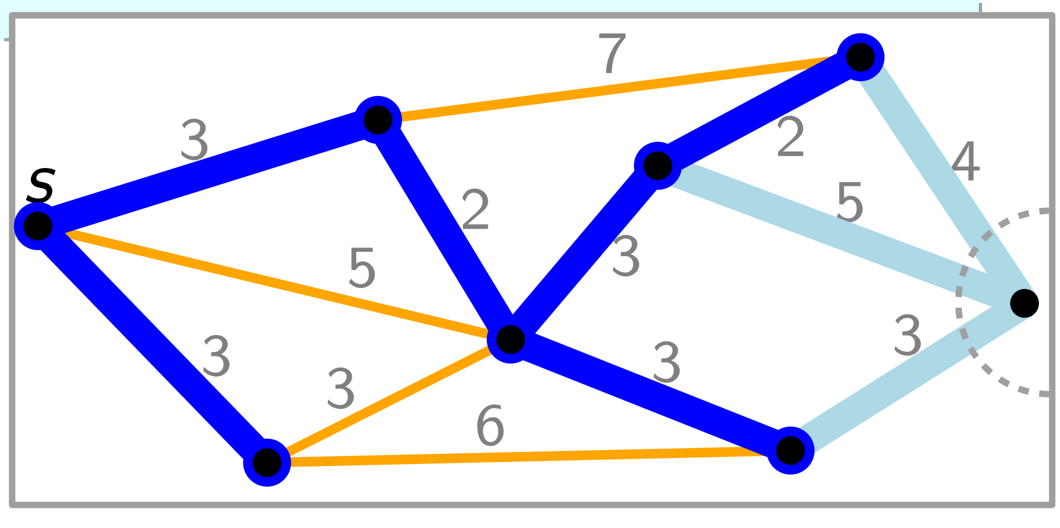
while not $S == V$ **do**

Wähle Schnitt $(S, V \setminus S)$ Blaue Regel

Färbe leichte Kante uv blau ($u \in S, v \in V \setminus S$)

$S = S \cup \{v\}$

$E' = E' \cup \{uv\}$



Der Algorithmus von Jarník-Prim (1930/1957)

Jarník-Prim(WeightedGraph $G = (V, E; w)$, Vertex s)

$S = \{s\}$

$E' = \emptyset$

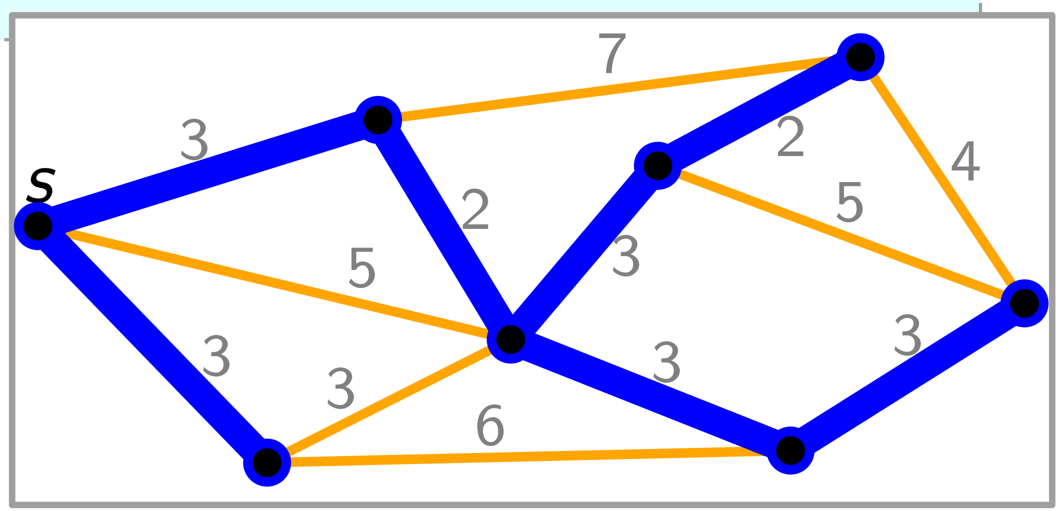
while not $S == V$ **do**

Wähle Schnitt $(S, V \setminus S)$ Blaue Regel

Färbe leichte Kante uv blau $(u \in S, v \in V \setminus S)$

$S = S \cup \{v\}$

$E' = E' \cup \{uv\}$



Der Algorithmus von Jarník-Prim (1930/1957)

Jarník-Prim(WeightedGraph $G = (V, E; w)$, Vertex s)

$S = \{s\}$

$E' = \emptyset$

while not $S == V$ **do**

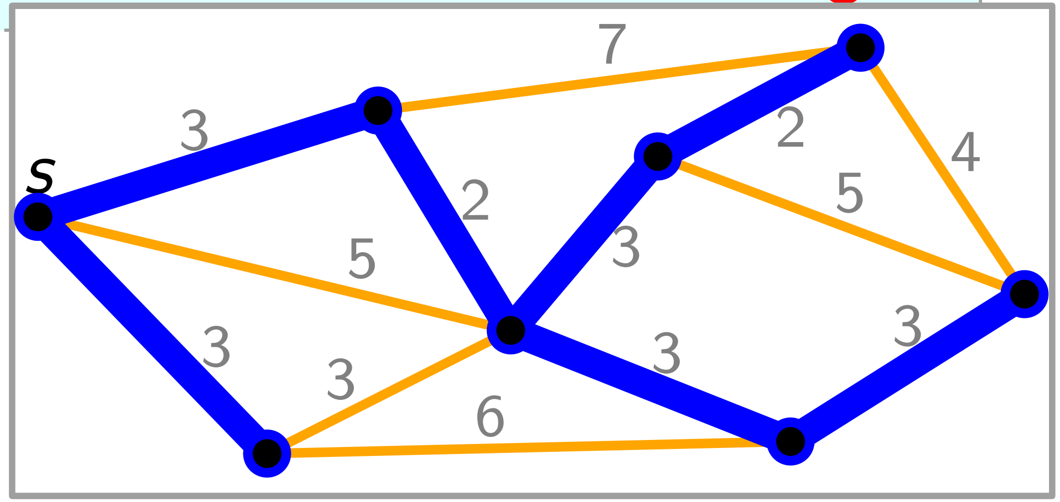
Wähle Schnitt $(S, V \setminus S)$ — Blaue Regel

Färbe leichte Kante uv blau $(u \in S, v \in V \setminus S)$

$S = S \cup \{v\}$

$E' = E' \cup \{uv\}$

Färbe alle anderen Kanten rot — Rote Regel



Der Algorithmus von Jarník-Prim (1930/1957)

Jarník-Prim(WeightedGraph $G = (V, E; w)$, Vertex s)

$S = \{s\}$

$E' = \emptyset$

while not $S == V$ **do**

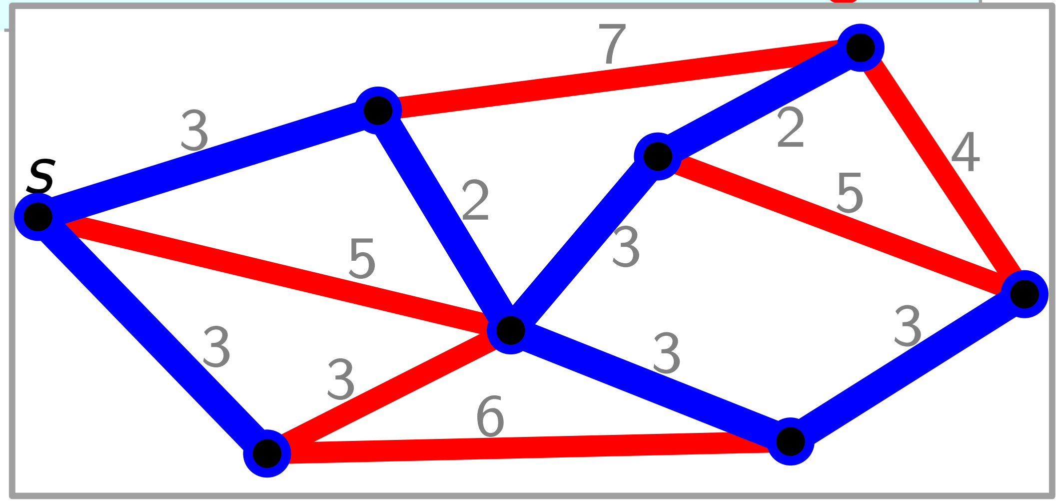
Wähle Schnitt $(S, V \setminus S)$ — Blaue Regel

Färbe leichte Kante uv blau $(u \in S, v \in V \setminus S)$

$S = S \cup \{v\}$

$E' = E' \cup \{uv\}$

Färbe alle anderen Kanten rot — Rote Regel



Der Algorithmus von Jarník-Prim (1930/1957)

Jarník-Prim(WeightedGraph $G = (V, E; w)$, Vertex s)

$S = \{s\}$

$E' = \emptyset$

while not $S == V$ **do**

Wähle Schnitt $(S, V \setminus S)$ — Blaue Regel

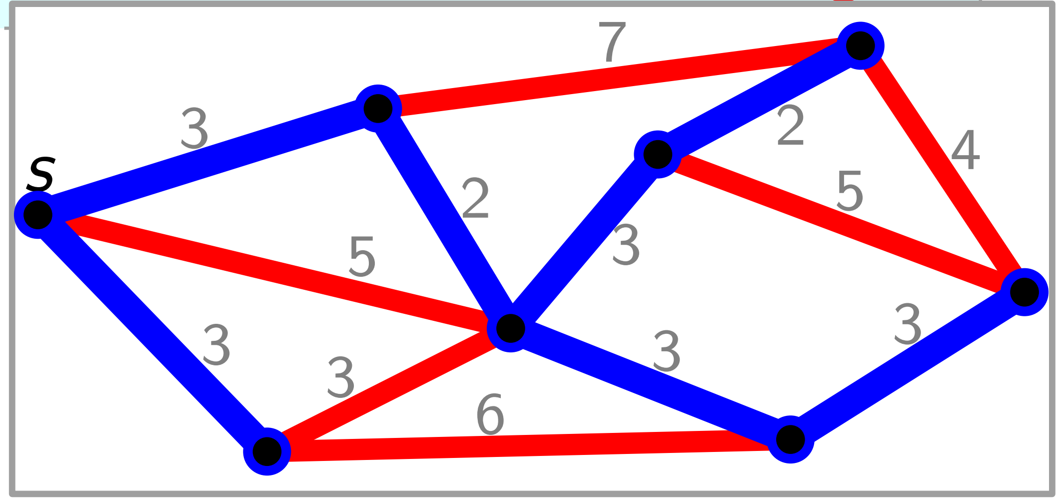
Färbe leichte Kante uv blau $(u \in S, v \in V \setminus S)$

$S = S \cup \{v\}$

$E' = E' \cup \{uv\}$

Färbe alle anderen Kanten rot — Rote Regel

Laufzeit?



Der Algorithmus von Jarník-Prim (1930/1957)

Jarník-Prim(WeightedGraph $G = (V, E; w)$, Vertex s)

$S = \{s\}$

$E' = \emptyset$

while not $S == V$ **do**

Wähle Schnitt $(S, V \setminus S)$ — Blaue Regel

Färbe leichte Kante uv blau $(u \in S, v \in V \setminus S)$

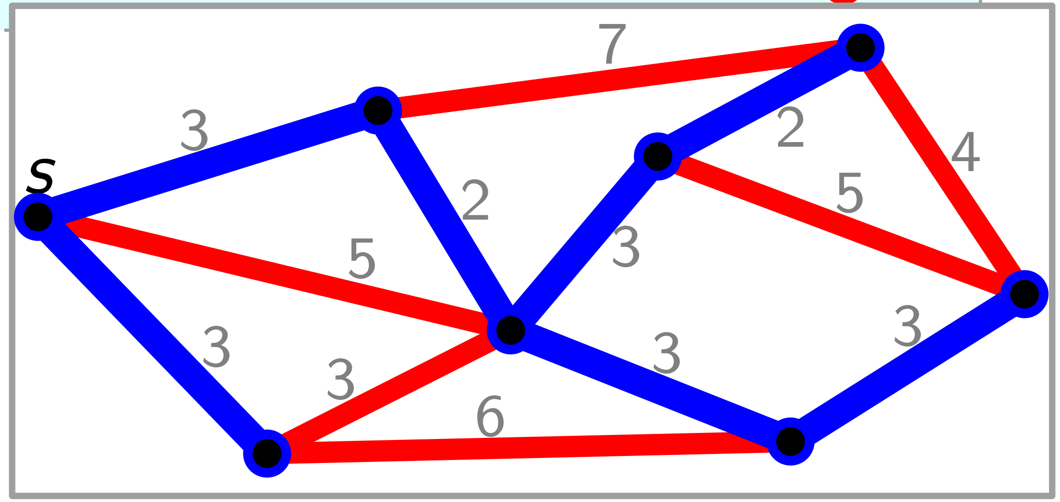
$S = S \cup \{v\}$

$E' = E' \cup \{uv\}$

Färbe alle anderen Kanten rot — Rote Regel

Laufzeit?

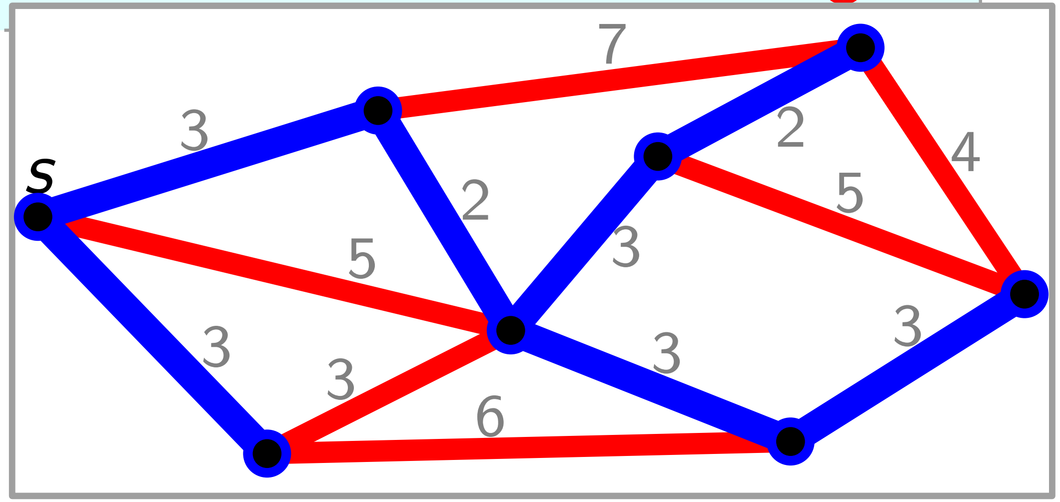
Wie Dijkstra!



Der Algorithmus von Jarník-Prim (1930/1957)

Jarník-Prim(WeightedGraph $G = (V, E; w)$, Vertex s)

$S = \{s\}$
$E' = \emptyset$
while not $S == V$ **do**
  Wähle Schnitt $(S, V \setminus S)$ — Blaue Regel
  Färbe leichte Kante uv blau $(u \in S, v \in V \setminus S)$
  $S = S \cup \{v\}$
  $E' = E' \cup \{uv\}$
Färbe alle anderen Kanten rot — Rote Regel



Laufzeit?

Wie Dijkstra!

$\Rightarrow O((E + V)\log V)$ [Heap/RS-Baum]

Der Algorithmus von Jarník-Prim (1930/1957)

Jarník-Prim(WeightedGraph $G = (V, E; w)$, Vertex s)

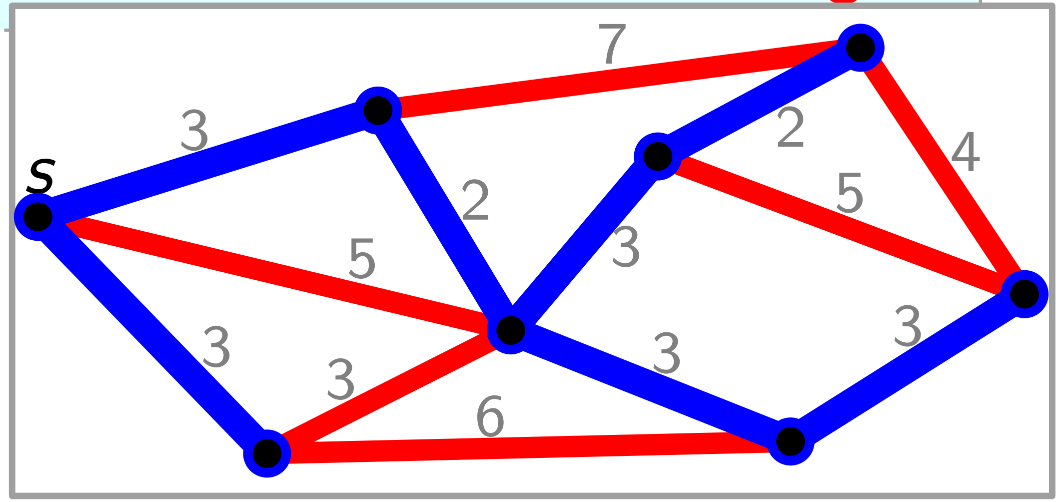
$S = \{s\}$
$E' = \emptyset$
while not $S == V$ **do**
  Wähle Schnitt $(S, V \setminus S)$ — Blaue Regel
  Färbe leichte Kante uv blau $(u \in S, v \in V \setminus S)$
  $S = S \cup \{v\}$
  $E' = E' \cup \{uv\}$
Färbe alle anderen Kanten rot — Rote Regel

Laufzeit?

Wie Dijkstra!

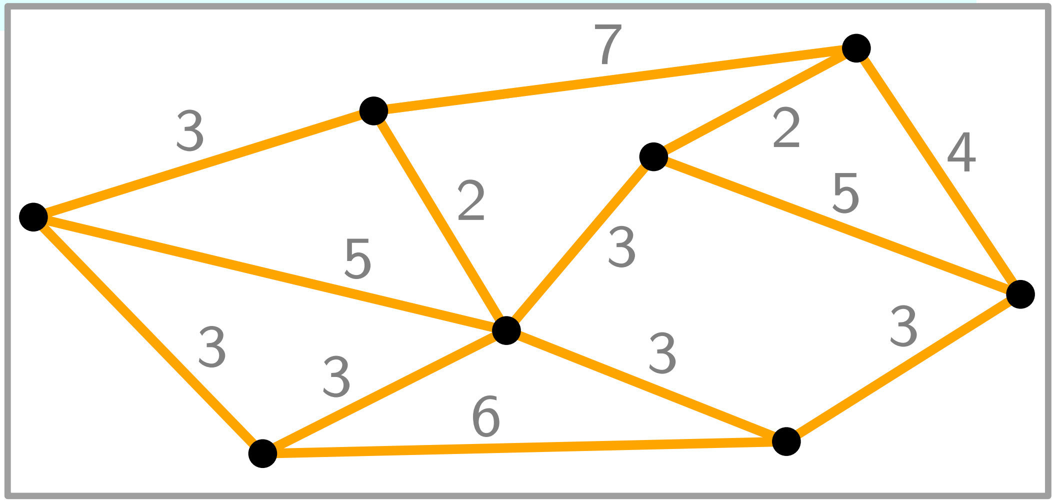
$\Rightarrow O((E + V) \log V)$ [Heap/RS-Baum]

$\Rightarrow O(E + V \log V)$ [Fibonacci-Heap]



Der Algorithmus von Kruskal (1956)

Kruskal(WeightedGraph $G = (V, E; w)$)

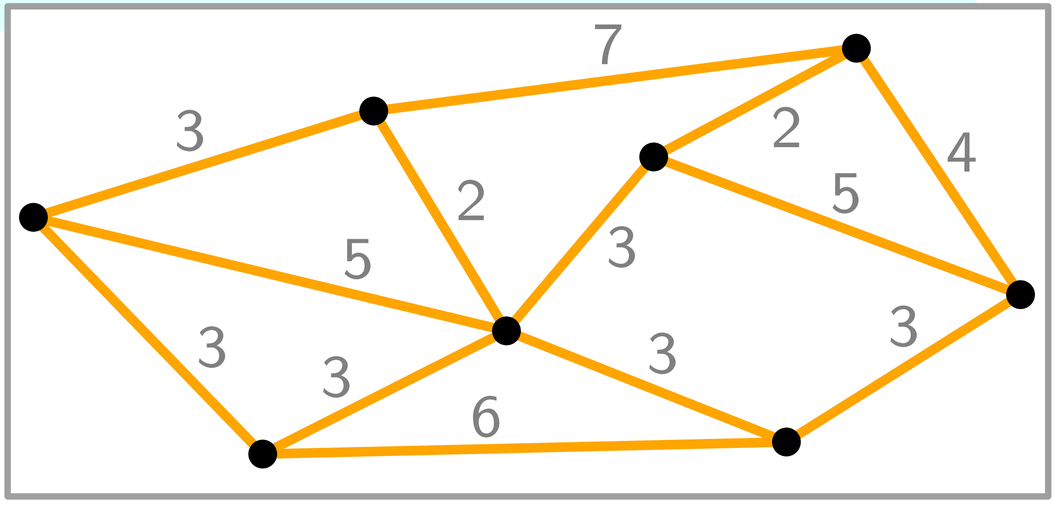


Der Algorithmus von Kruskal (1956)

Kruskal(WeightedGraph $G = (V, E; w)$)

$E' = \emptyset$

Sortiere E nicht-absteigend nach Gewicht w



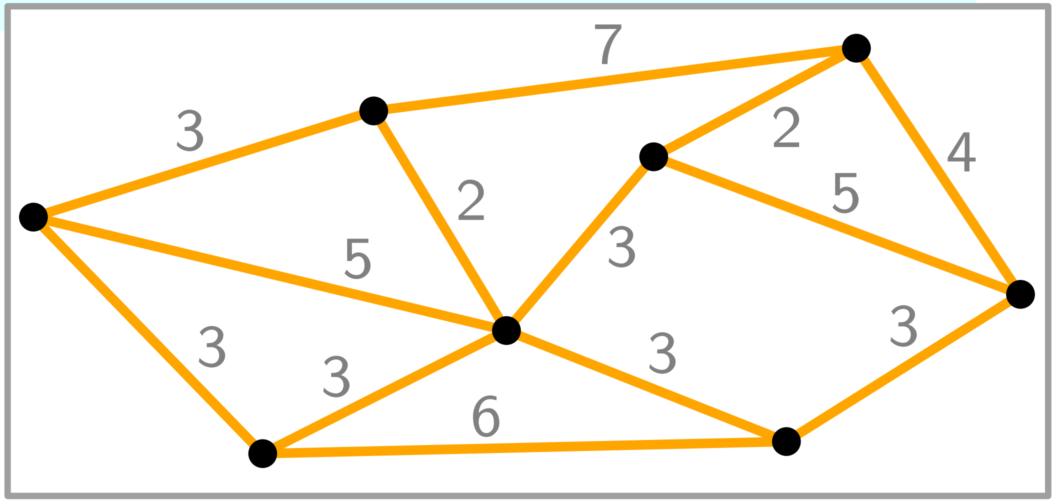
Der Algorithmus von Kruskal (1956)

Kruskal(WeightedGraph $G = (V, E; w)$)

$E' = \emptyset$

Sortiere E nicht-absteigend nach Gewicht w

foreach $uv \in E$ **do**



Der Algorithmus von Kruskal (1956)

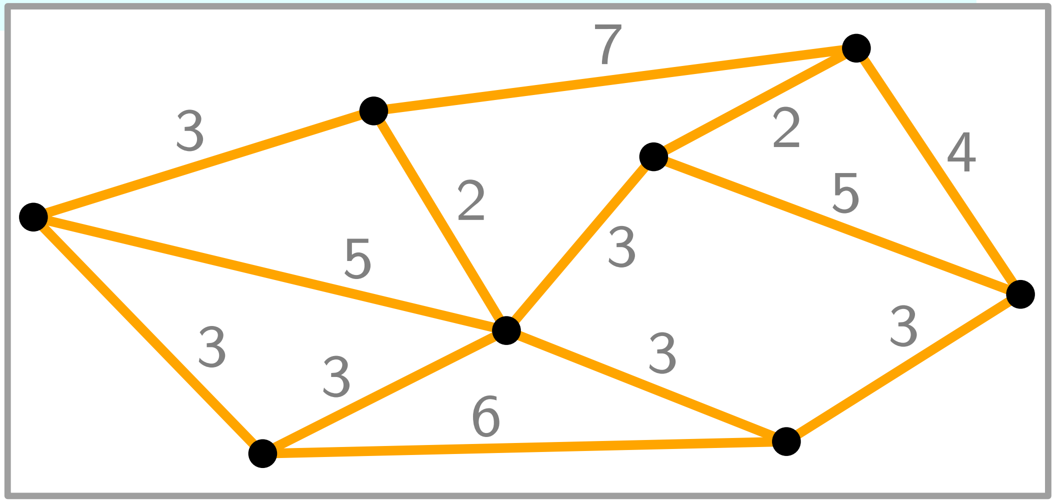
Kruskal(WeightedGraph $G = (V, E; w)$)

$E' = \emptyset$

Sortiere E nicht-absteigend nach Gewicht w

foreach $uv \in E$ **do**

if $E' \cup \{uv\}$ enthält keinen Kreis **then**



Der Algorithmus von Kruskal (1956)

Kruskal(WeightedGraph $G = (V, E; w)$)

$E' = \emptyset$

Sortiere E nicht-absteigend nach Gewicht w

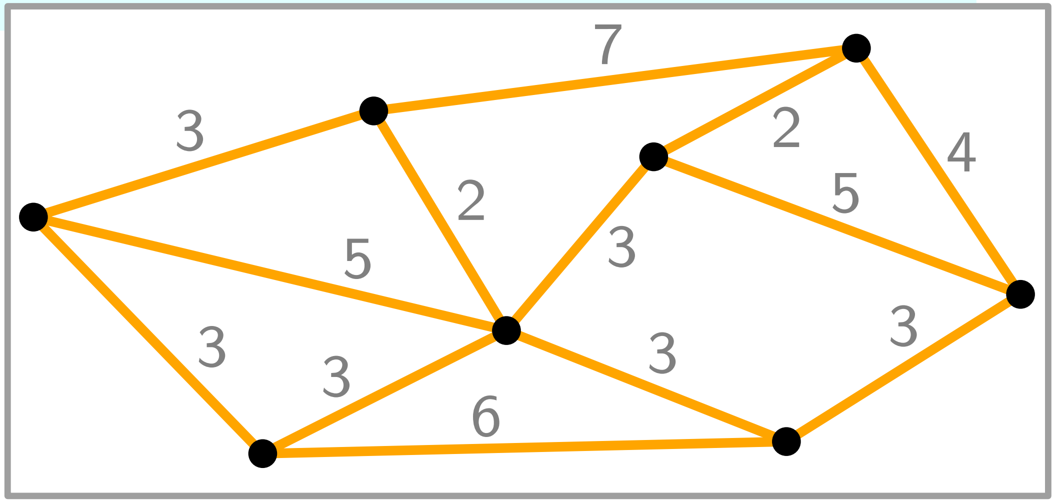
foreach $uv \in E$ **do**

  if $E' \cup \{uv\}$ enthält keinen Kreis **then**

    Färbe uv blau

    $E' = E' \cup \{uv\}$

Blaue Regel



Der Algorithmus von Kruskal (1956)

Kruskal(WeightedGraph $G = (V, E; w)$)

$E' = \emptyset$

Sortiere E nicht-absteigend nach Gewicht w

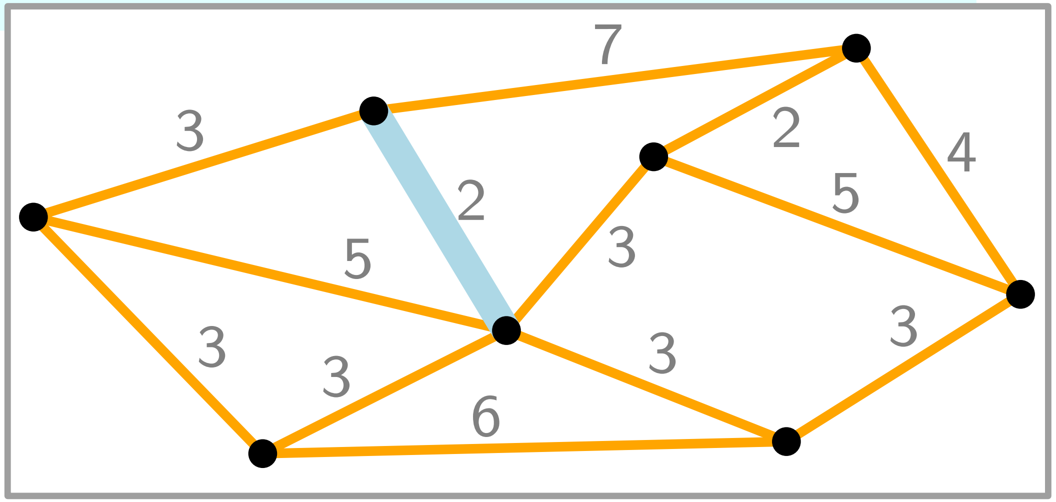
foreach $uv \in E$ **do**

  if $E' \cup \{uv\}$ enthält keinen Kreis **then**

    Färbe uv blau

    $E' = E' \cup \{uv\}$

Blaue Regel



Der Algorithmus von Kruskal (1956)

Kruskal(WeightedGraph $G = (V, E; w)$)

$E' = \emptyset$

Sortiere E nicht-absteigend nach Gewicht w

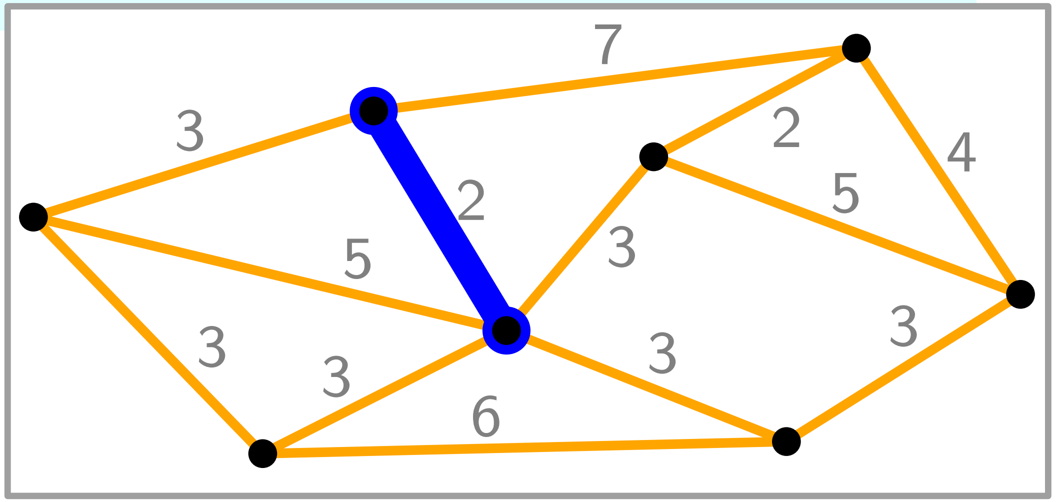
foreach $uv \in E$ **do**

if $E' \cup \{uv\}$ enthält keinen Kreis **then**

Färbe uv blau

$E' = E' \cup \{uv\}$

Blaue Regel



Der Algorithmus von Kruskal (1956)

Kruskal(WeightedGraph $G = (V, E; w)$)

$E' = \emptyset$

Sortiere E nicht-absteigend nach Gewicht w

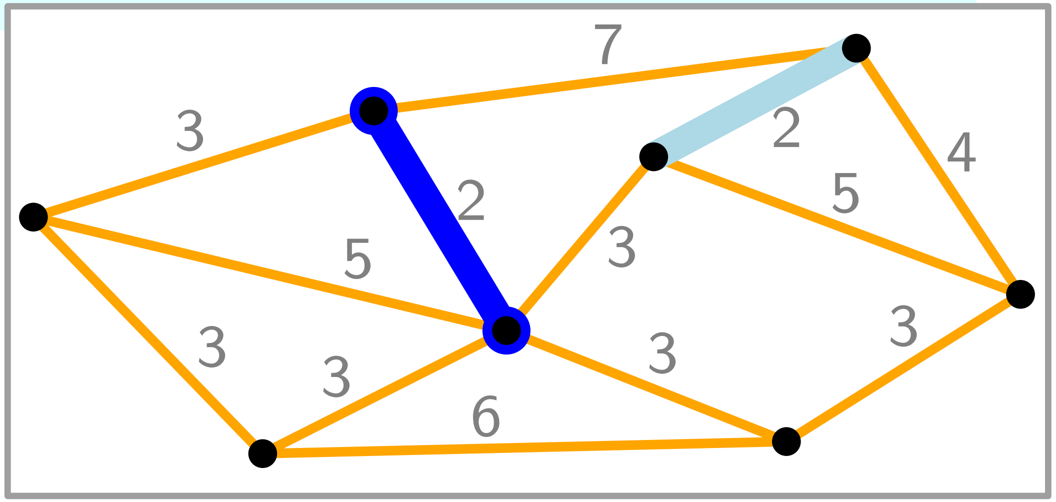
foreach $uv \in E$ **do**

  if $E' \cup \{uv\}$ enthält keinen Kreis **then**

    Färbe uv blau

    $E' = E' \cup \{uv\}$

Blaue Regel



Der Algorithmus von Kruskal (1956)

Kruskal(WeightedGraph $G = (V, E; w)$)

$E' = \emptyset$

Sortiere E nicht-absteigend nach Gewicht w

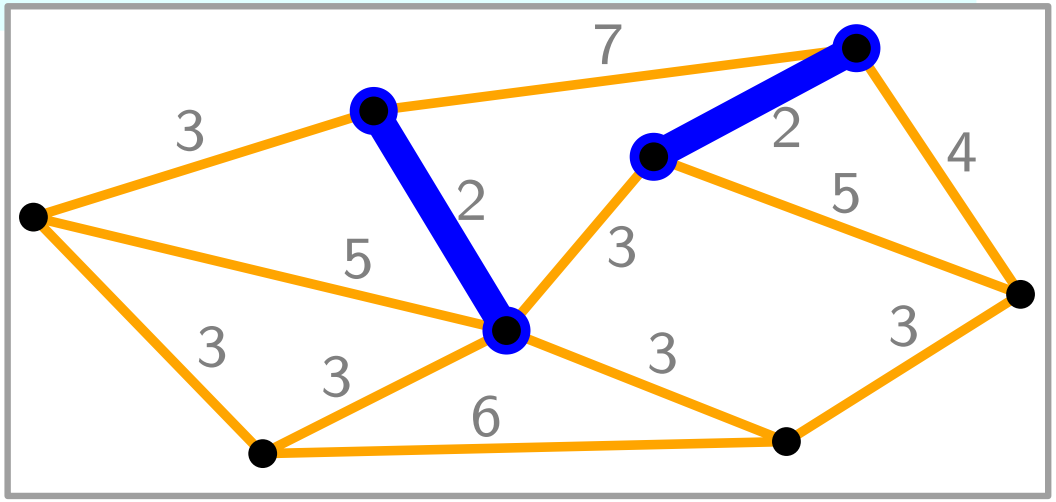
foreach $uv \in E$ **do**

  if $E' \cup \{uv\}$ enthält keinen Kreis **then**

    Färbe uv blau

    $E' = E' \cup \{uv\}$

Blaue Regel



Der Algorithmus von Kruskal (1956)

Kruskal(WeightedGraph $G = (V, E; w)$)

$E' = \emptyset$

Sortiere E nicht-absteigend nach Gewicht w

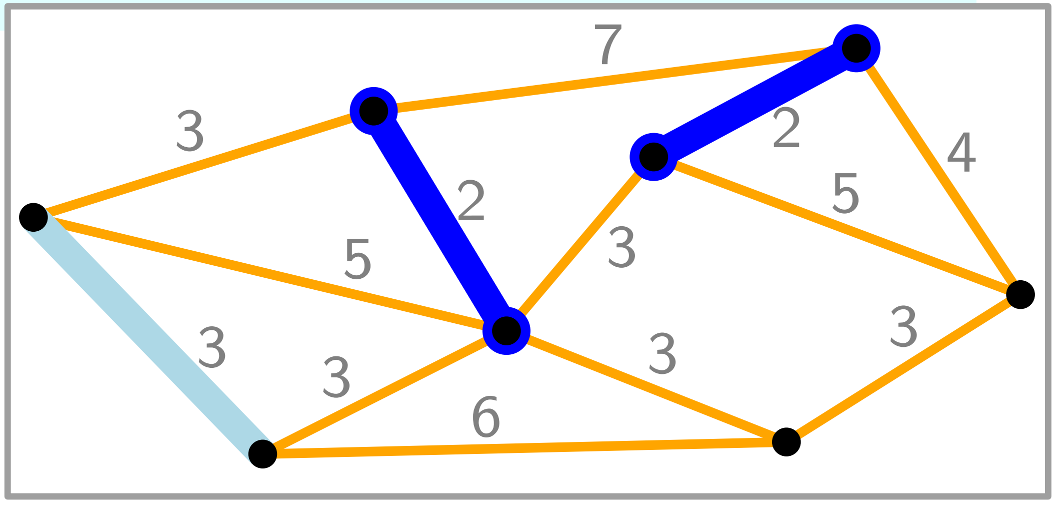
foreach $uv \in E$ **do**

  if $E' \cup \{uv\}$ enthält keinen Kreis **then**

    Färbe uv blau

    $E' = E' \cup \{uv\}$

Blaue Regel



Der Algorithmus von Kruskal (1956)

Kruskal(WeightedGraph $G = (V, E; w)$)

$E' = \emptyset$

Sortiere E nicht-absteigend nach Gewicht w

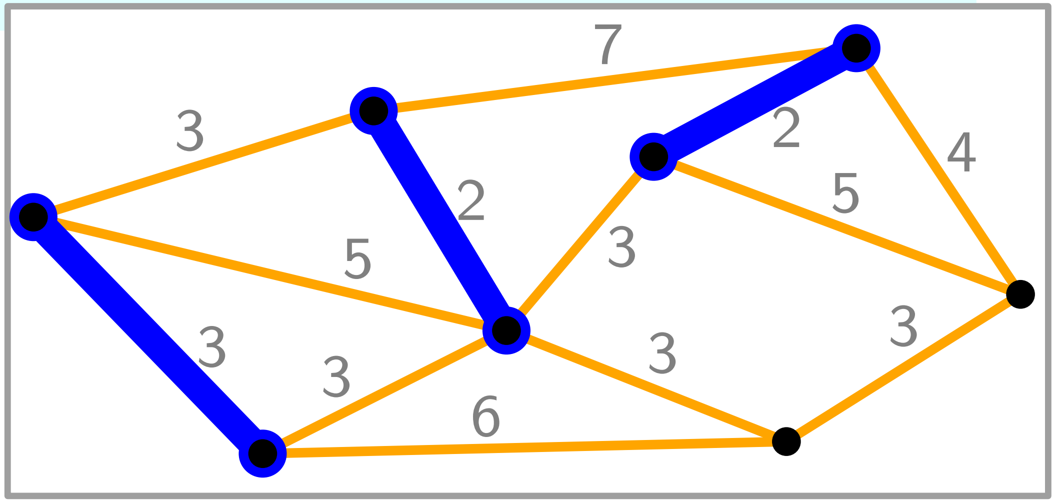
foreach $uv \in E$ **do**

  if $E' \cup \{uv\}$ enthält keinen Kreis **then**

    Färbe uv blau

    $E' = E' \cup \{uv\}$

Blaue Regel



Der Algorithmus von Kruskal (1956)

Kruskal(WeightedGraph $G = (V, E; w)$)

$E' = \emptyset$

Sortiere E nicht-absteigend nach Gewicht w

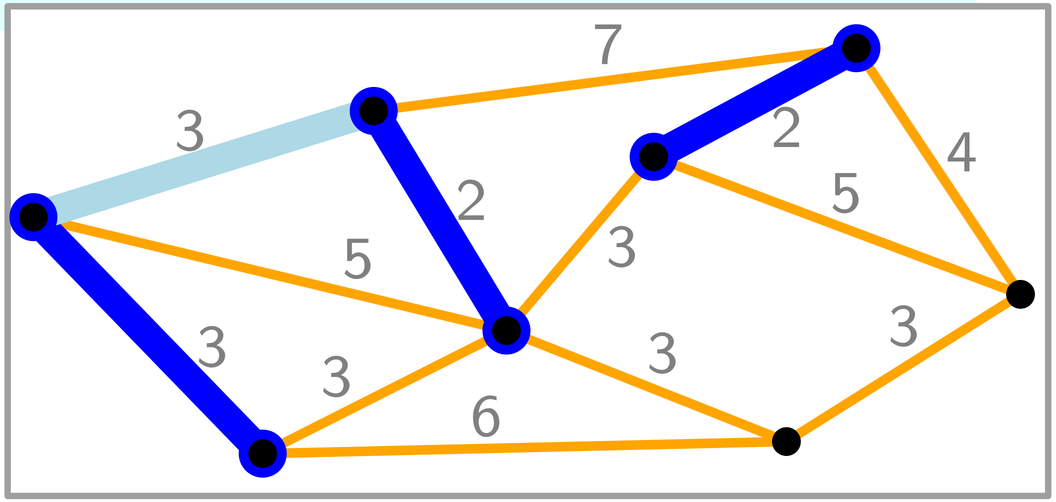
foreach $uv \in E$ **do**

  if $E' \cup \{uv\}$ enthält keinen Kreis **then**

    Färbe uv blau

    $E' = E' \cup \{uv\}$

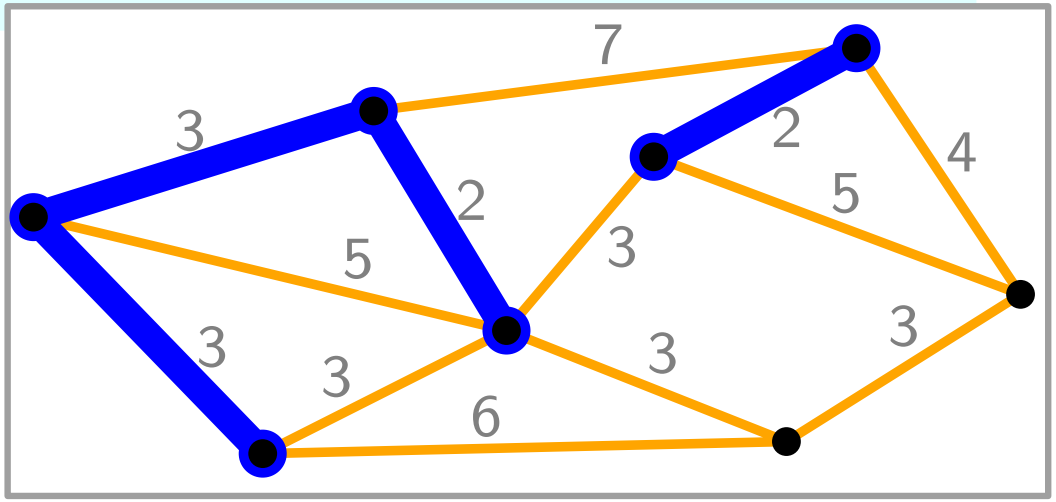
Blaue Regel



Der Algorithmus von Kruskal (1956)

Kruskal(WeightedGraph $G = (V, E; w)$)
  $E' = \emptyset$
  Sortiere E nicht-absteigend nach Gewicht w
  foreach $uv \in E$ **do**
    if $E' \cup \{uv\}$ enthält keinen Kreis **then**
      Färbe uv blau
      $E' = E' \cup \{uv\}$

Blaue Regel



Der Algorithmus von Kruskal (1956)

Kruskal(WeightedGraph $G = (V, E; w)$)

$E' = \emptyset$

Sortiere E nicht-absteigend nach Gewicht w

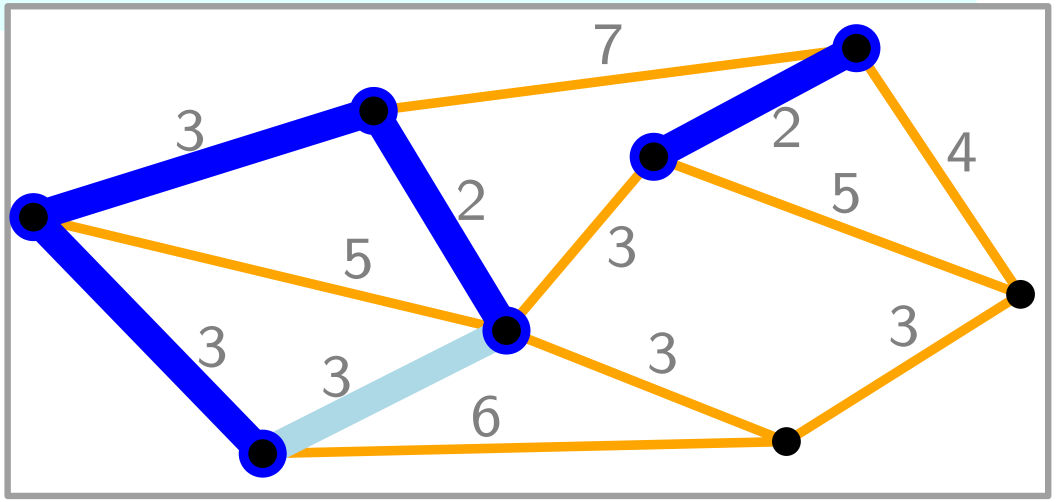
foreach $uv \in E$ **do**

  if $E' \cup \{uv\}$ enthält keinen Kreis **then**

    Färbe uv blau

    $E' = E' \cup \{uv\}$

Blaue Regel



Der Algorithmus von Kruskal (1956)

Kruskal(WeightedGraph $G = (V, E; w)$)

$E' = \emptyset$

Sortiere E nicht-absteigend nach Gewicht w

foreach $uv \in E$ **do**

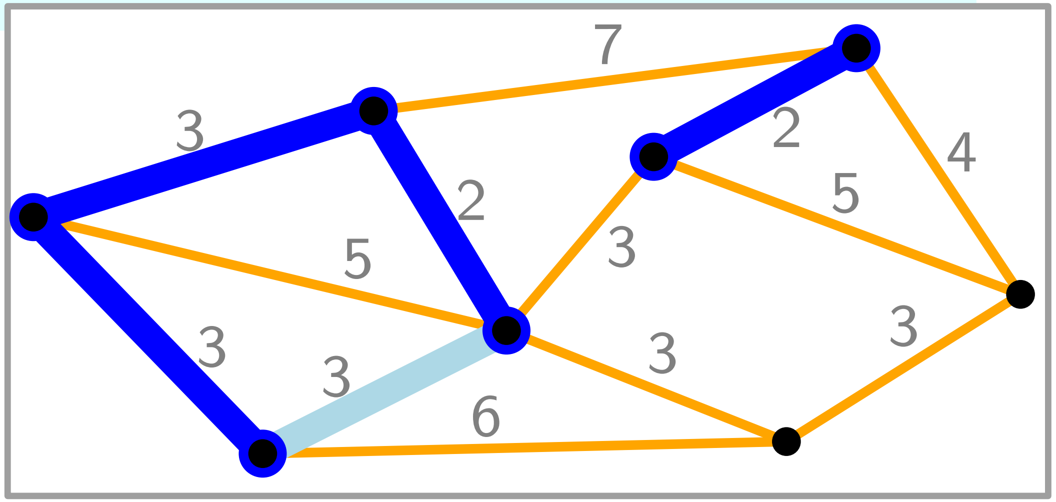
  if $E' \cup \{uv\}$ enthält keinen Kreis **then**

    Färbe uv blau Blaue Regel

    $E' = E' \cup \{uv\}$

  else

    Färbe uv rot Rote Regel



Der Algorithmus von Kruskal (1956)

Kruskal(WeightedGraph $G = (V, E; w)$)

$E' = \emptyset$

Sortiere E nicht-absteigend nach Gewicht w

foreach $uv \in E$ **do**

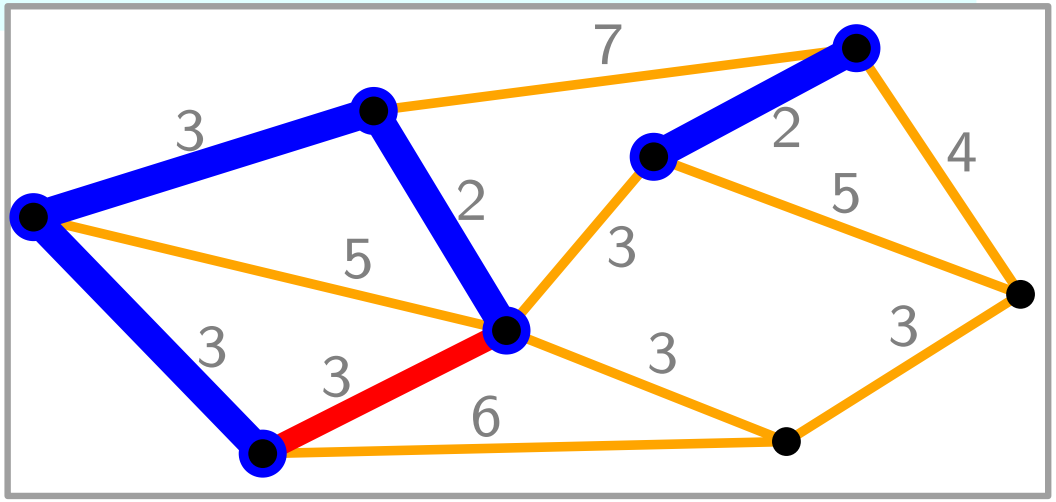
  if $E' \cup \{uv\}$ enthält keinen Kreis **then**

    Färbe uv blau — Blaue Regel

    $E' = E' \cup \{uv\}$

  else

    Färbe uv rot — Rote Regel



Der Algorithmus von Kruskal (1956)

Kruskal(WeightedGraph $G = (V, E; w)$)

$E' = \emptyset$

Sortiere E nicht-absteigend nach Gewicht w

foreach $uv \in E$ **do**

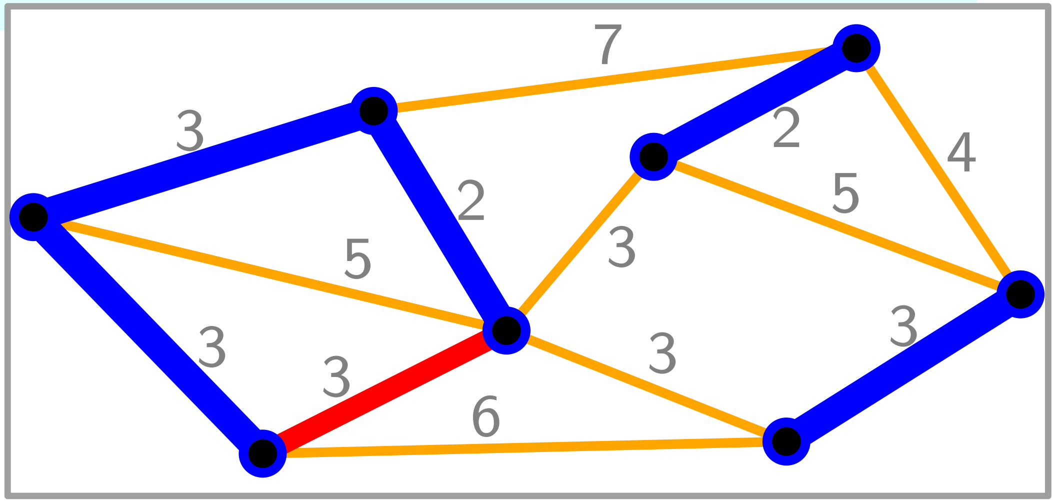
if $E' \cup \{uv\}$ enthält keinen Kreis **then**

Färbe uv blau — Blaue Regel

$E' = E' \cup \{uv\}$

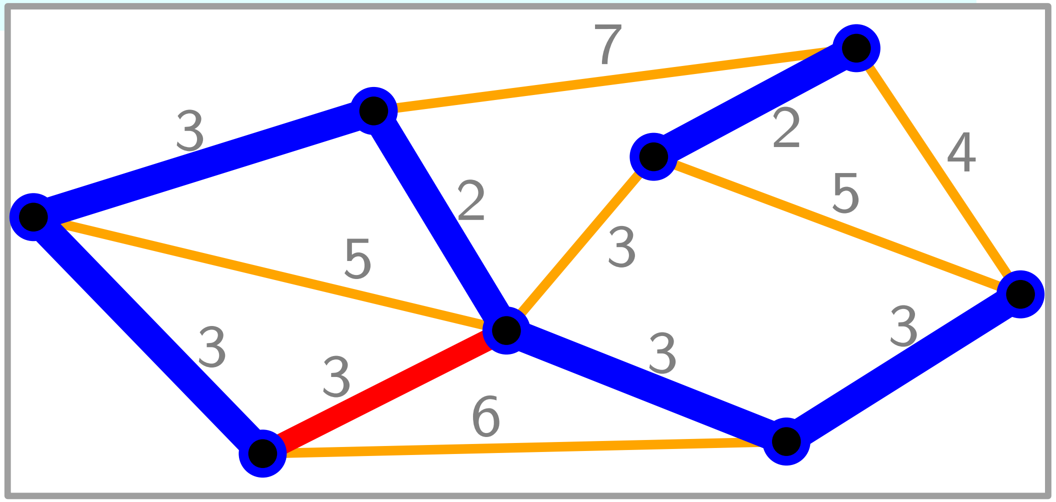
else

Färbe uv rot — Rote Regel



Der Algorithmus von Kruskal (1956)

Kruskal(WeightedGraph $G = (V, E; w)$)
  $E' = \emptyset$
  Sortiere E nicht-absteigend nach Gewicht w
  foreach $uv \in E$ **do**
    if $E' \cup \{uv\}$ enthält keinen Kreis **then**
      Färbe uv blau — Blaue Regel
      $E' = E' \cup \{uv\}$
    else
      Färbe uv rot — Rote Regel



Der Algorithmus von Kruskal (1956)

Kruskal(WeightedGraph $G = (V, E; w)$)

$E' = \emptyset$

Sortiere E nicht-absteigend nach Gewicht w

foreach $uv \in E$ **do**

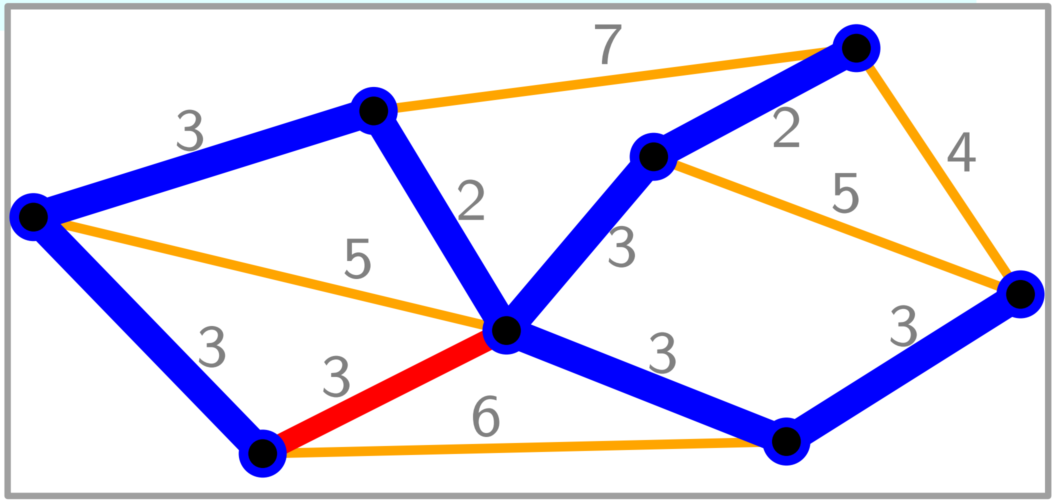
if $E' \cup \{uv\}$ enthält keinen Kreis **then**

Färbe uv blau — Blaue Regel

$E' = E' \cup \{uv\}$

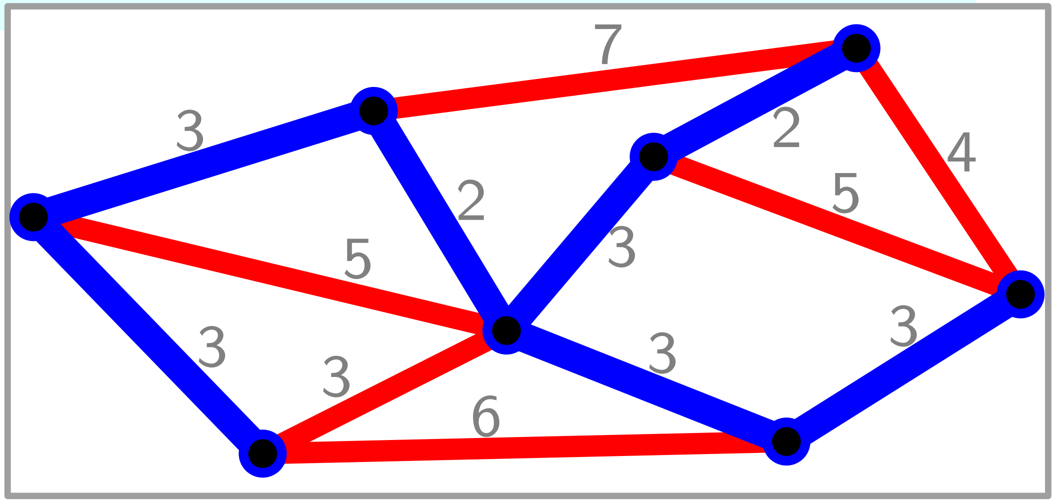
else

Färbe uv rot — Rote Regel



Der Algorithmus von Kruskal (1956)

```
Kruskal(WeightedGraph G = (V, E; w))
  E' = ∅
  Sortiere E nicht-absteigend nach Gewicht w
  foreach uv ∈ E do
      if E' ∪ {uv} enthält keinen Kreis then
          Färbe uv blau                          Blaue Regel
          E' = E' ∪ {uv}
      else
          Färbe uv rot                           Rote Regel
```



Der Algorithmus von Kruskal (1956)

Kruskal(WeightedGraph $G = (V, E; w)$)

$E' = \emptyset$

Sortiere E nicht-absteigend nach Gewicht w

foreach $uv \in E$ **do**

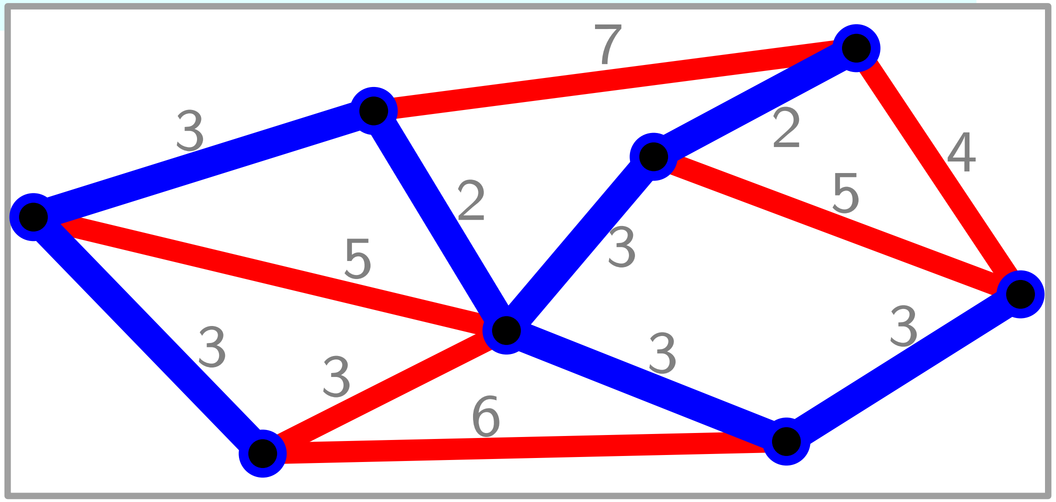
  if $E' \cup \{uv\}$ enthält keinen Kreis **then**

    Färbe uv blau — Blaue Regel

    $E' = E' \cup \{uv\}$

  else

    Färbe uv rot — Rote Regel



Laufzeit?

Der Algorithmus von Kruskal (1956)

Kruskal(WeightedGraph $G = (V, E; w)$)

$E' = \emptyset$

Sortiere E nicht-absteigend nach Gewicht w

foreach $uv \in E$ **do**

  if $E' \cup \{uv\}$ enthält keinen Kreis **then**

    Färbe uv blau — Blaue Regel

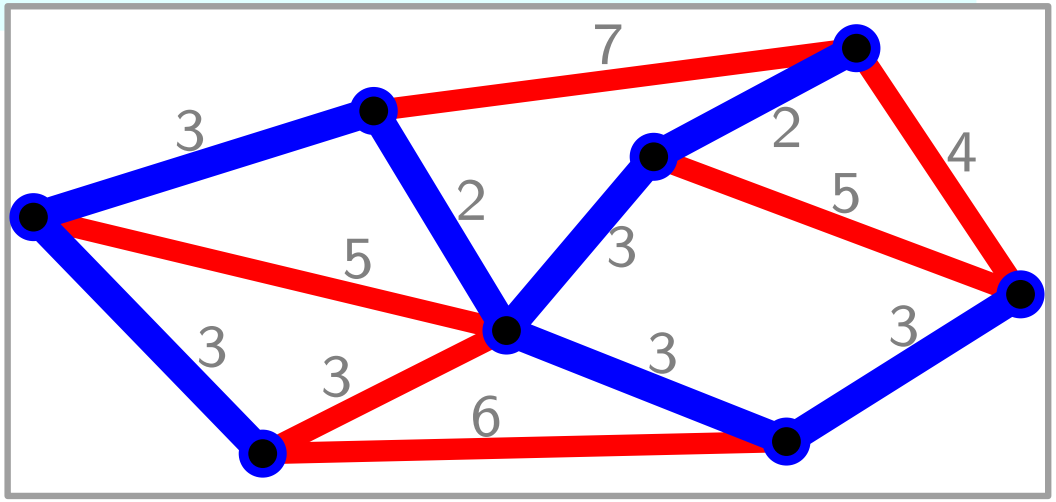
    $E' = E' \cup \{uv\}$

  else

    Färbe uv rot — Rote Regel

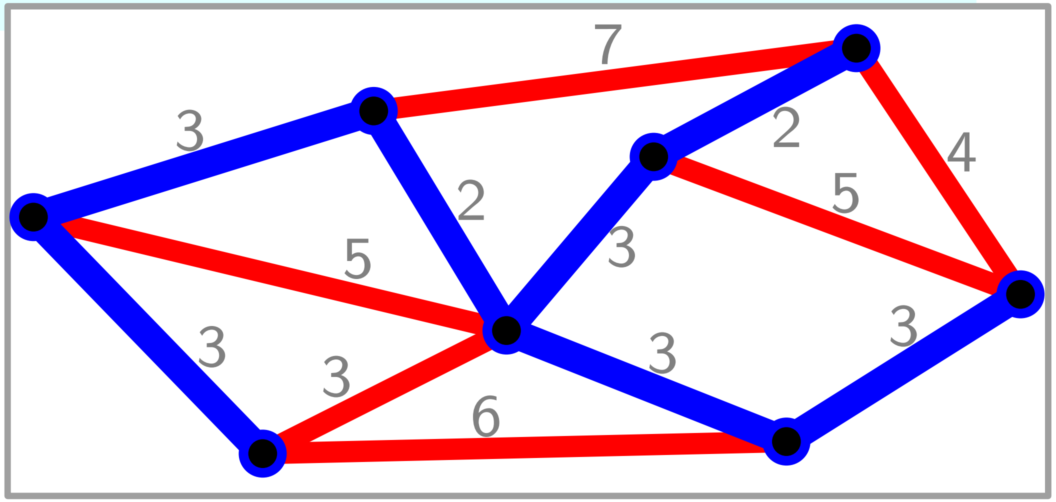
Laufzeit?

$O(E \log V)$



Der Algorithmus von Kruskal (1956)

Kruskal(WeightedGraph $G = (V, E; w)$)
  $E' = \emptyset$
  Sortiere E nicht-absteigend nach Gewicht w
  foreach $uv \in E$ **do**
    if $E' \cup \{uv\}$ enthält keinen Kreis **then**
      Färbe uv blau — Blaue Regel
      $E' = E' \cup \{uv\}$
    else
      Färbe uv rot — Rote Regel



Laufzeit?

$O(E \log V)$

$O(E \cdot \alpha(V))$ falls vorsortiert

Der Algorithmus von Kruskal (1956)

Kruskal(WeightedGraph $G = (V, E; w)$)

$E' = \emptyset$

Sortiere E nicht-absteigend nach Gewicht w

foreach $uv \in E$ **do**

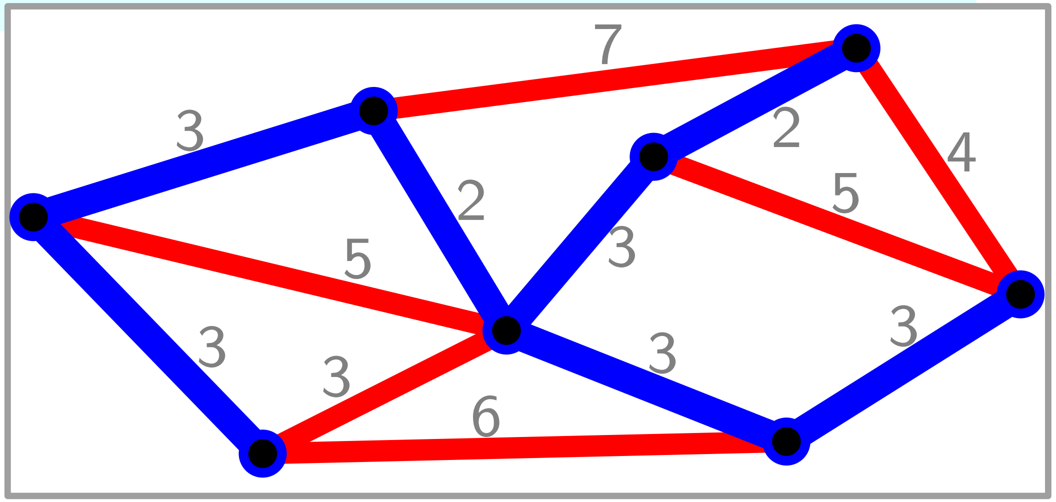
if $E' \cup \{uv\}$ enthält keinen Kreis **then**

Färbe uv blau — Blaue Regel

$E' = E' \cup \{uv\}$

else

Färbe uv rot — Rote Regel



Laufzeit?

$O(E \log V)$

$O(E \cdot \alpha(V))$ falls vorsortiert

≤ 4 für $|V| \leq 10^{80}$