

# Algorithmische Graphentheorie

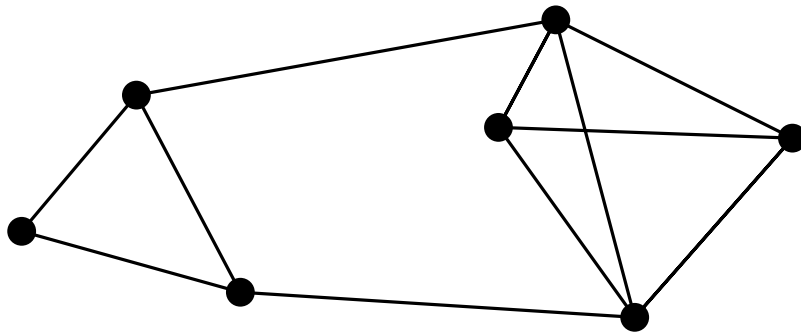
Sommersemester 2020

8. Vorlesung

Randomisierte Algorithmen  
für MINCUT

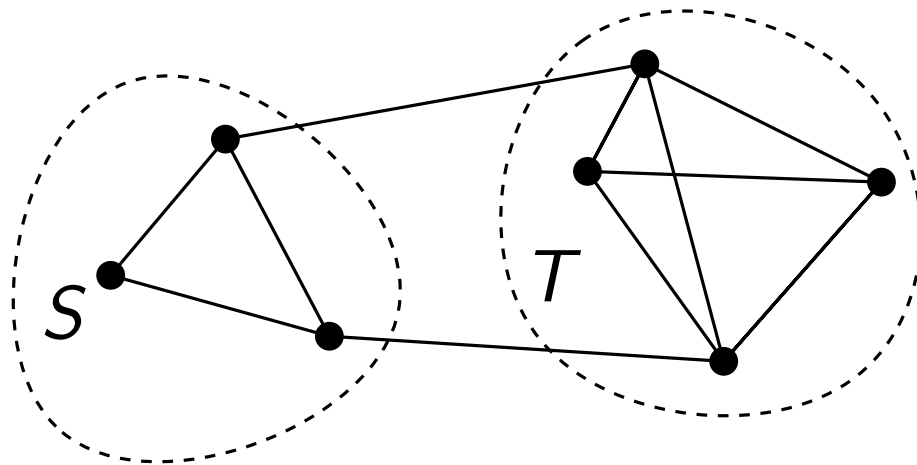
# MINCUT – kleinste Schnitte

**Def.** Gegeben sei ein ungerichteter Multigraph  $G = (V, E)$ . Gesucht ist eine Zerlegung  $(S, T)$  von  $V$  mit  $S, T \neq \emptyset$ , so dass die Anzahl der Kanten  $uv \in E$  mit  $u \in S$  und  $v \in T$  möglichst klein ist.



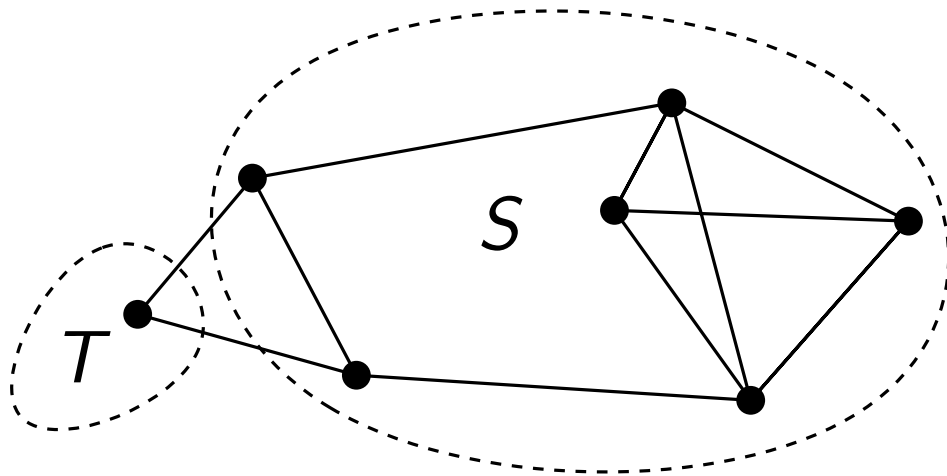
# MINCUT – kleinste Schnitte

**Def.** Gegeben sei ein ungerichteter Multigraph  $G = (V, E)$ . Gesucht ist eine Zerlegung  $(S, T)$  von  $V$  mit  $S, T \neq \emptyset$ , so dass die Anzahl der Kanten  $uv \in E$  mit  $u \in S$  und  $v \in T$  möglichst klein ist.



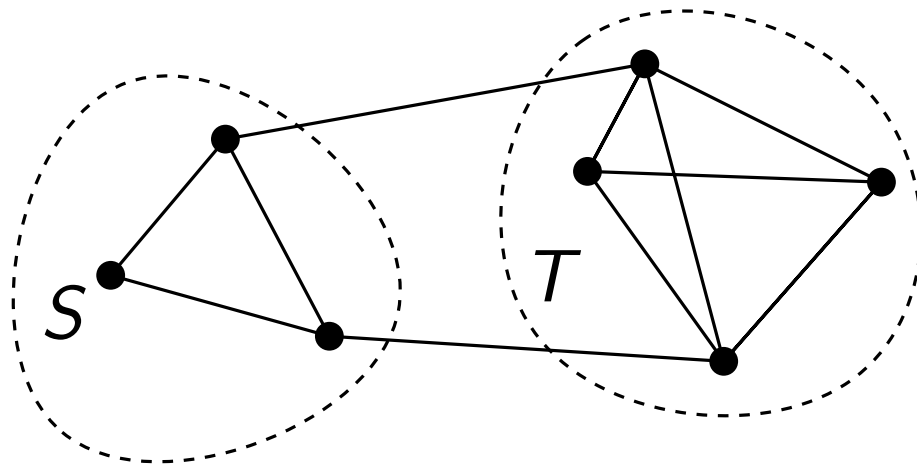
# MINCUT – kleinste Schnitte

**Def.** Gegeben sei ein ungerichteter Multigraph  $G = (V, E)$ . Gesucht ist eine Zerlegung  $(S, T)$  von  $V$  mit  $S, T \neq \emptyset$ , so dass die Anzahl der Kanten  $uv \in E$  mit  $u \in S$  und  $v \in T$  möglichst klein ist.



# MINCUT – kleinste Schnitte

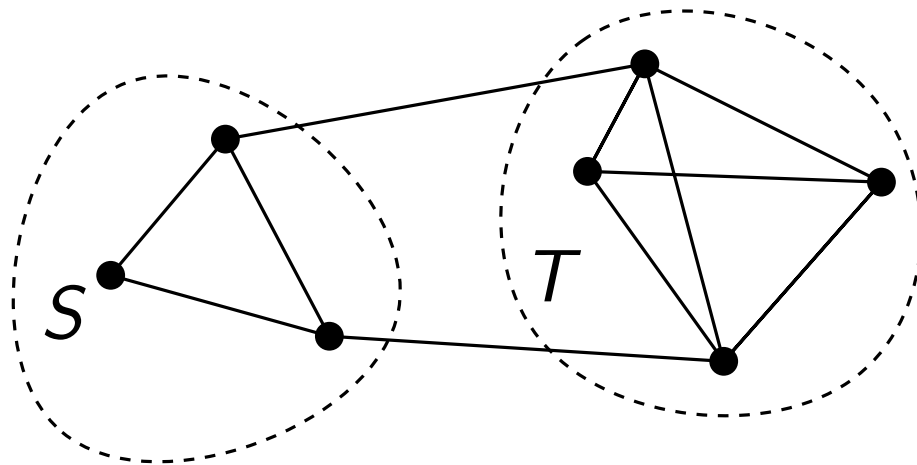
**Def.** Gegeben sei ein ungerichteter Multigraph  $G = (V, E)$ . Gesucht ist eine Zerlegung  $(S, T)$  von  $V$  mit  $S, T \neq \emptyset$ , so dass die Anzahl der Kanten  $uv \in E$  mit  $u \in S$  und  $v \in T$  möglichst klein ist.



**Motivation: ?**

# MINCUT – kleinste Schnitte

**Def.** Gegeben sei ein ungerichteter Multigraph  $G = (V, E)$ . Gesucht ist eine Zerlegung  $(S, T)$  von  $V$  mit  $S, T \neq \emptyset$ , so dass die Anzahl der Kanten  $uv \in E$  mit  $u \in S$  und  $v \in T$  möglichst klein ist.

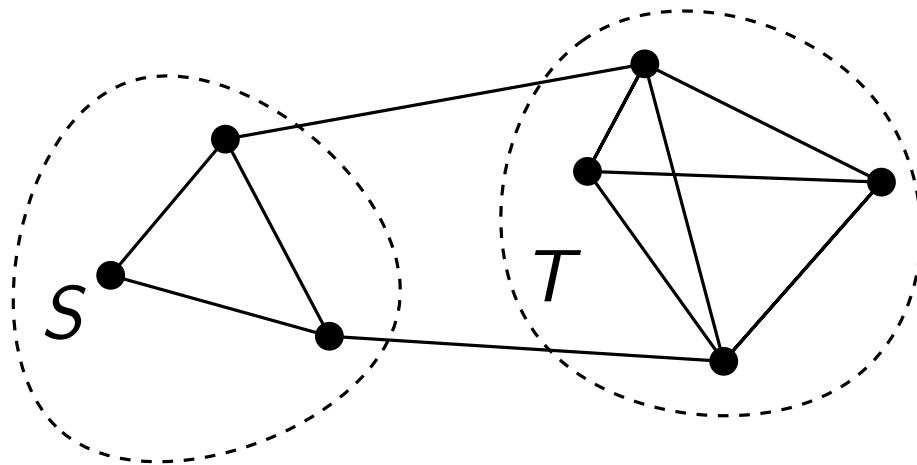


## Motivation:

Z.B. Finden von Schwachstellen in Kommunikationsnetzwerken.

# MINCUT – kleinste Schnitte

**Def.** Gegeben sei ein ungerichteter Multigraph  $G = (V, E)$ . Gesucht ist eine Zerlegung  $(S, T)$  von  $V$  mit  $S, T \neq \emptyset$ , so dass die Anzahl der Kanten  $uv \in E$  mit  $u \in S$  und  $v \in T$  möglichst klein ist.



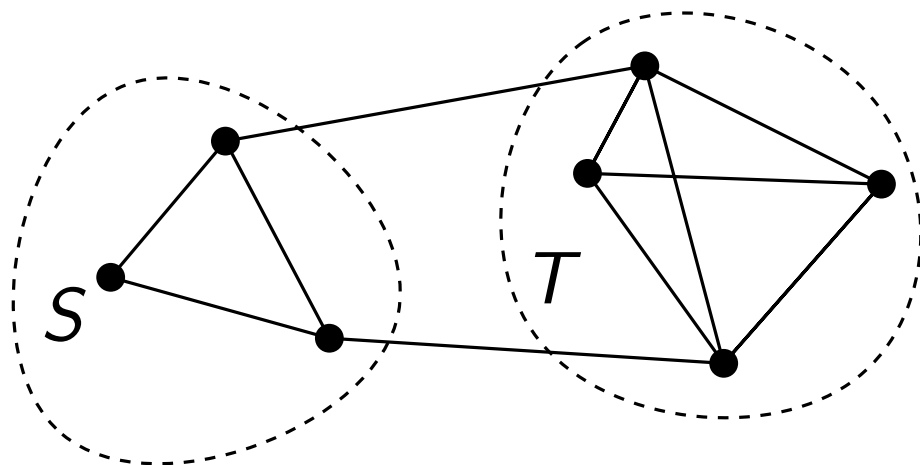
## Motivation:

Z.B. Finden von Schwachstellen in Kommunikationsnetzwerken.

**Beachte:** Im Gegensatz zu  $s$ - $t$ -Schnitten ist hier *kein* zu trennendes Knotenpaar  $(s, t)$  vorgegeben.

# MINCUT – kleinste Schnitte

**Def.** Gegeben sei ein ungerichteter Multigraph  $G = (V, E)$ . Gesucht ist eine Zerlegung  $(S, T)$  von  $V$  mit  $S, T \neq \emptyset$ , so dass die Anzahl der Kanten  $uv \in E$  mit  $u \in S$  und  $v \in T$  möglichst klein ist.



## Motivation:

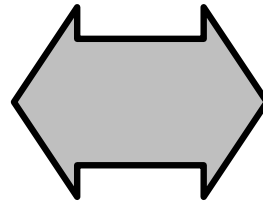
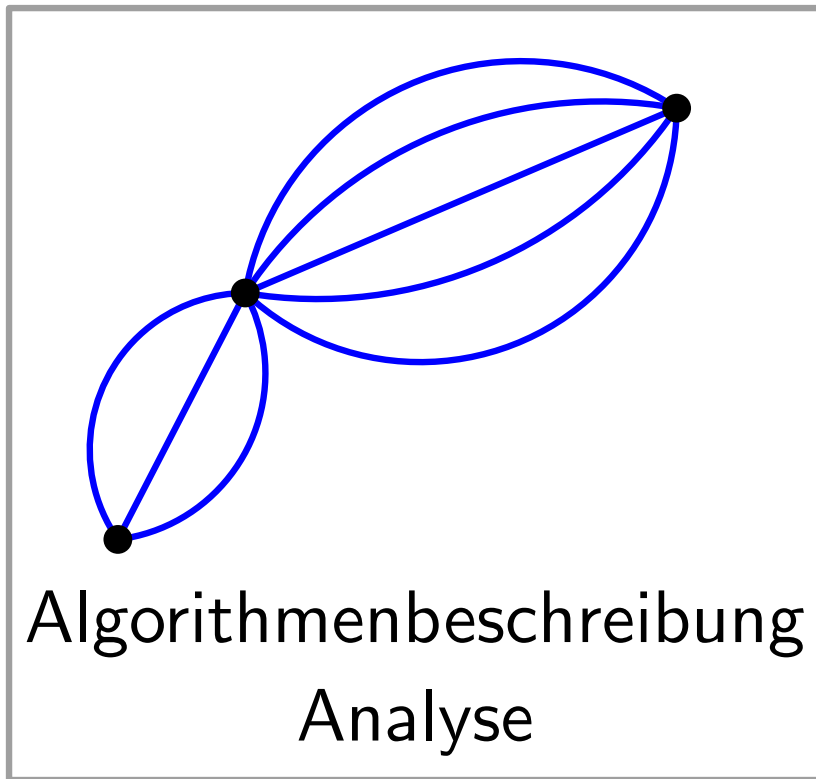
Z.B. Finden von Schwachstellen in Kommunikationsnetzwerken.

**Beachte:** Im Gegensatz zu  $s$ - $t$ -Schnitten ist hier *kein* zu trennendes Knotenpaar  $(s, t)$  vorgegeben.

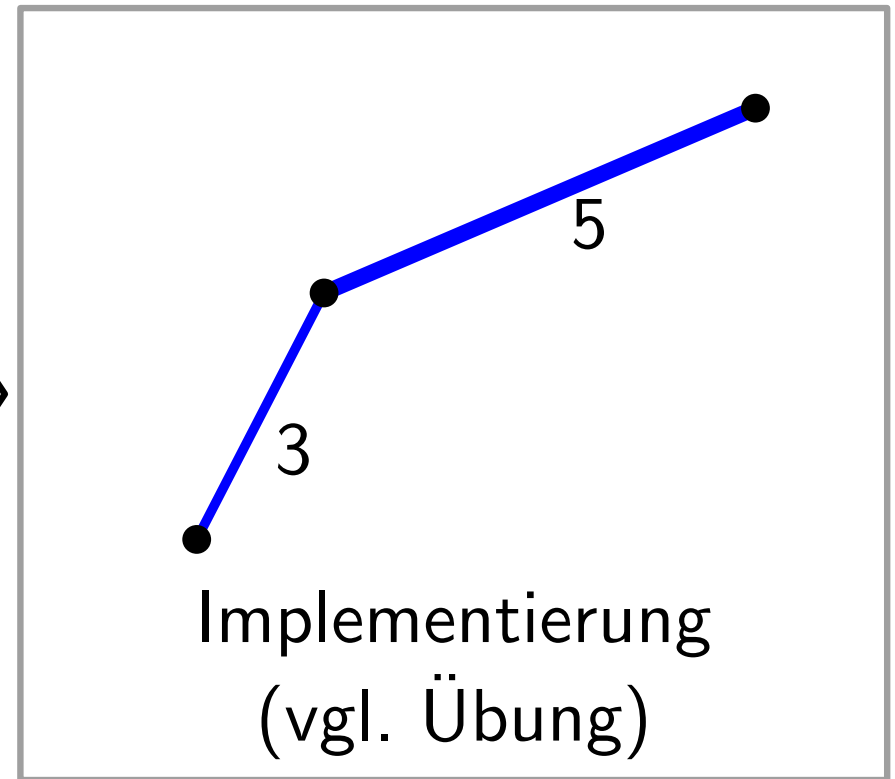
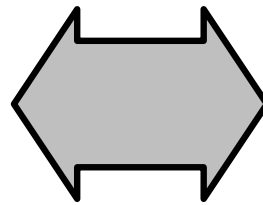
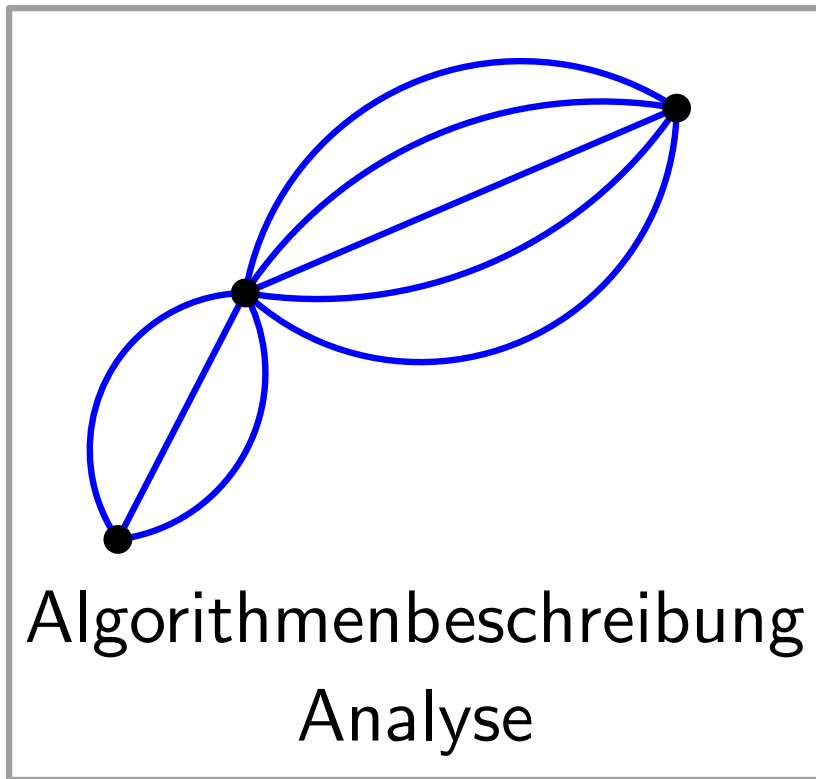
O.B.d.A. Multigraph  $G$  zusammenhängend, und  $n := |V| \geq 2$ .



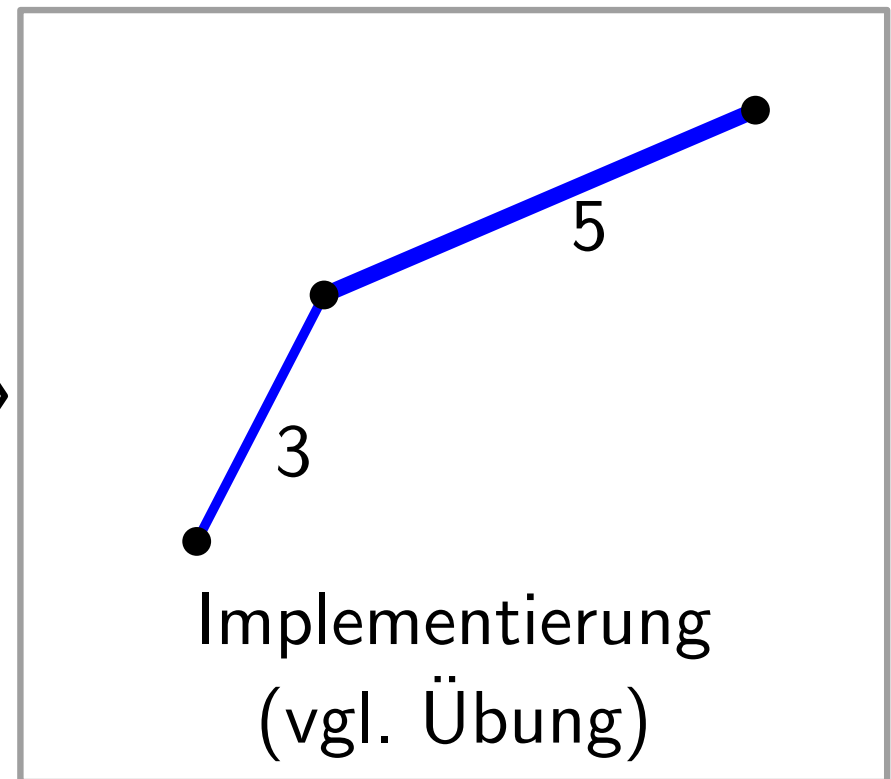
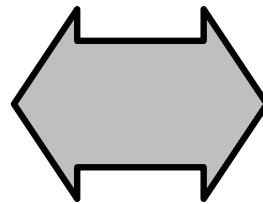
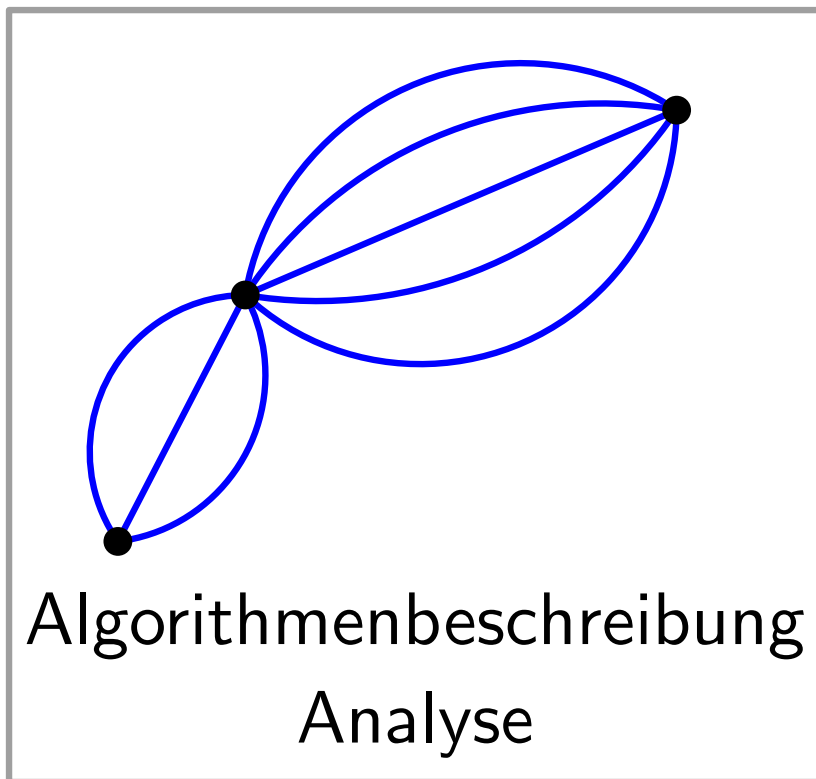
# Multigraphen vs. Kantengewichte



# Multigraphen vs. Kantengewichte



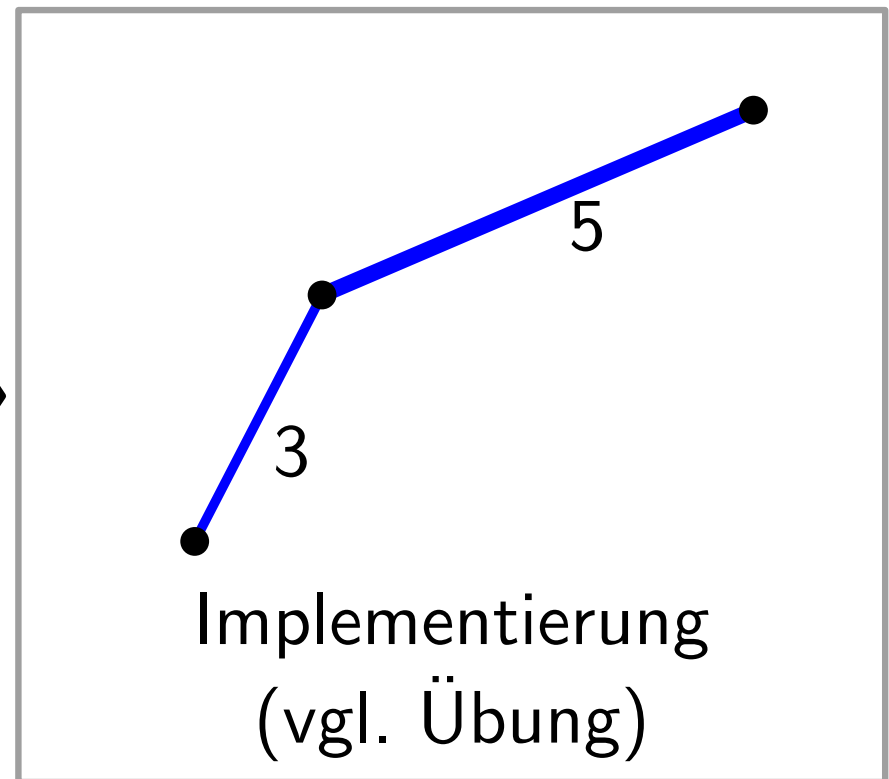
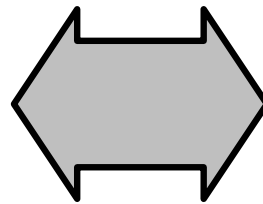
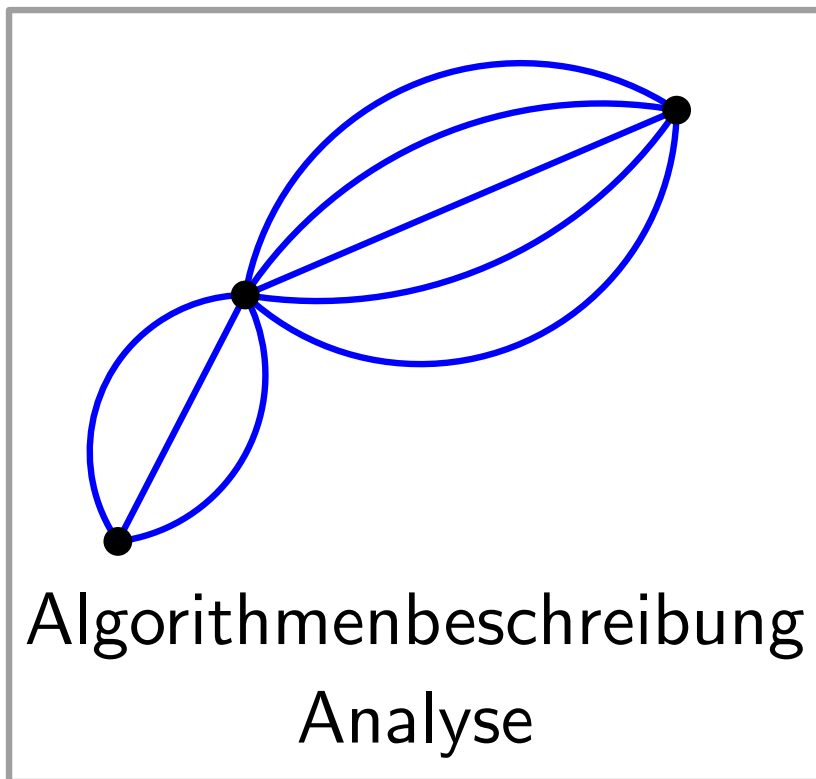
# Multigraphen vs. Kantengewichte



Gewichtetes MINCUT:

... minimiere Kosten  $\sum_{\substack{u \in S, v \in T \\ uv \in E}} c(uv)$ , wobei  $c: E \rightarrow \mathbb{N}$ .

# Multigraphen vs. Kantengewichte

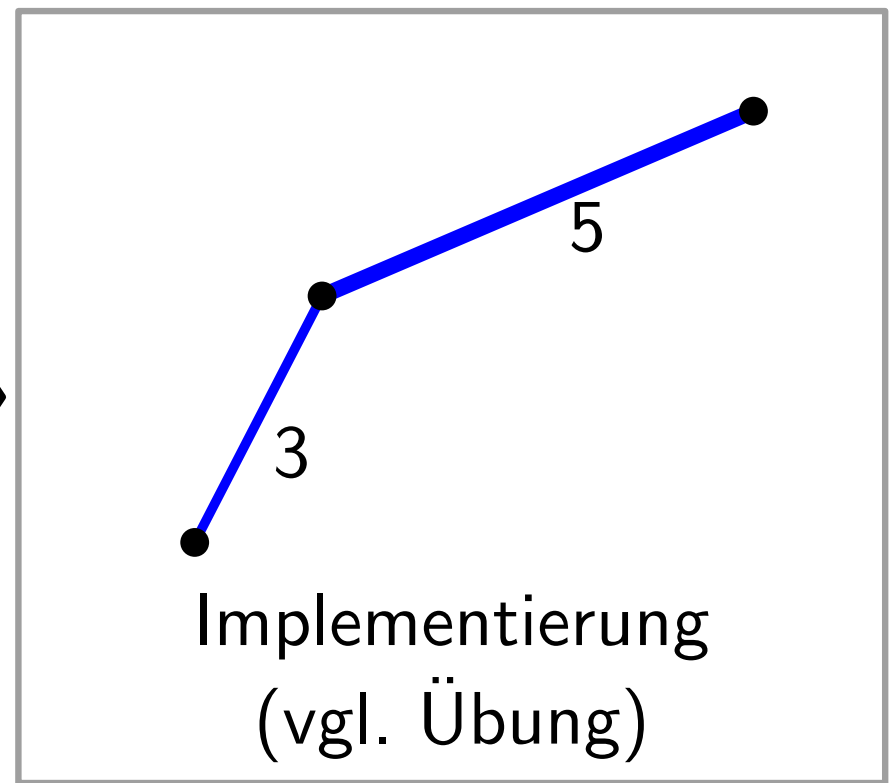
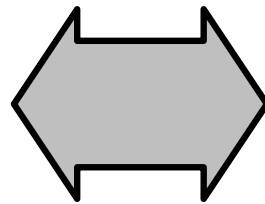
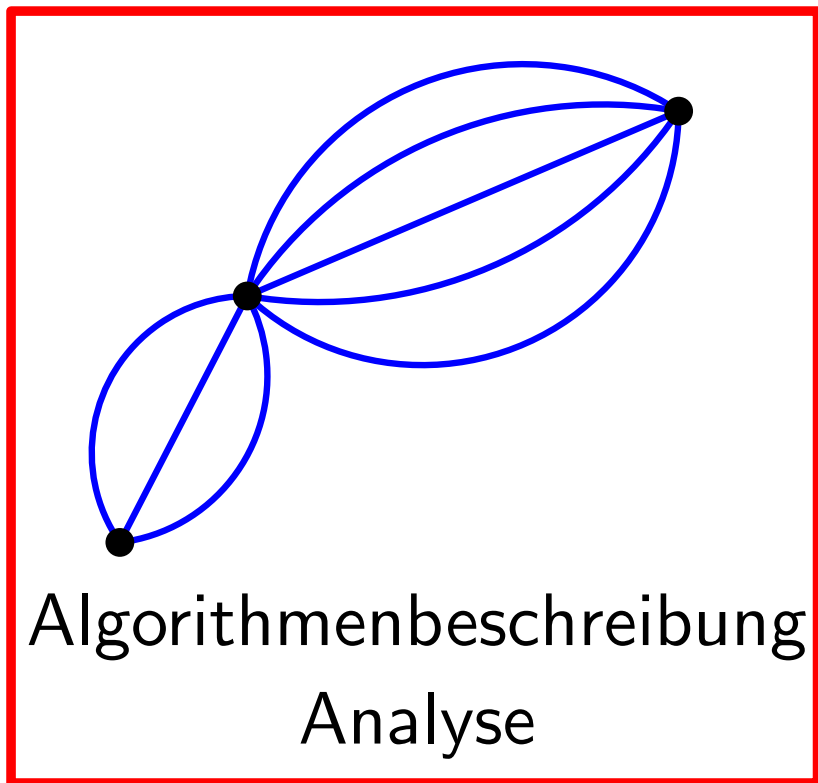


Gewichtetes MINCUT:

... minimiere Kosten  $\sum_{\substack{u \in S, v \in T \\ uv \in E}} c(uv)$ , wobei  $c: E \rightarrow \mathbb{N}$ .

... Algorithmen funktionieren auch für gewichtetes MINCUT

# Multigraphen vs. Kantengewichte



Gewichtetes MINCUT:

... minimiere Kosten  $\sum_{\substack{u \in S, v \in T \\ uv \in E}} c(uv)$ , wobei  $c: E \rightarrow \mathbb{N}$ .

... Algorithmen funktionieren auch für gewichtetes MINCUT

# Naiver (deterministischer) Ansatz

Übung:

Geben Sie einen Polynomialzeit-Algorithmus für MINCUT an!

# Naiver (deterministischer) Ansatz

Übung:

Geben Sie einen Polynomialzeit-Algorithmus für MINCUT an!

- Für jedes Paar  $\{s, t\} \in \binom{V}{2}$ ,  
berechne

# Naiver (deterministischer) Ansatz

Übung:

Geben Sie einen Polynomialzeit-Algorithmus für MINCUT an!

- Für jedes Paar  $\{s, t\} \in \binom{V}{2}$ ,  
berechne **kleinsten  $s$ - $t$ -Schnitt** mittels Edmonds-Karp.



# Naiver (deterministischer) Ansatz

Übung:

Geben Sie einen Polynomialzeit-Algorithmus für MINCUT an!

- Für jedes Paar  $\{s, t\} \in \binom{V}{2}$ ,  
berechne **kleinsten  $s$ - $t$ -Schnitt** mittels Edmonds-Karp.
- Gib den kleinsten dieser  $s$ - $t$ -Schnitte zurück.

# Naiver (deterministischer) Ansatz

Übung:

Geben Sie einen Polynomialzeit-Algorithmus für MINCUT an!

- Für jedes Paar  $\{s, t\} \in \binom{V}{2}$ ,  
berechne **kleinsten  $s$ - $t$ -Schnitt** mittels Edmonds-Karp.
- Gib den kleinsten dieser  $s$ - $t$ -Schnitte zurück.

Gesamtlaufzeit:  $O(V^2 \cdot VE^2) \subseteq O(V^7)$ .

# Naiver (deterministischer) Ansatz

Übung:

Geben Sie einen Polynomialzeit-Algorithmus für MINCUT an!

- Für jedes Paar  $\{s, t\} \in \binom{V}{2}$ ,  
berechne **kleinsten  $s$ - $t$ -Schnitt** mittels Edmonds-Karp.
- Gib den kleinsten dieser  $s$ - $t$ -Schnitte zurück.

Gesamtlaufzeit:  $O(V^2 \cdot VE^2) \subseteq O(V^7)$ .

Es geht auch in  $O(V^6)$  Zeit...

# Naiver (deterministischer) Ansatz

Übung:

Geben Sie einen Polynomialzeit-Algorithmus für MINCUT an!

- Für jedes Paar  $\{s, t\} \in \binom{V}{2}$ ,  
berechne **kleinsten  $s$ - $t$ -Schnitt** mittels Edmonds-Karp.
- Gib den kleinsten dieser  $s$ - $t$ -Schnitte zurück.

Gesamtlaufzeit:  $O(V^2 \cdot VE^2) \subseteq O(V^7)$ .

Es geht auch in  $O(V^6)$  Zeit...

– Wähle  $s$  fest.

# Naiver (deterministischer) Ansatz

Übung:

Geben Sie einen Polynomialzeit-Algorithmus für MINCUT an!

- Für jedes Paar  $\{s, t\} \in \binom{V}{2}$ ,  
berechne **kleinsten  $s$ - $t$ -Schnitt** mittels Edmonds-Karp.
- Gib den kleinsten dieser  $s$ - $t$ -Schnitte zurück.

Gesamtlaufzeit:  $O(V^2 \cdot VE^2) \subseteq O(V^7)$ .

Es geht auch in  $O(V^6)$  Zeit...

- Wähle  $s$  fest.
- Berechne den kleinsten  $(s, t)$ -Schnitt für jedes  $t \in V \setminus \{s\}$ .

# Algorithmus CONTRACT

David R. Karger



Ein einfacher **randomisierter** Algorithmus [Karger SODA'93]

# Algorithmus CONTRACT

CONTRACT(zsghd. Multigraph  $G = (V, E)$ )

$H \leftarrow G$

**while**  $H$  hat mehr als zwei Knoten **do**

David R. Karger



Ein einfacher **randomisierter** Algorithmus [Karger SODA'93]

# Algorithmus CONTRACT

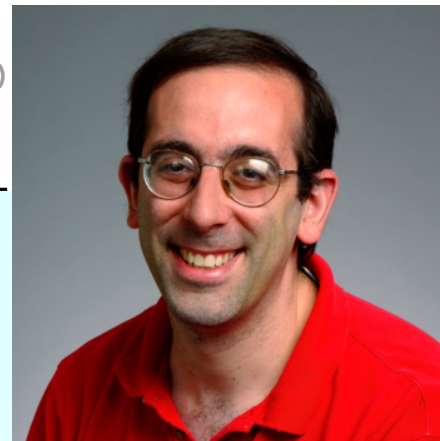
CONTRACT(zsghd. Multigraph  $G = (V, E)$ )

$H \leftarrow G$

**while**  $H$  hat mehr als zwei Knoten **do**

| wähle Kante  $e$  in  $H$  zufällig und gleichverteilt

David R. Karger



Ein einfacher **randomisierter** Algorithmus [Karger SODA'93]



# Algorithmus CONTRACT

CONTRACT(zsghd. Multigraph  $G = (V, E)$ )

$H \leftarrow G$

**while**  $H$  hat mehr als zwei Knoten **do**

    wähle Kante  $e$  in  $H$  zufällig und gleichverteilt

$H \leftarrow H/e$      /\* kontrahiere  $e$  \*/

David R. Karger



Ein einfacher **randomisierter** Algorithmus [Karger SODA'93]

# Algorithmus CONTRACT

CONTRACT(zsghd. Multigraph  $G = (V, E)$ )

$H \leftarrow G$

**while**  $H$  hat mehr als zwei Knoten **do**

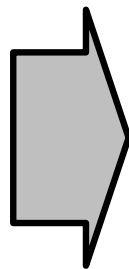
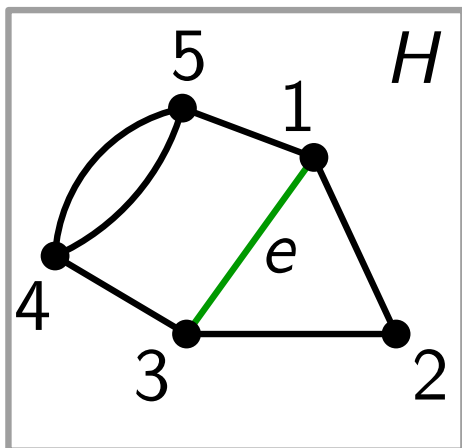
    wähle Kante  $e$  in  $H$  zufällig und gleichverteilt

$H \leftarrow H/e$      /\* kontrahiere  $e$  \*/

David R. Karger



Ein einfacher **randomisierter** Algorithmus [Karger SODA'93]



# Algorithmus CONTRACT

CONTRACT(zsghd. Multigraph  $G = (V, E)$ )

$H \leftarrow G$

**while**  $H$  hat mehr als zwei Knoten **do**

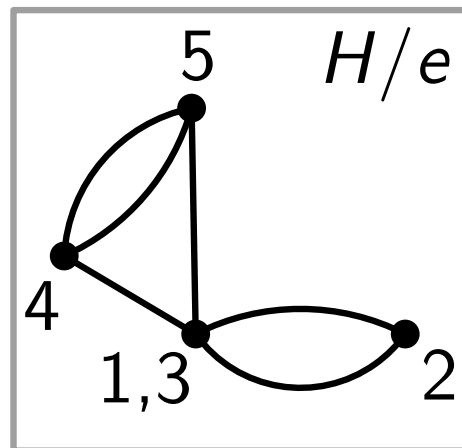
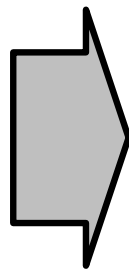
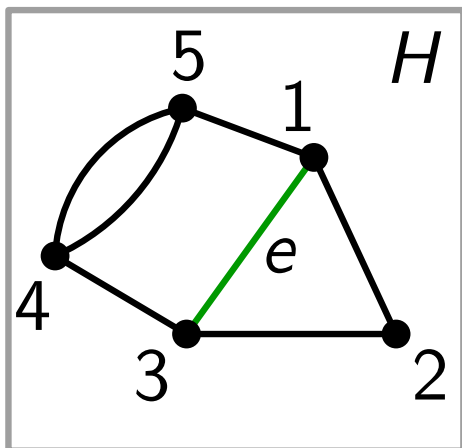
    wähle Kante  $e$  in  $H$  zufällig und gleichverteilt

$H \leftarrow H/e$      /\* kontrahiere  $e$  \*/

David R. Karger



Ein einfacher **randomisierter** Algorithmus [Karger SODA'93]



# Algorithmus CONTRACT

CONTRACT(zsghd. Multigraph  $G = (V, E)$ )

$H \leftarrow G$

**while**  $H$  hat mehr als zwei Knoten **do**

    wähle Kante  $e$  in  $H$  zufällig und gleichverteilt

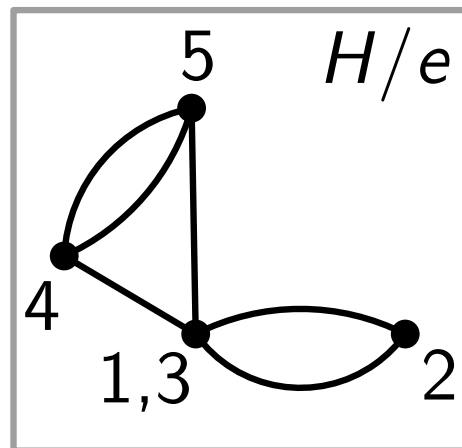
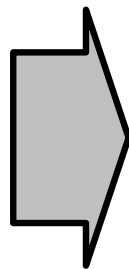
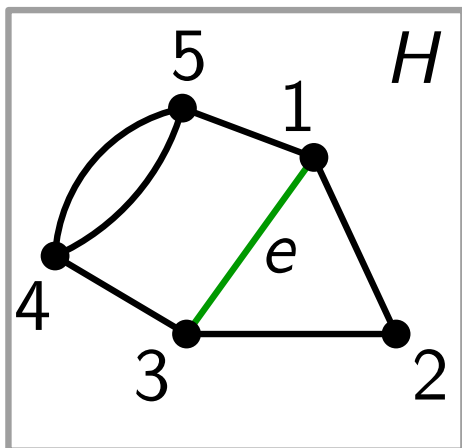
$H \leftarrow H/e$      /\* kontrahiere  $e$  \*/

**return** Zerlegung  $(S, T)$  von  $G$ , die den beiden letzten Knoten in  $H$  entspricht.

David R. Karger



Ein einfacher **randomisierter** Algorithmus [Karger SODA'93]



# Algorithmus CONTRACT

CONTRACT(zsghd. Multigraph  $G = (V, E)$ )

$H \leftarrow G$

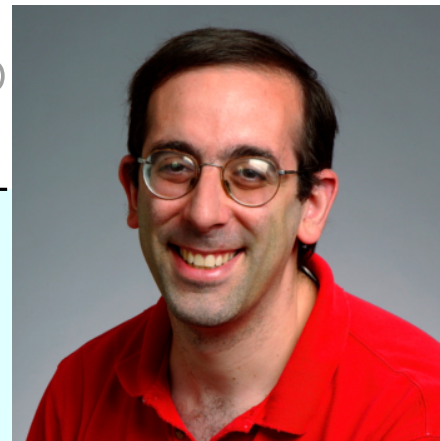
**while**  $H$  hat mehr als zwei Knoten **do**

    wähle Kante  $e$  in  $H$  zufällig und gleichverteilt

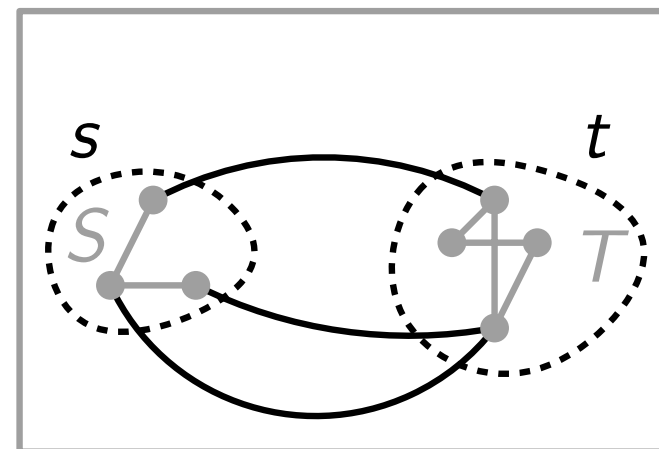
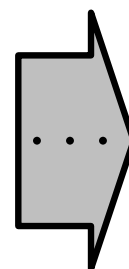
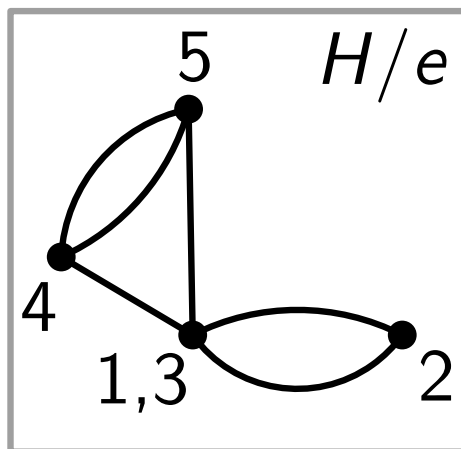
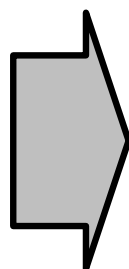
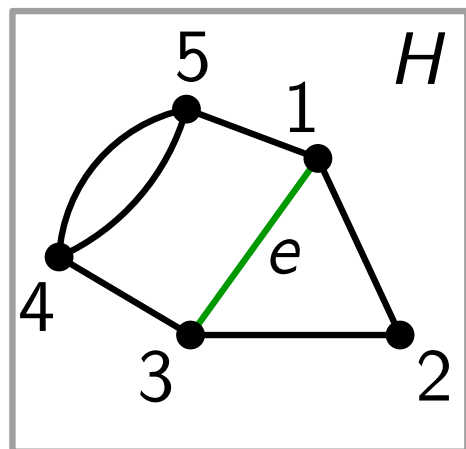
$H \leftarrow H/e$      /\* kontrahiere  $e$  \*/

**return** Zerlegung  $(S, T)$  von  $G$ , die den beiden letzten Knoten in  $H$  entspricht.

David R. Karger



Ein einfacher **randomisierter** Algorithmus [Karger SODA'93]



# Algorithmus CONTRACT

CONTRACT(zsghd. Multigraph  $G = (V, E)$ )

$H \leftarrow G$

**while**  $H$  hat mehr als zwei Knoten **do**

    wähle Kante  $e$  in  $H$  zufällig und gleichverteilt

$H \leftarrow H/e$      /\* kontrahiere  $e$  \*/

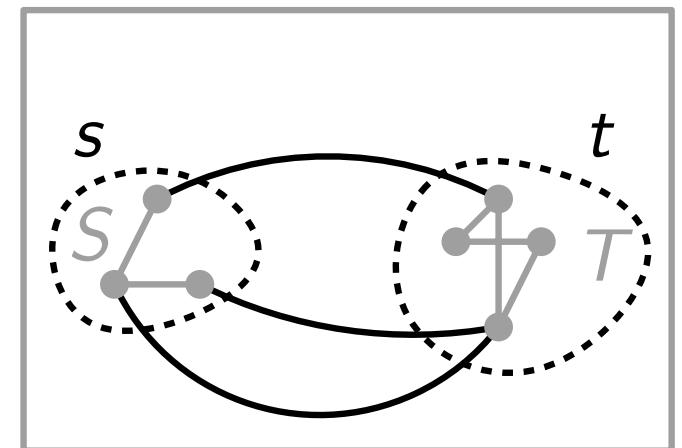
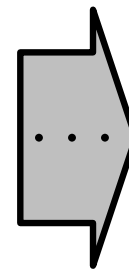
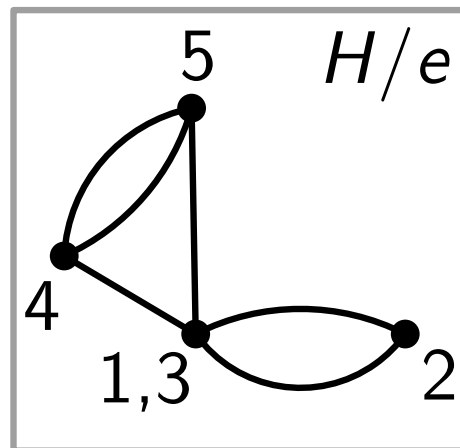
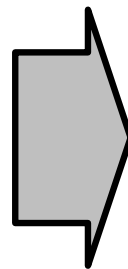
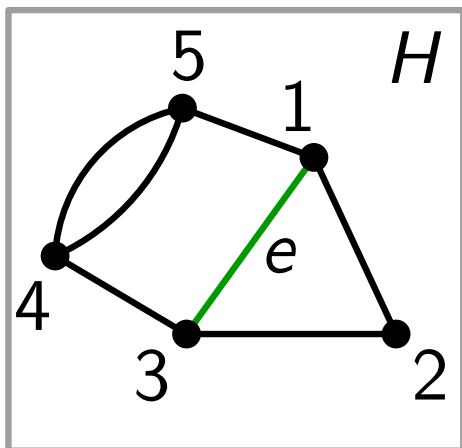
**return** Zerlegung  $(S, T)$  von  $G$ , die den beiden letzten Knoten in  $H$  entspricht.

*vgl. deterministischer Algorithmus!*

David R. Karger



Ein **einfacher** randomisierter Algorithmus [Karger SODA'93]



# Laufzeit

CONTRACT(zsghd. Multigraph  $G = (V, E)$ )

$H \leftarrow G$

**while**  $H$  hat mehr als zwei Knoten **do**

    wähle Kante  $e$  in  $H$  zufällig und gleichverteilt

$H \leftarrow H/e$

**return** Zerlegung  $(S, T)$  von  $G$ , die den beiden letzten Knoten in  $H$  entspricht.

# Laufzeit

CONTRACT(zsghd. Multigraph  $G = (V, E)$ )

$H \leftarrow G$

**while**  $H$  hat mehr als zwei Knoten **do**

    wähle Kante  $e$  in  $H$  zufällig und gleichverteilt

$H \leftarrow H/e$

**return** Zerlegung  $(S, T)$  von  $G$ , die den beiden letzten Knoten in  $H$  entspricht.

- Anzahl der Knoten sinkt in jeder Iteration um 1.



# Laufzeit

CONTRACT(zsghd. Multigraph  $G = (V, E)$ )

$H \leftarrow G$

**while**  $H$  hat mehr als zwei Knoten **do**

    wähle Kante  $e$  in  $H$  zufällig und gleichverteilt

$H \leftarrow H/e$

**return** Zerlegung  $(S, T)$  von  $G$ , die den beiden letzten Knoten in  $H$  entspricht.

- Anzahl der Knoten sinkt in jeder Iteration um 1.
- Jede Iteration benötigt  $O(E)$  Zeit.

# Laufzeit

CONTRACT(zsghd. Multigraph  $G = (V, E)$ )

$H \leftarrow G$

**while**  $H$  hat mehr als zwei Knoten **do**

    wähle Kante  $e$  in  $H$  zufällig und gleichverteilt

$H \leftarrow H/e$

**return** Zerlegung  $(S, T)$  von  $G$ , die den beiden letzten Knoten in  $H$  entspricht.

- Anzahl der Knoten sinkt in jeder Iteration um 1.
- Jede Iteration benötigt  $O(E)$  Zeit.

Übg.: Implementierung mit  $O(V)$  Zeit pro Iteration möglich!

# Laufzeit

CONTRACT(zsghd. Multigraph  $G = (V, E)$ )

$H \leftarrow G$

**while**  $H$  hat mehr als zwei Knoten **do**

    wähle Kante  $e$  in  $H$  zufällig und gleichverteilt

$H \leftarrow H/e$

**return** Zerlegung  $(S, T)$  von  $G$ , die den beiden letzten Knoten in  $H$  entspricht.

- Anzahl der Knoten sinkt in jeder Iteration um 1.
- Jede Iteration benötigt  $O(E)$  Zeit.

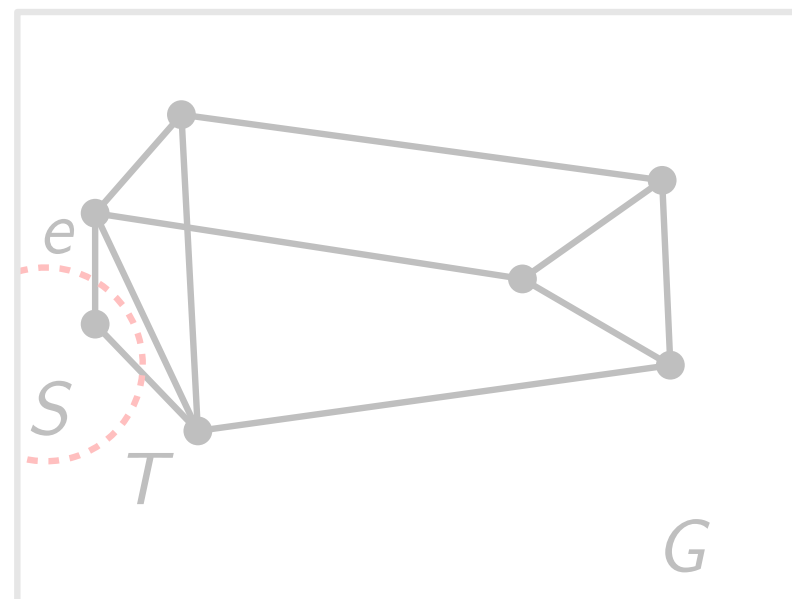
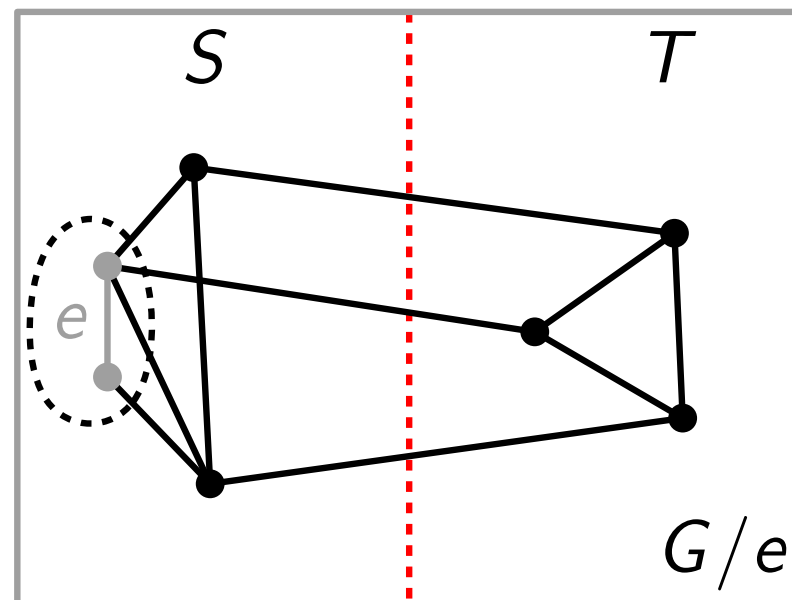
Übg.: Implementierung mit  $O(V)$  Zeit pro Iteration möglich!

---

Gesamtlaufzeit:  $O(V^2)$

# Beobachtung Kantenkontraktion

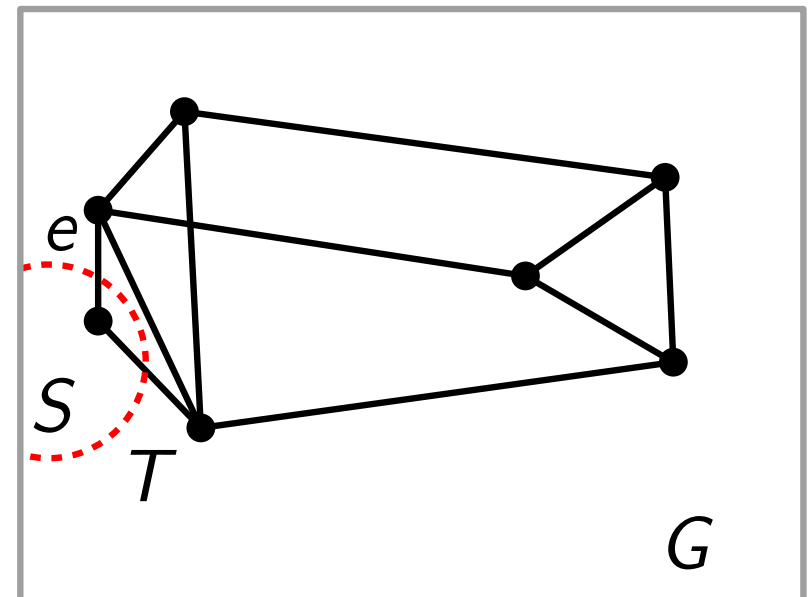
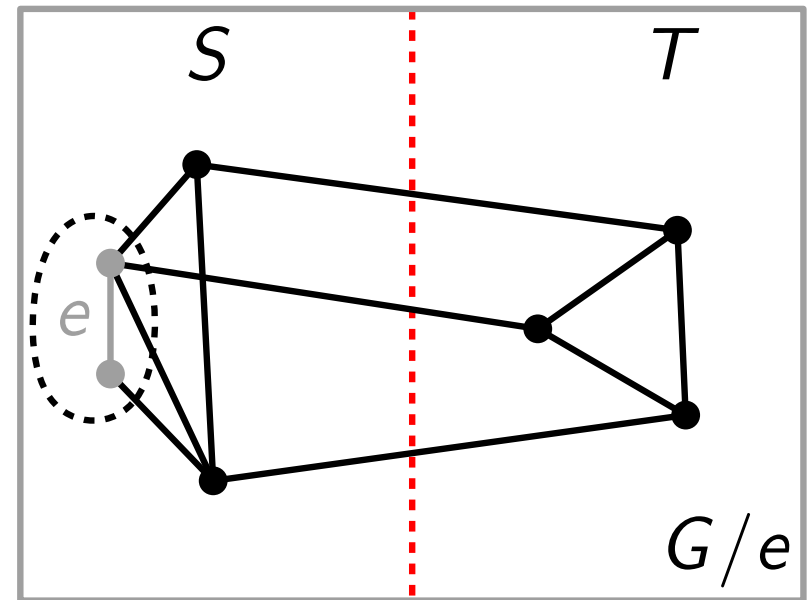
Jeder Schnitt in  $G/e$  korrespondiert zu Schnitt gleicher Größe in  $G$ .



# Beobachtung Kantenkontraktion

Jeder Schnitt in  $G/e$  korrespondiert zu Schnitt gleicher Größe in  $G$ .

$\rightsquigarrow$  Größe eines kleinsten Schnitts steigt (schwach) monoton während Ausführung von CONTRACT



# Erfolgswahrscheinlichkeit – Einschätzung

**Satz.** Sei  $(S, T)$  ein kleinster Schnitt. Die Wahrscheinlichkeit, dass CONTRACT diesen Schnitt findet, ist  $\geq \frac{2}{n(n-1)}$ .

# Erfolgswahrscheinlichkeit – Einschätzung

**Satz.** Sei  $(S, T)$  ein kleinster Schnitt. Die Wahrscheinlichkeit, dass CONTRACT diesen Schnitt findet, ist  $\geq \frac{2}{n(n-1)}$ .

Diese Wahrscheinlichkeit ist **überraschend hoch**, da es  
viele Schnitte gibt.

# Erfolgswahrscheinlichkeit – Einschätzung

**Satz.** Sei  $(S, T)$  ein kleinster Schnitt. Die Wahrscheinlichkeit, dass CONTRACT diesen Schnitt findet, ist  $\geq \frac{2}{n(n-1)}$ .

Diese Wahrscheinlichkeit ist **überraschend hoch**, da es  $(2^n - 2)/2 = 2^{n-1} - 1$  viele Schnitte gibt.



# Erfolgswahrscheinlichkeit – Einschätzung

**Satz.** Sei  $(S, T)$  ein kleinster Schnitt. Die Wahrscheinlichkeit, dass CONTRACT diesen Schnitt findet, ist  $\geq \frac{2}{n(n-1)}$ .

Diese Wahrscheinlichkeit ist **überraschend hoch**, da es  $(2^n - 2)/2 = 2^{n-1} - 1$  viele Schnitte gibt.

Die Wahrscheinlichkeit, dass ein zufällig und gleichverteilt gewählter Schnitt genau  $(S, T)$  trifft, ist also nur  $\frac{1}{2^{\Omega(n)}}$ , also **exponentiell klein**.

# Erfolgswahrscheinlichkeit – Beweis

**Satz.** Sei  $(S, T)$  ein kleinster Schnitt. Die Wahrscheinlichkeit, dass CONTRACT diesen Schnitt findet, ist  $\geq \frac{2}{n(n-1)}$ .

## Beweis.

Sei  $(S, T)$  ein kleinster Schnitt,  $C = \{ uv \in E \mid u \in S, v \in T \}$  und  $k = |C|$ .

# Erfolgswahrscheinlichkeit – Beweis

**Satz.** Sei  $(S, T)$  ein kleinster Schnitt. Die Wahrscheinlichkeit, dass CONTRACT diesen Schnitt findet, ist  $\geq \frac{2}{n(n-1)}$ .

## Beweis.

Sei  $(S, T)$  ein kleinster Schnitt,  $C = \{ uv \in E \mid u \in S, v \in T \}$  und  $k = |C|$ .

Graph  $H_i$  ( $H$  zu Beginn von Iteration  $i$ ) hat  $n - i + 1$  Knoten.

# Erfolgswahrscheinlichkeit – Beweis

**Satz.** Sei  $(S, T)$  ein kleinster Schnitt. Die Wahrscheinlichkeit, dass CONTRACT diesen Schnitt findet, ist  $\geq \frac{2}{n(n-1)}$ .

## Beweis.

Sei  $(S, T)$  ein kleinster Schnitt,  $C = \{ uv \in E \mid u \in S, v \in T \}$  und  $k = |C|$ .

Graph  $H_i$  ( $H$  zu Beginn von Iteration  $i$ ) hat  $n - i + 1$  Knoten.

Kleinster Schnitt in  $H_i$  hat Größe  $\geq k$ .

# Erfolgswahrscheinlichkeit – Beweis

**Satz.** Sei  $(S, T)$  ein kleinster Schnitt. Die Wahrscheinlichkeit, dass CONTRACT diesen Schnitt findet, ist  $\geq \frac{2}{n(n-1)}$ .

## Beweis.

Sei  $(S, T)$  ein kleinster Schnitt,  $C = \{ uv \in E \mid u \in S, v \in T \}$  und  $k = |C|$ .

Graph  $H_i$  ( $H$  zu Beginn von Iteration  $i$ ) hat  $n - i + 1$  Knoten.

Kleinster Schnitt in  $H_i$  hat Größe  $\geq k$ .

$\Rightarrow$  Jeder Knoten  $u$  in  $H_i$  hat Grad  $\geq k$ .

# Erfolgswahrscheinlichkeit – Beweis

**Satz.** Sei  $(S, T)$  ein kleinster Schnitt. Die Wahrscheinlichkeit, dass CONTRACT diesen Schnitt findet, ist  $\geq \frac{2}{n(n-1)}$ .

## Beweis.

Sei  $(S, T)$  ein kleinster Schnitt,  $C = \{ uv \in E \mid u \in S, v \in T \}$  und  $k = |C|$ .

Graph  $H_i$  ( $H$  zu Beginn von Iteration  $i$ ) hat  $n - i + 1$  Knoten.

Kleinster Schnitt in  $H_i$  hat Größe  $\geq k$ .

$\Rightarrow$  Jeder Knoten  $u$  in  $H_i$  hat Grad  $\geq k$ .

sonst hätte  
Schnitt  $(u, V - u)$   
Größe  $< k$

# Erfolgswahrscheinlichkeit – Beweis

**Satz.** Sei  $(S, T)$  ein kleinster Schnitt. Die Wahrscheinlichkeit, dass CONTRACT diesen Schnitt findet, ist  $\geq \frac{2}{n(n-1)}$ .

## Beweis.

Sei  $(S, T)$  ein kleinster Schnitt,  $C = \{ uv \in E \mid u \in S, v \in T \}$  und  $k = |C|$ .

Graph  $H_i$  ( $H$  zu Beginn von Iteration  $i$ ) hat  $n - i + 1$  Knoten.

Kleinster Schnitt in  $H_i$  hat Größe  $\geq k$ .

$\Rightarrow$  Jeder Knoten  $u$  in  $H_i$  hat Grad  $\geq k$ .

sonst hätte  
Schnitt  $(u, V - u)$   
Größe  $< k$

$\Rightarrow H_i$  hat  $\geq$  Kanten.

# Erfolgswahrscheinlichkeit – Beweis

**Satz.** Sei  $(S, T)$  ein kleinster Schnitt. Die Wahrscheinlichkeit, dass CONTRACT diesen Schnitt findet, ist  $\geq \frac{2}{n(n-1)}$ .

## Beweis.

Sei  $(S, T)$  ein kleinster Schnitt,  $C = \{ uv \in E \mid u \in S, v \in T \}$  und  $k = |C|$ .

Graph  $H_i$  ( $H$  zu Beginn von Iteration  $i$ ) hat  $n - i + 1$  Knoten.

Kleinster Schnitt in  $H_i$  hat Größe  $\geq k$ .

$\Rightarrow$  Jeder Knoten  $u$  in  $H_i$  hat Grad  $\geq k$ .

sonst hätte  
Schnitt  $(u, V - u)$   
Größe  $< k$

$\Rightarrow H_i$  hat  $\geq k(n - i + 1)/2$  Kanten.



# Erfolgswahrscheinlichkeit – Beweis

**Satz.** Sei  $(S, T)$  ein kleinster Schnitt. Die Wahrscheinlichkeit, dass CONTRACT diesen Schnitt findet, ist  $\geq \frac{2}{n(n-1)}$ .

**Beweis.** (Forts.)

$H_i$  hat  $\geq k(n - i + 1)/2$  Kanten.

# Erfolgswahrscheinlichkeit – Beweis

**Satz.** Sei  $(S, T)$  ein kleinster Schnitt. Die Wahrscheinlichkeit, dass CONTRACT diesen Schnitt findet, ist  $\geq \frac{2}{n(n-1)}$ .

**Beweis.** (Forts.)

$H_i$  hat  $\geq k(n - i + 1)/2$  Kanten.

Sei  $\mathcal{E}_i$  Ereignis, dass in Iteration  $i$  in  $H_i$  keine Kante aus  $C$  kontrahiert wird.

# Erfolgswahrscheinlichkeit – Beweis

**Satz.** Sei  $(S, T)$  ein kleinster Schnitt. Die Wahrscheinlichkeit, dass CONTRACT diesen Schnitt findet, ist  $\geq \frac{2}{n(n-1)}$ .

**Beweis.** (Forts.)

$H_i$  hat  $\geq k(n - i + 1)/2$  Kanten.

Sei  $\mathcal{E}_i$  Ereignis, dass in Iteration  $i$  in  $H_i$  keine Kante aus  $C$  kontrahiert wird.

WK, dass in Iteration  $i$  keine Kante aus  $C$  kontrahiert wird (vorausgesetzt es wurde bislang noch keine solche kontrahiert):

# Erfolgswahrscheinlichkeit – Beweis

**Satz.** Sei  $(S, T)$  ein kleinster Schnitt. Die Wahrscheinlichkeit, dass CONTRACT diesen Schnitt findet, ist  $\geq \frac{2}{n(n-1)}$ .

**Beweis.** (Forts.)

$H_i$  hat  $\geq k(n - i + 1)/2$  Kanten.

Sei  $\mathcal{E}_i$  Ereignis, dass in Iteration  $i$  in  $H_i$  keine Kante aus  $C$  kontrahiert wird.

WK, dass in Iteration  $i$  keine Kante aus  $C$  kontrahiert wird (vorausgesetzt es wurde bislang noch keine solche kontrahiert):

$$\Pr \left[ \mathcal{E}_i \mid \bigcap_{j=1}^{i-1} \mathcal{E}_j \right] \geq 1 - \frac{2}{n(n-1)}$$

# Erfolgswahrscheinlichkeit – Beweis

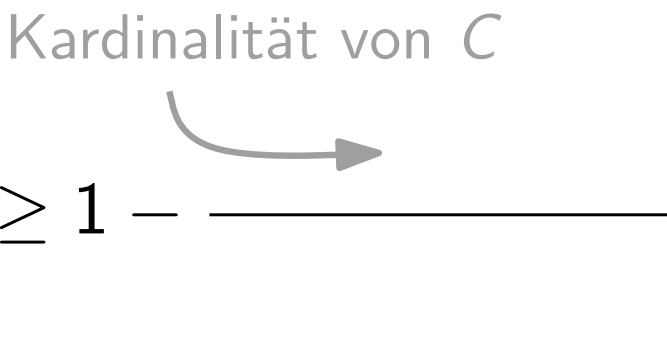
**Satz.** Sei  $(S, T)$  ein kleinster Schnitt. Die Wahrscheinlichkeit, dass CONTRACT diesen Schnitt findet, ist  $\geq \frac{2}{n(n-1)}$ .

**Beweis.** (Forts.)

$H_i$  hat  $\geq k(n - i + 1)/2$  Kanten.

Sei  $\mathcal{E}_i$  Ereignis, dass in Iteration  $i$  in  $H_i$  keine Kante aus  $C$  kontrahiert wird.

WK, dass in Iteration  $i$  keine Kante aus  $C$  kontrahiert wird (vorausgesetzt es wurde bislang noch keine solche kontrahiert):

$$\Pr \left[ \mathcal{E}_i \mid \bigcap_{j=1}^{i-1} \mathcal{E}_j \right] \geq 1 - \frac{\text{Kardinalität von } C}{\dots}$$


# Erfolgswahrscheinlichkeit – Beweis

**Satz.** Sei  $(S, T)$  ein kleinster Schnitt. Die Wahrscheinlichkeit, dass CONTRACT diesen Schnitt findet, ist  $\geq \frac{2}{n(n-1)}$ .

**Beweis.** (Forts.)

$H_i$  hat  $\geq k(n - i + 1)/2$  Kanten.

Sei  $\mathcal{E}_i$  Ereignis, dass in Iteration  $i$  in  $H_i$  keine Kante aus  $C$  kontrahiert wird.

WK, dass in Iteration  $i$  keine Kante aus  $C$  kontrahiert wird (vorausgesetzt es wurde bislang noch keine solche kontrahiert):

$$\Pr \left[ \mathcal{E}_i \mid \bigcap_{j=1}^{i-1} \mathcal{E}_j \right] \geq 1 - \frac{\text{Kardinalität von } C}{\text{Gesamtzahl der Kanten in } H_i}$$

# Erfolgswahrscheinlichkeit – Beweis

**Satz.** Sei  $(S, T)$  ein kleinster Schnitt. Die Wahrscheinlichkeit, dass CONTRACT diesen Schnitt findet, ist  $\geq \frac{2}{n(n-1)}$ .

**Beweis.** (Forts.)

$H_i$  hat  $\geq k(n - i + 1)/2$  Kanten.

Sei  $\mathcal{E}_i$  Ereignis, dass in Iteration  $i$  in  $H_i$  keine Kante aus  $C$  kontrahiert wird.

WK, dass in Iteration  $i$  keine Kante aus  $C$  kontrahiert wird (vorausgesetzt es wurde bislang noch keine solche kontrahiert):

$$\Pr \left[ \mathcal{E}_i \mid \bigcap_{j=1}^{i-1} \mathcal{E}_j \right] \geq 1 - \frac{\overset{\text{Kardinalität von } C}{k}}{\overset{\text{Gesamtzahl der Kanten in } H_i}{k(n - i + 1)/2}} =$$

# Erfolgswahrscheinlichkeit – Beweis

**Satz.** Sei  $(S, T)$  ein kleinster Schnitt. Die Wahrscheinlichkeit, dass CONTRACT diesen Schnitt findet, ist  $\geq \frac{2}{n(n-1)}$ .

**Beweis.** (Forts.)

$H_i$  hat  $\geq k(n - i + 1)/2$  Kanten.

Sei  $\mathcal{E}_i$  Ereignis, dass in Iteration  $i$  in  $H_i$  keine Kante aus  $C$  kontrahiert wird.

WK, dass in Iteration  $i$  keine Kante aus  $C$  kontrahiert wird (vorausgesetzt es wurde bislang noch keine solche kontrahiert):

$$\Pr \left[ \mathcal{E}_i \mid \bigcap_{j=1}^{i-1} \mathcal{E}_j \right] \geq 1 - \frac{\text{Kardinalität von } C}{\text{Gesamtzahl der Kanten in } H_i} = 1 - \frac{k}{k(n - i + 1)/2} = 1 - \frac{2}{n - i + 1}$$



# Erfolgswahrscheinlichkeit – Beweis

**Satz.** Sei  $(S, T)$  ein kleinster Schnitt. Die Wahrscheinlichkeit, dass CONTRACT diesen Schnitt findet, ist  $\geq \frac{2}{n(n-1)}$ .

**Beweis.**

$$\Pr \left[ \mathcal{E}_i \mid \bigcap_{j=1}^{i-1} \mathcal{E}_j \right] \geq 1 - \frac{2}{n - i + 1}$$

# Erfolgswahrscheinlichkeit – Beweis

**Satz.** Sei  $(S, T)$  ein kleinster Schnitt. Die Wahrscheinlichkeit, dass CONTRACT diesen Schnitt findet, ist  $\geq \frac{2}{n(n-1)}$ .

**Beweis.**

$$\Pr \left[ \mathcal{E}_i \mid \bigcap_{j=1}^{i-1} \mathcal{E}_j \right] \geq 1 - \frac{2}{n - i + 1}$$

Wahrscheinlichkeit, dass CONTRACT *keine* Kante aus  $C$  kontrahiert:

# Erfolgswahrscheinlichkeit – Beweis

**Satz.** Sei  $(S, T)$  ein kleinster Schnitt. Die Wahrscheinlichkeit, dass CONTRACT diesen Schnitt findet, ist  $\geq \frac{2}{n(n-1)}$ .

**Beweis.**

$$\Pr \left[ \mathcal{E}_i \mid \bigcap_{j=1}^{i-1} \mathcal{E}_j \right] \geq 1 - \frac{2}{n-i+1}$$

Wahrscheinlichkeit, dass CONTRACT *keine* Kante aus  $C$  kontrahiert:

$$\Pr \left[ \bigcap_{i=1}^{n-2} \mathcal{E}_i \right] \geq$$

# Erfolgswahrscheinlichkeit – Beweis

**Satz.** Sei  $(S, T)$  ein kleinster Schnitt. Die Wahrscheinlichkeit, dass CONTRACT diesen Schnitt findet, ist  $\geq \frac{2}{n(n-1)}$ .

**Beweis.**

$$\Pr \left[ \mathcal{E}_i \mid \bigcap_{j=1}^{i-1} \mathcal{E}_j \right] \geq 1 - \frac{2}{n-i+1}$$

Wahrscheinlichkeit, dass CONTRACT *keine* Kante aus  $C$  kontrahiert:

$$\Pr \left[ \bigcap_{i=1}^{n-2} \mathcal{E}_i \right] \geq \prod_{i=1}^{n-2} \left( 1 - \frac{2}{n-i+1} \right) =$$

# Erfolgswahrscheinlichkeit – Beweis

**Satz.** Sei  $(S, T)$  ein kleinster Schnitt. Die Wahrscheinlichkeit, dass CONTRACT diesen Schnitt findet, ist  $\geq \frac{2}{n(n-1)}$ .

**Beweis.**

$$\Pr \left[ \mathcal{E}_i \mid \bigcap_{j=1}^{i-1} \mathcal{E}_j \right] \geq 1 - \frac{2}{n-i+1}$$

Wahrscheinlichkeit, dass CONTRACT *keine* Kante aus  $C$  kontrahiert:

$$\Pr \left[ \bigcap_{i=1}^{n-2} \mathcal{E}_i \right] \geq \prod_{i=1}^{n-2} \left( 1 - \frac{2}{n-i+1} \right) = \prod_{i=1}^{n-2} \frac{n-i-1}{n-i+1}$$

# Erfolgswahrscheinlichkeit – Beweis

**Satz.** Sei  $(S, T)$  ein kleinster Schnitt. Die Wahrscheinlichkeit, dass CONTRACT diesen Schnitt findet, ist  $\geq \frac{2}{n(n-1)}$ .

**Beweis.**

$$\Pr \left[ \mathcal{E}_i \mid \bigcap_{j=1}^{i-1} \mathcal{E}_j \right] \geq 1 - \frac{2}{n-i+1}$$

Wahrscheinlichkeit, dass CONTRACT *keine* Kante aus  $C$  kontrahiert:

$$\begin{aligned} \Pr \left[ \bigcap_{i=1}^{n-2} \mathcal{E}_i \right] &\geq \prod_{i=1}^{n-2} \left( 1 - \frac{2}{n-i+1} \right) = \prod_{i=1}^{n-2} \frac{n-i-1}{n-i+1} \\ &= \frac{(n-2) \cdot (n-3) \cdot \dots \cdot 3 \cdot 2 \cdot 1}{n \cdot (n-1) \cdot (n-2) \cdot \dots \cdot 3} = \end{aligned}$$

# Erfolgswahrscheinlichkeit – Beweis

**Satz.** Sei  $(S, T)$  ein kleinster Schnitt. Die Wahrscheinlichkeit, dass CONTRACT diesen Schnitt findet, ist  $\geq \frac{2}{n(n-1)}$ .

**Beweis.**

$$\Pr \left[ \mathcal{E}_i \mid \bigcap_{j=1}^{i-1} \mathcal{E}_j \right] \geq 1 - \frac{2}{n-i+1}$$

Wahrscheinlichkeit, dass CONTRACT *keine* Kante aus  $C$  kontrahiert:

$$\begin{aligned} \Pr \left[ \bigcap_{i=1}^{n-2} \mathcal{E}_i \right] &\geq \prod_{i=1}^{n-2} \left( 1 - \frac{2}{n-i+1} \right) = \prod_{i=1}^{n-2} \frac{n-i-1}{n-i+1} \\ &= \frac{\cancel{(n-2)} \cdot \cancel{(n-3)} \cdot \dots \cdot \cancel{3} \cdot 2 \cdot 1}{n \cdot (n-1) \cdot \cancel{(n-2)} \cdot \dots \cdot \cancel{3}} = \end{aligned}$$

# Erfolgswahrscheinlichkeit – Beweis

**Satz.** Sei  $(S, T)$  ein kleinster Schnitt. Die Wahrscheinlichkeit, dass CONTRACT diesen Schnitt findet, ist  $\geq \frac{2}{n(n-1)}$ .

**Beweis.**

$$\Pr \left[ \mathcal{E}_i \mid \bigcap_{j=1}^{i-1} \mathcal{E}_j \right] \geq 1 - \frac{2}{n-i+1}$$

Wahrscheinlichkeit, dass CONTRACT *keine* Kante aus  $C$  kontrahiert:

$$\begin{aligned} \Pr \left[ \bigcap_{i=1}^{n-2} \mathcal{E}_i \right] &\geq \prod_{i=1}^{n-2} \left( 1 - \frac{2}{n-i+1} \right) = \prod_{i=1}^{n-2} \frac{n-i-1}{n-i+1} \\ &= \frac{\cancel{(n-2)} \cdot \cancel{(n-3)} \cdot \dots \cdot \cancel{3} \cdot 2 \cdot 1}{n \cdot (n-1) \cdot \cancel{(n-2)} \cdot \dots \cdot \cancel{3}} = \frac{2}{n(n-1)} \quad \square \end{aligned}$$



# Verringerung der Fehlerwahrscheinlichkeit

Erfolgswahrscheinlichkeit  $\frac{2}{n(n-1)} \geq \frac{2}{n^2}$  zu klein für praktische Zwecke.

# Verringerung der Fehlerwahrscheinlichkeit

Erfolgswahrscheinlichkeit  $\frac{2}{n(n-1)} \geq \frac{2}{n^2}$  zu klein für praktische Zwecke.

⇒ Wiederhole  $n^2 \ln n$  mal und gib beste Lösung aus.

# Verringerung der Fehlerwahrscheinlichkeit

Erfolgswahrscheinlichkeit  $\frac{2}{n(n-1)} \geq \frac{2}{n^2}$  zu klein für praktische Zwecke.

⇒ Wiederhole  $n^2 \ln n$  mal und gib beste Lösung aus.

**Fehler**wahrscheinlichkeit ist höchstens

$$\left(1 - \frac{2}{n^2}\right)$$

# Verringerung der Fehlerwahrscheinlichkeit

Erfolgswahrscheinlichkeit  $\frac{2}{n(n-1)} \geq \frac{2}{n^2}$  zu klein für praktische Zwecke.

⇒ Wiederhole  $n^2 \ln n$  mal und gib beste Lösung aus.

**Fehler**wahrscheinlichkeit ist höchstens

$$\left(1 - \frac{2}{n^2}\right)^{n^2 \ln n} \leq$$

# Verringerung der Fehlerwahrscheinlichkeit

Erfolgswahrscheinlichkeit  $\frac{2}{n(n-1)} \geq \frac{2}{n^2}$  zu klein für praktische Zwecke.

⇒ Wiederhole  $n^2 \ln n$  mal und gib beste Lösung aus.

**Fehler**wahrscheinlichkeit ist höchstens

$$\left(1 - \frac{2}{n^2}\right)^{n^2 \ln n} \leq$$

$$1 + x \leq e^x$$

für alle  $x \in \mathbb{R}$

# Verringerung der Fehlerwahrscheinlichkeit

Erfolgswahrscheinlichkeit  $\frac{2}{n(n-1)} \geq \frac{2}{n^2}$  zu klein für praktische Zwecke.

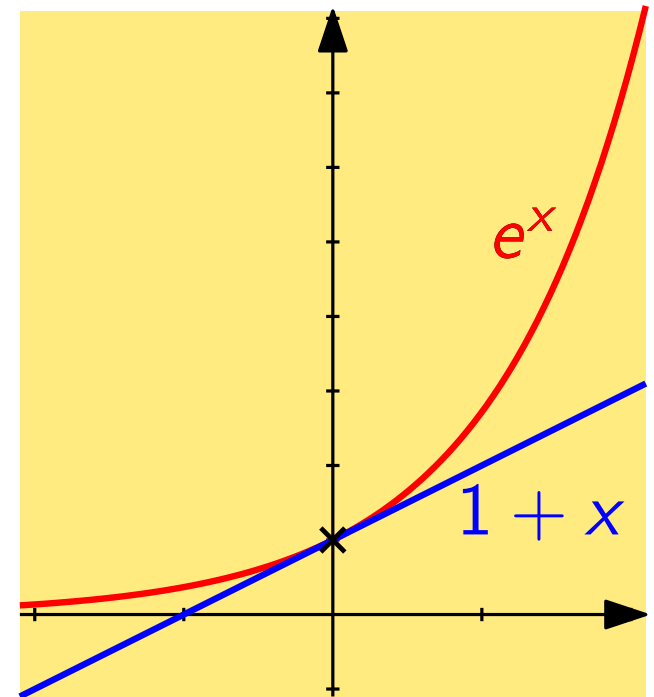
⇒ Wiederhole  $n^2 \ln n$  mal und gib beste Lösung aus.

**Fehler**wahrscheinlichkeit ist höchstens

$$\left(1 - \frac{2}{n^2}\right)^{n^2 \ln n} \leq$$

$$1 + x \leq e^x$$

für alle  $x \in \mathbb{R}$



# Verringerung der Fehlerwahrscheinlichkeit

Erfolgswahrscheinlichkeit  $\frac{2}{n(n-1)} \geq \frac{2}{n^2}$  zu klein für praktische Zwecke.

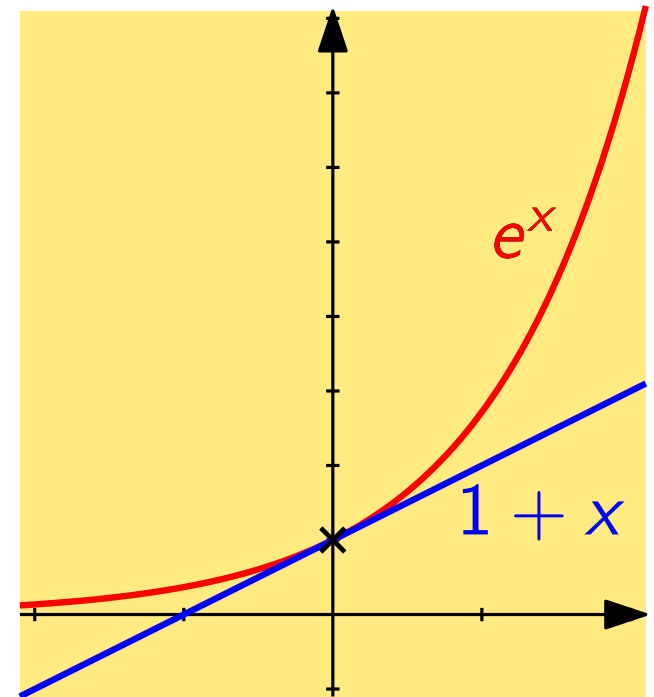
⇒ Wiederhole  $n^2 \ln n$  mal und gib beste Lösung aus.

**Fehler**wahrscheinlichkeit ist höchstens

$$\left(1 - \frac{2}{n^2}\right)^{n^2 \ln n} \leq e^{-\frac{2n^2 \ln n}{n^2}} =$$

$$1 + x \leq e^x$$

für alle  $x \in \mathbb{R}$



# Verringerung der Fehlerwahrscheinlichkeit

Erfolgswahrscheinlichkeit  $\frac{2}{n(n-1)} \geq \frac{2}{n^2}$  zu klein für praktische Zwecke.

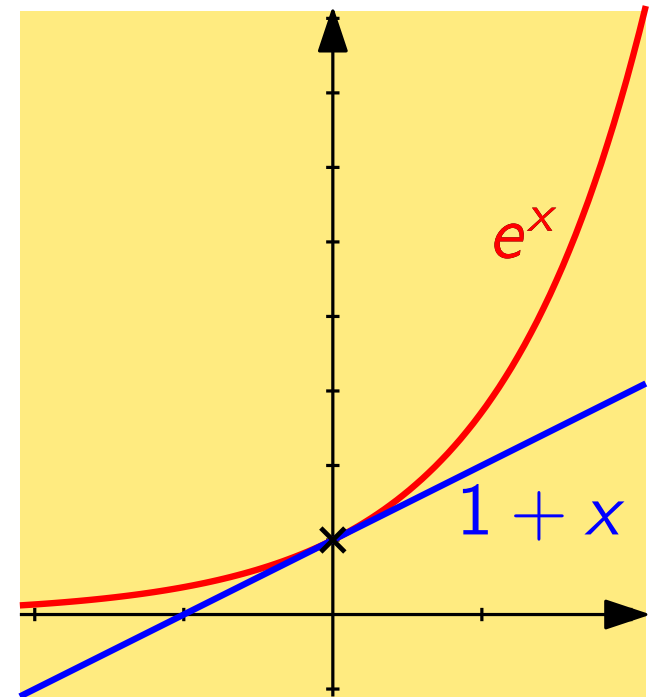
⇒ Wiederhole  $n^2 \ln n$  mal und gib beste Lösung aus.

**Fehler**wahrscheinlichkeit ist höchstens

$$\left(1 - \frac{2}{n^2}\right)^{n^2 \ln n} \leq e^{-\frac{2n^2 \ln n}{n^2}} = e^{-2 \ln n} =$$

$$1 + x \leq e^x$$

für alle  $x \in \mathbb{R}$





# Verringerung der Fehlerwahrscheinlichkeit

Erfolgswahrscheinlichkeit  $\frac{2}{n(n-1)} \geq \frac{2}{n^2}$  zu klein für praktische Zwecke.

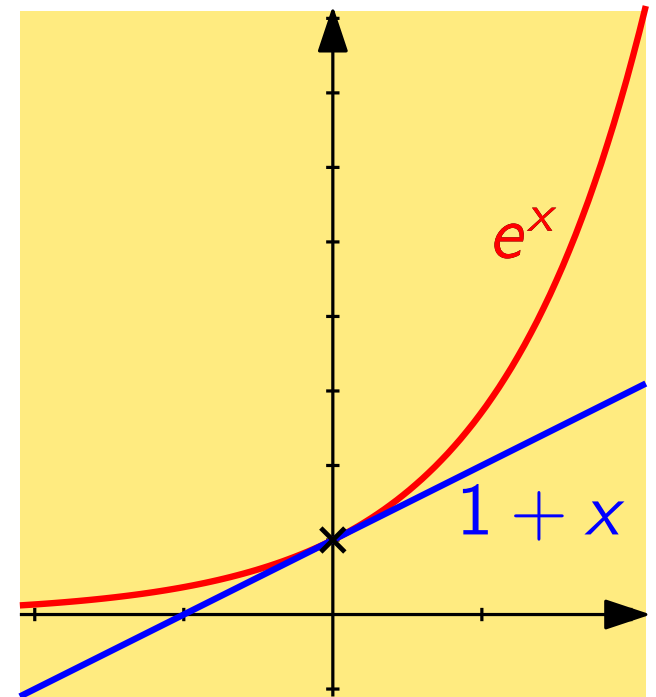
⇒ Wiederhole  $n^2 \ln n$  mal und gib beste Lösung aus.

**Fehler**wahrscheinlichkeit ist höchstens

$$\left(1 - \frac{2}{n^2}\right)^{n^2 \ln n} \leq e^{-\frac{2n^2 \ln n}{n^2}} = e^{-2 \ln n} = \frac{1}{n^2}$$

$$1 + x \leq e^x$$

für alle  $x \in \mathbb{R}$



# Verringerung der Fehlerwahrscheinlichkeit

Erfolgswahrscheinlichkeit  $\frac{2}{n(n-1)} \geq \frac{2}{n^2}$  zu klein für praktische Zwecke.

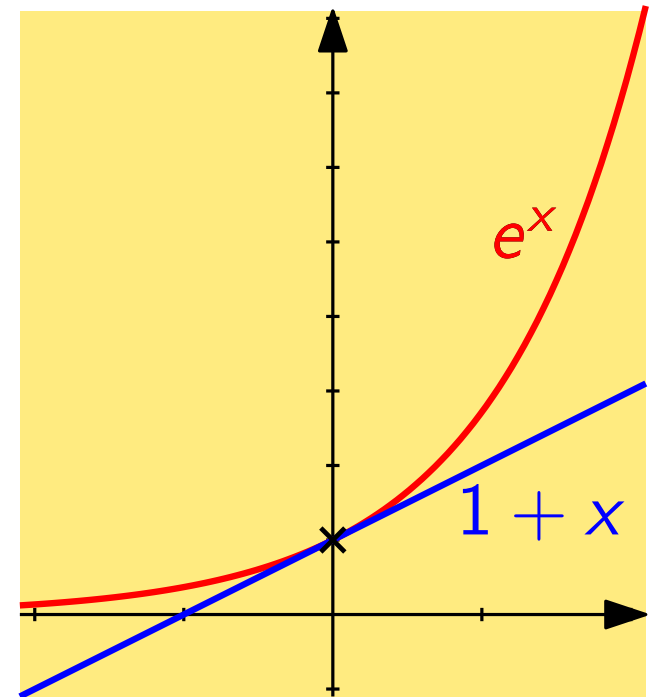
⇒ Wiederhole  $n^2 \ln n$  mal und gib beste Lösung aus.

**Fehler**wahrscheinlichkeit ist höchstens

$$\left(1 - \frac{2}{n^2}\right)^{n^2 \ln n} \leq e^{-\frac{2n^2 \ln n}{n^2}} = e^{-2 \ln n} = \frac{1}{n^2}$$

$$1 + x \leq e^x$$

für alle  $x \in \mathbb{R}$



⇒ **Erfolgswahrscheinlichkeit**  $1 - \frac{1}{n^2}$  konvergiert schnell gegen 1.

# Resümee

Es handelt sich um einen sog. **Monte-Carlo-Algorithmus**, da randomisiert und nicht immer korrekt.

Eine andere Art randomisierter Algorithmen sind sog. **Las-Vegas-Algorithmen**, bei denen das Ergebnis immer korrekt ist, aber die Laufzeit eine Zufallsvariable ist (Bsp. RandomizedQuickSort in der ADS).

# Resümee

Es handelt sich um einen sog. **Monte-Carlo-Algorithmus**, da randomisiert und nicht immer korrekt.

Eine andere Art randomisierter Algorithmen sind sog. **Las-Vegas-Algorithmen**, bei denen das Ergebnis immer korrekt ist, aber die Laufzeit eine Zufallsvariable ist (Bsp. RandomizedQuickSort in der ADS).

Aber: Der Alg. ist mit **hoher Wahrscheinlichkeit** korrekt; Fehlerwahrscheinlichkeit konvergiert mit steigendem  $n$  gegen 0.

# Resümee

Es handelt sich um einen sog. **Monte-Carlo-Algorithmus**, da randomisiert und nicht immer korrekt.

Eine andere Art randomisierter Algorithmen sind sog. **Las-Vegas-Algorithmen**, bei denen das Ergebnis immer korrekt ist, aber die Laufzeit eine Zufallsvariable ist (Bsp. RandomizedQuickSort in der ADS).

Aber: Der Alg. ist mit **hoher Wahrscheinlichkeit** korrekt; Fehlerwahrscheinlichkeit konvergiert mit steigendem  $n$  gegen 0.

Gesamtlaufzeit  $O(n^4 \log n)$  deutlich besser als  $O(n^6)$  bei naivem deterministischen Ansatz.

# Resümee

Es handelt sich um einen sog. **Monte-Carlo-Algorithmus**, da randomisiert und nicht immer korrekt.

Eine andere Art randomisierter Algorithmen sind sog. **Las-Vegas-Algorithmen**, bei denen das Ergebnis immer korrekt ist, aber die Laufzeit eine Zufallsvariable ist (Bsp. RandomizedQuickSort in der ADS).

Aber: Der Alg. ist mit **hoher Wahrscheinlichkeit** korrekt; Fehlerwahrscheinlichkeit konvergiert mit steigendem  $n$  gegen 0.

Gesamtlaufzeit  $O(n^4 \log n)$  deutlich besser als  $O(n^6)$  bei naivem deterministischen Ansatz.

Außerdem ist der randomisierte Algorithmus viel einfacher!

# Resümee

Es handelt sich um einen sog. **Monte-Carlo-Algorithmus**, da randomisiert und nicht immer korrekt.

Eine andere Art randomisierter Algorithmen sind sog. **Las-Vegas-Algorithmen**, bei denen das Ergebnis immer korrekt ist, aber die Laufzeit eine Zufallsvariable ist (Bsp. RandomizedQuickSort in der ADS).

Aber: Der Alg. ist mit **hoher Wahrscheinlichkeit** korrekt; Fehlerwahrscheinlichkeit konvergiert mit steigendem  $n$  gegen 0.

Gesamtlaufzeit  $O(n^4 \log n)$  deutlich besser als  $O(n^6)$  bei naivem deterministischen Ansatz.

Außerdem ist der randomisierte Algorithmus viel einfacher!

**Satz.** Es gibt einen randomisierten Algorithmus für MINCUT mit Laufzeit  $O(n^4 \log n)$  und Erfolgswahrscheinlichkeit  $1 - 1/n^2$ .

# Es geht noch deutlich schneller!

**Satz.** Es gibt einen randomisierten Algorithmus für MINCUT mit  $O(n^2 \log^3 n)$  Laufzeit, der mit hoher Wahrscheinlichkeit einen kleinsten Schnitt ermittelt.

[Karger & Stein, STOC'93]



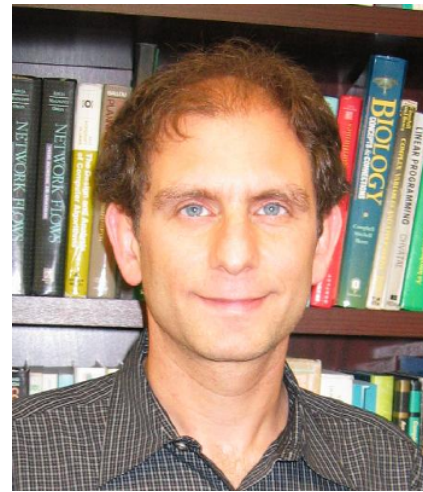
# Es geht noch deutlich schneller!

**Satz.** Es gibt einen randomisierten Algorithmus für MINCUT mit  $O(n^2 \log^3 n)$  Laufzeit, der mit hoher Wahrscheinlichkeit einen kleinsten Schnitt ermittelt.

[Karger & Stein, STOC'93]



David R. Karger



Clifford Stein

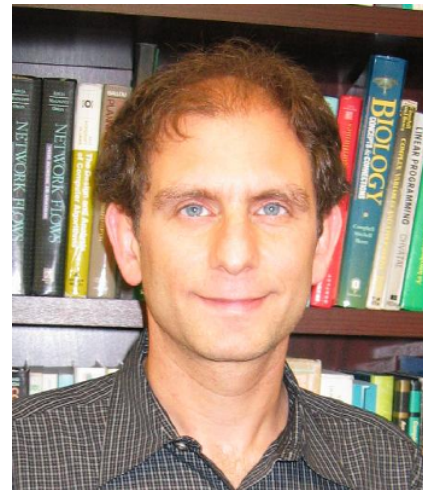
# Es geht noch deutlich schneller!

**Satz.** Es gibt einen randomisierten Algorithmus für MINCUT mit  $O(n^2 \log^3 n)$  Laufzeit, der mit hoher Wahrscheinlichkeit einen kleinsten Schnitt ermittelt.

[Karger & Stein, STOC'93]



David R. Karger



Clifford Stein

Dies ist überraschend schnell, da ein Graph  $\Theta(n^2)$  Kanten haben kann.  
In diesem Fall ist die Laufzeit beinahe linear!

# Partielle Ausführung CONTRACT

Erfolgswahrscheinlichkeit CONTRACT

$$\prod_{i=1}^{n-2} \left( 1 - \frac{2}{n-i+1} \right) = \left( 1 - \frac{2}{n} \right) \cdot \left( 1 - \frac{2}{n-1} \right) \cdot \dots \cdot \frac{1}{3}$$

# Partielle Ausführung CONTRACT

Erfolgswahrscheinlichkeit CONTRACT sinkt mit wachsendem  $i$

$$\prod_{i=1}^{n-2} \left( 1 - \frac{2}{n-i+1} \right) = \left( 1 - \frac{2}{n} \right) \cdot \left( 1 - \frac{2}{n-1} \right) \cdot \dots \cdot \frac{1}{3}$$

# Partielle Ausführung CONTRACT

Erfolgswahrscheinlichkeit CONTRACT sinkt mit wachsendem  $i$

$$\prod_{i=1}^{n-2} \left( 1 - \frac{2}{n-i+1} \right) = \left( 1 - \frac{2}{n} \right) \cdot \left( 1 - \frac{2}{n-1} \right) \cdot \dots \cdot \frac{1}{3}$$

**Idee:** Ändere Strategie mit wachsender Iteration  $i$ .

# Partielle Ausführung CONTRACT

Erfolgswahrscheinlichkeit CONTRACT sinkt mit wachsendem  $i$

$$\prod_{i=1}^{n-2} \left(1 - \frac{2}{n-i+1}\right) = \left(1 - \frac{2}{n}\right) \cdot \left(1 - \frac{2}{n-1}\right) \cdot \dots \cdot \frac{1}{3}$$

**Idee:** Ändere Strategie mit wachsender Iteration  $i$ .

CONTRACT( $G, t$ ): Stoppe, wenn Graph noch  $t$  Knoten enthält.

# Partielle Ausführung CONTRACT

Erfolgswahrscheinlichkeit CONTRACT sinkt mit wachsendem  $i$

$$\prod_{i=1}^{n-2} \left(1 - \frac{2}{n-i+1}\right) = \left(1 - \frac{2}{n}\right) \cdot \left(1 - \frac{2}{n-1}\right) \cdot \dots \cdot \frac{1}{3}$$

**Idee:** Ändere Strategie mit wachsender Iteration  $i$ .

CONTRACT( $G, t$ ): Stoppe, wenn Graph noch  $t$  Knoten enthält.

Schätze Wahrscheinlichkeit ab, dass CONTRACT( $G, t$ ) keine Kante aus optimalem Schnitt  $C$  kontrahiert:

# Partielle Ausführung CONTRACT

Erfolgswahrscheinlichkeit CONTRACT sinkt mit wachsendem  $i$

$$\prod_{i=1}^{n-2} \left( 1 - \frac{2}{n-i+1} \right) = \left( 1 - \frac{2}{n} \right) \cdot \left( 1 - \frac{2}{n-1} \right) \cdot \dots \cdot \frac{1}{3}$$

**Idee:** Ändere Strategie mit wachsender Iteration  $i$ .

CONTRACT( $G, t$ ): Stoppe, wenn Graph noch  $t$  Knoten enthält.

Schätze Wahrscheinlichkeit ab, dass CONTRACT( $G, t$ ) keine Kante aus optimalem Schnitt  $C$  kontrahiert:

$$\Pr \left[ \bigcap_{i=1}^{n-t} \mathcal{E}_i \right] \geq \prod_{i=1}^{n-t} \left( 1 - \frac{2}{n-i+1} \right) = \prod_{i=1}^{n-t} \frac{n-i-1}{n-i+1}$$



# Partielle Ausführung CONTRACT

Erfolgswahrscheinlichkeit CONTRACT sinkt mit wachsendem  $i$

$$\prod_{i=1}^{n-2} \left( 1 - \frac{2}{n-i+1} \right) = \left( 1 - \frac{2}{n} \right) \cdot \left( 1 - \frac{2}{n-1} \right) \cdot \dots \cdot \frac{1}{3}$$

**Idee:** Ändere Strategie mit wachsender Iteration  $i$ .

CONTRACT( $G, t$ ): Stoppe, wenn Graph noch  $t$  Knoten enthält.

Schätze Wahrscheinlichkeit ab, dass CONTRACT( $G, t$ ) keine Kante aus optimalem Schnitt  $C$  kontrahiert:

$$\begin{aligned} \Pr \left[ \bigcap_{i=1}^{n-t} \mathcal{E}_i \right] &\geq \prod_{i=1}^{n-t} \left( 1 - \frac{2}{n-i+1} \right) = \prod_{i=1}^{n-t} \frac{n-i-1}{n-i+1} \\ &= \frac{(n-2) \cdot (n-3) \cdot \dots \cdot t \cdot (t-1)}{n \cdot (n-1) \cdot (n-2) \cdot \dots \cdot (t+1)} = \end{aligned}$$

# Partielle Ausführung CONTRACT

Erfolgswahrscheinlichkeit CONTRACT sinkt mit wachsendem  $i$

$$\prod_{i=1}^{n-2} \left( 1 - \frac{2}{n-i+1} \right) = \left( 1 - \frac{2}{n} \right) \cdot \left( 1 - \frac{2}{n-1} \right) \cdot \dots \cdot \frac{1}{3}$$

**Idee:** Ändere Strategie mit wachsender Iteration  $i$ .

CONTRACT( $G, t$ ): Stoppe, wenn Graph noch  $t$  Knoten enthält.

Schätze Wahrscheinlichkeit ab, dass CONTRACT( $G, t$ ) keine Kante aus optimalem Schnitt  $C$  kontrahiert:

$$\begin{aligned} \Pr \left[ \bigcap_{i=1}^{n-t} \mathcal{E}_i \right] &\geq \prod_{i=1}^{n-t} \left( 1 - \frac{2}{n-i+1} \right) = \prod_{i=1}^{n-t} \frac{n-i-1}{n-i+1} \\ &= \frac{\cancel{(n-2)} \cdot \cancel{(n-3)} \cdot \dots \cdot t \cdot (t-1)}{n \cdot (n-1) \cdot \cancel{(n-2)} \cdot \dots \cdot \cancel{(t+1)}} = \end{aligned}$$

# Partielle Ausführung CONTRACT

Erfolgswahrscheinlichkeit CONTRACT sinkt mit wachsendem  $i$

$$\prod_{i=1}^{n-2} \left( 1 - \frac{2}{n-i+1} \right) = \left( 1 - \frac{2}{n} \right) \cdot \left( 1 - \frac{2}{n-1} \right) \cdot \dots \cdot \frac{1}{3}$$

**Idee:** Ändere Strategie mit wachsender Iteration  $i$ .

CONTRACT( $G, t$ ): Stoppe, wenn Graph noch  $t$  Knoten enthält.

Schätze Wahrscheinlichkeit ab, dass CONTRACT( $G, t$ ) keine Kante aus optimalem Schnitt  $C$  kontrahiert:

$$\begin{aligned} \Pr \left[ \bigcap_{i=1}^{n-t} \mathcal{E}_i \right] &\geq \prod_{i=1}^{n-t} \left( 1 - \frac{2}{n-i+1} \right) = \prod_{i=1}^{n-t} \frac{n-i-1}{n-i+1} \\ &= \frac{\cancel{(n-2)} \cdot \cancel{(n-3)} \cdot \dots \cdot t \cdot (t-1)}{n \cdot (n-1) \cdot \cancel{(n-2)} \cdot \dots \cdot \cancel{(t+1)}} = \frac{t(t-1)}{n(n-1)} \end{aligned}$$

# Partielle Ausführung CONTRACT

Erfolgswahrscheinlichkeit CONTRACT sinkt mit wachsendem  $i$

$$\prod_{i=1}^{n-2} \left( 1 - \frac{2}{n-i+1} \right) = \left( 1 - \frac{2}{n} \right) \cdot \left( 1 - \frac{2}{n-1} \right) \cdot \dots \cdot \frac{1}{3}$$

**Idee:** Ändere Strategie mit wachsender Iteration  $i$ .

CONTRACT( $G, t$ ): Stoppe, wenn Graph noch  $t$  Knoten enthält.

Schätze Wahrscheinlichkeit ab, dass CONTRACT( $G, t$ ) keine Kante aus optimalem Schnitt  $C$  kontrahiert:

$$\begin{aligned} \Pr \left[ \bigcap_{i=1}^{n-t} \mathcal{E}_i \right] &\geq \prod_{i=1}^{n-t} \left( 1 - \frac{2}{n-i+1} \right) = \prod_{i=1}^{n-t} \frac{n-i-1}{n-i+1} \quad (\star) \\ &= \frac{\cancel{(n-2)} \cdot \cancel{(n-3)} \cdot \dots \cdot t \cdot (t-1)}{n \cdot (n-1) \cdot \cancel{(n-2)} \cdot \dots \cdot \cancel{(t+1)}} = \frac{t(t-1)}{n(n-1)} \end{aligned}$$

# Algorithmus FASTCUT

FASTCUT(zsghd. Multigraph  $G = (V, E)$ )

$n \leftarrow |V|$

**if**  $n \leq 6$  **then**

| löse Problem „brute force“

# Algorithmus FASTCUT

FASTCUT(zsghd. Multigraph  $G = (V, E)$ )

$n \leftarrow |V|$

**if**  $n \leq 6$  **then**

| löse Problem „brute force“

**else**

|  $t \leftarrow \lceil 1 + n/\sqrt{2} \rceil \approx 0,7 \cdot n$

|  $G_1 \leftarrow \text{CONTRACT}(G, t)$

# Algorithmus FASTCUT

FASTCUT(zsghd. Multigraph  $G = (V, E)$ )

$n \leftarrow |V|$

**if**  $n \leq 6$  **then**

| löse Problem „brute force“

**else**

|  $t \leftarrow \lceil 1 + n/\sqrt{2} \rceil \approx 0,7 \cdot n$

|  $G_1 \leftarrow \text{CONTRACT}(G, t)$

|  $G_2 \leftarrow \text{CONTRACT}(G, t)$

# Algorithmus FASTCUT

FASTCUT(zsghd. Multigraph  $G = (V, E)$ )

$n \leftarrow |V|$

**if**  $n \leq 6$  **then**

| löse Problem „brute force“

**else**

|  $t \leftarrow \lceil 1 + n/\sqrt{2} \rceil \approx 0,7 \cdot n$

|  $G_1 \leftarrow \text{CONTRACT}(G, t)$

|  $G_2 \leftarrow \text{CONTRACT}(G, t)$

← unabhängige Ausführungen



# Algorithmus FASTCUT

FASTCUT(zsghd. Multigraph  $G = (V, E)$ )

$n \leftarrow |V|$

**if**  $n \leq 6$  **then**

| löse Problem „brute force“

**else**

|  $t \leftarrow \lceil 1 + n/\sqrt{2} \rceil \approx 0,7 \cdot n$

|  $G_1 \leftarrow \text{CONTRACT}(G, t)$

|  $G_2 \leftarrow \text{CONTRACT}(G, t)$

|  $(S_1, T_1) \leftarrow \text{FASTCUT}(G_1)$

|  $(S_2, T_2) \leftarrow \text{FASTCUT}(G_2)$

← unabhängige Ausführungen

# Algorithmus FASTCUT

FASTCUT(zsghd. Multigraph  $G = (V, E)$ )

$n \leftarrow |V|$

**if**  $n \leq 6$  **then**

    | löse Problem „brute force“

**else**

$t \leftarrow \lceil 1 + n/\sqrt{2} \rceil \approx 0,7 \cdot n$

$G_1 \leftarrow \text{CONTRACT}(G, t)$

$G_2 \leftarrow \text{CONTRACT}(G, t)$

$(S_1, T_1) \leftarrow \text{FASTCUT}(G_1)$

$(S_2, T_2) \leftarrow \text{FASTCUT}(G_2)$

**return** den kleineren der Schnitte  $(S_1, T_1), (S_2, T_2)$

← unabhängige Ausführungen

# Algorithmus FASTCUT

FASTCUT(zsghd. Multigraph  $G = (V, E)$ )

$n \leftarrow |V|$

**if**  $n \leq 6$  **then**

    | löse Problem „brute force“

**else**

$t \leftarrow \lceil 1 + n/\sqrt{2} \rceil \approx 0,7 \cdot n$

$G_1 \leftarrow \text{CONTRACT}(G, t)$

$G_2 \leftarrow \text{CONTRACT}(G, t)$

$(S_1, T_1) \leftarrow \text{FASTCUT}(G_1)$

$(S_2, T_2) \leftarrow \text{FASTCUT}(G_2)$

**return** den kleineren der Schnitte  $(S_1, T_1), (S_2, T_2)$

← unabhängige Ausführungen

*Laufzeit?*

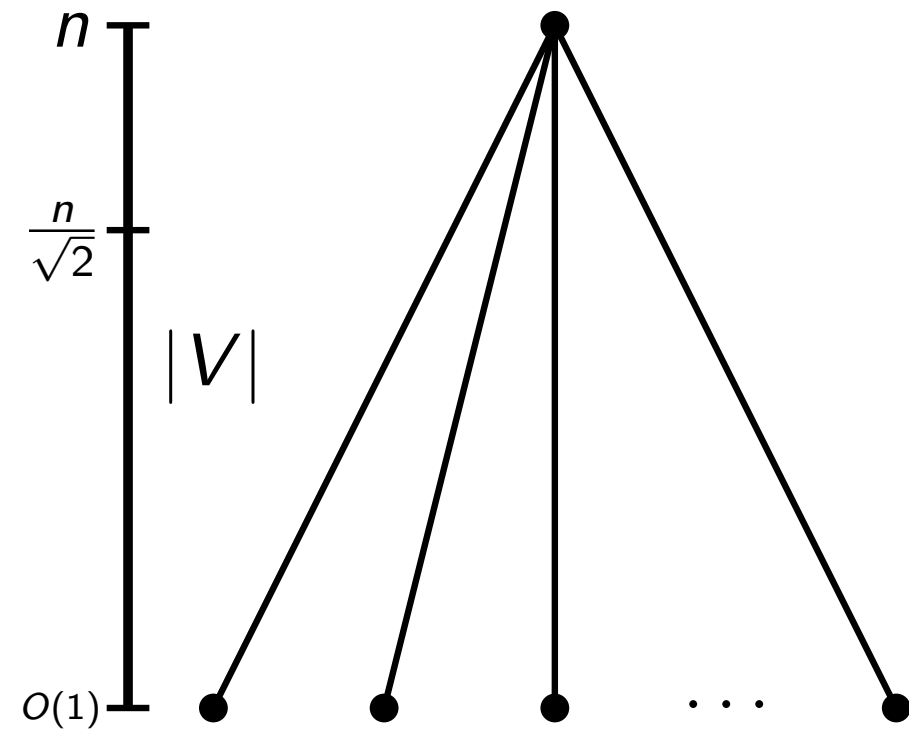
# Intuition FASTCUT vs. CONTRACT

$$\prod_{i=1}^{n-2} \left( 1 - \frac{2}{n-i+1} \right) \leftarrow \text{Erfolgswahrscheinlichkeit}$$

# Intuition FASTCUT vs. CONTRACT

$$\prod_{i=1}^{n-2} \left(1 - \frac{2}{n-i+1}\right) \leftarrow \text{Erfolgswahrscheinlichkeit}$$

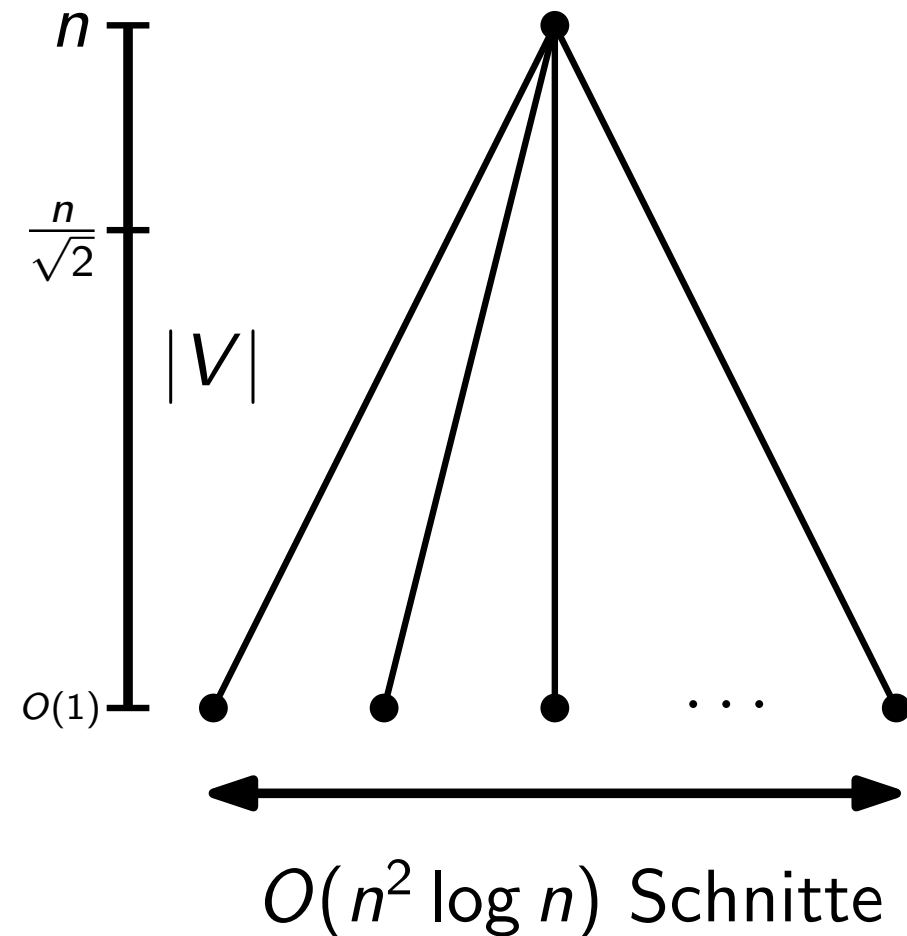
CONTRACT (mit Wh.)



# Intuition FASTCUT vs. CONTRACT

$$\prod_{i=1}^{n-2} \left(1 - \frac{2}{n-i+1}\right) \leftarrow \text{Erfolgswahrscheinlichkeit}$$

CONTRACT (mit Wh.)

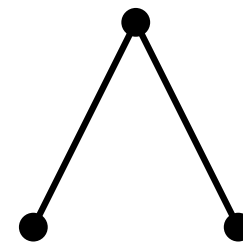
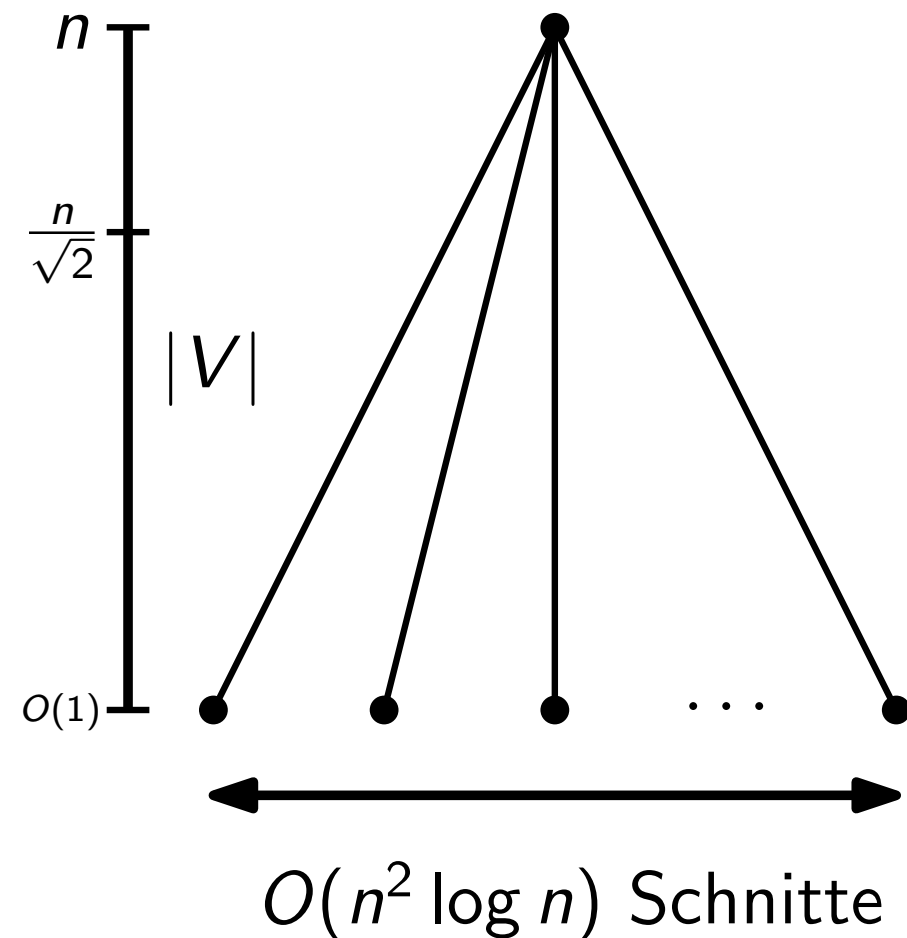


# Intuition FASTCUT vs. CONTRACT

$$\prod_{i=1}^{n-2} \left(1 - \frac{2}{n-i+1}\right) \leftarrow \text{Erfolgswahrscheinlichkeit}$$

CONTRACT (mit Wh.)

FASTCUT (einmal)

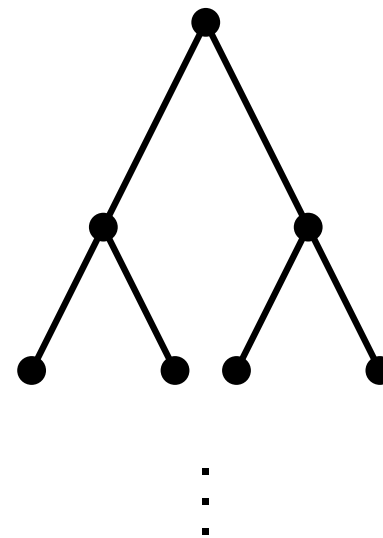
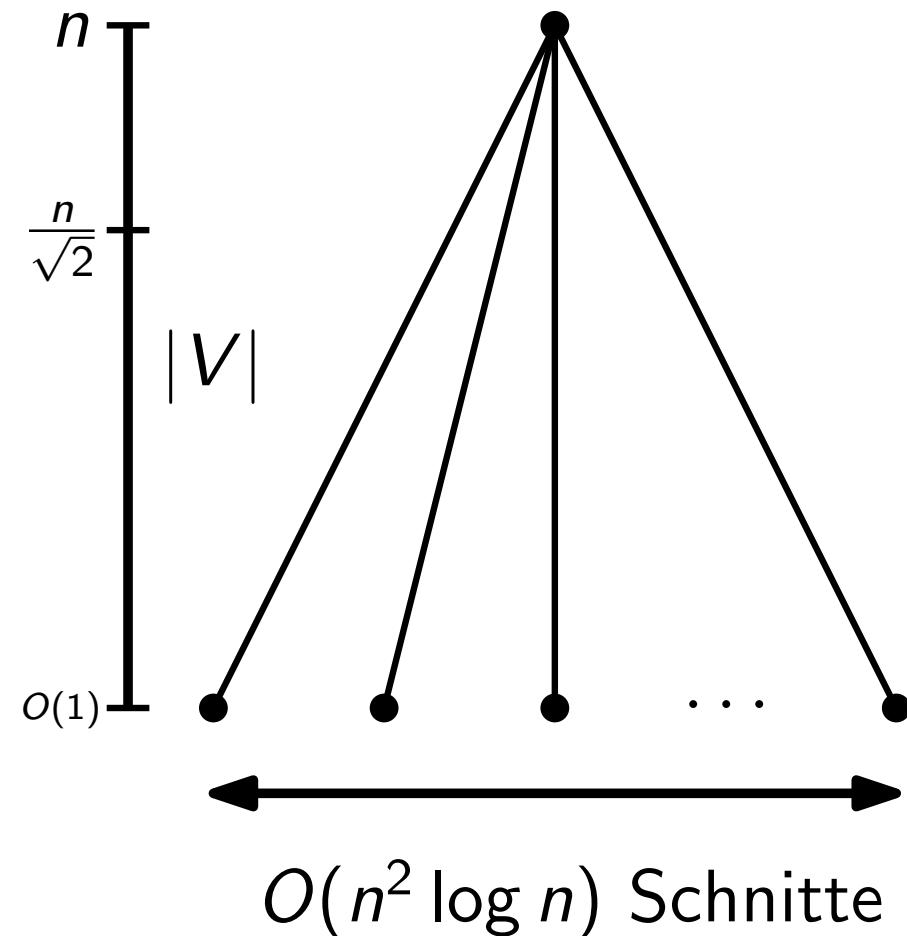


# Intuition FASTCUT vs. CONTRACT

$$\prod_{i=1}^{n-2} \left(1 - \frac{2}{n-i+1}\right) \longleftarrow \text{Erfolgswahrscheinlichkeit}$$

CONTRACT (mit Wh.)

FASTCUT (einmal)



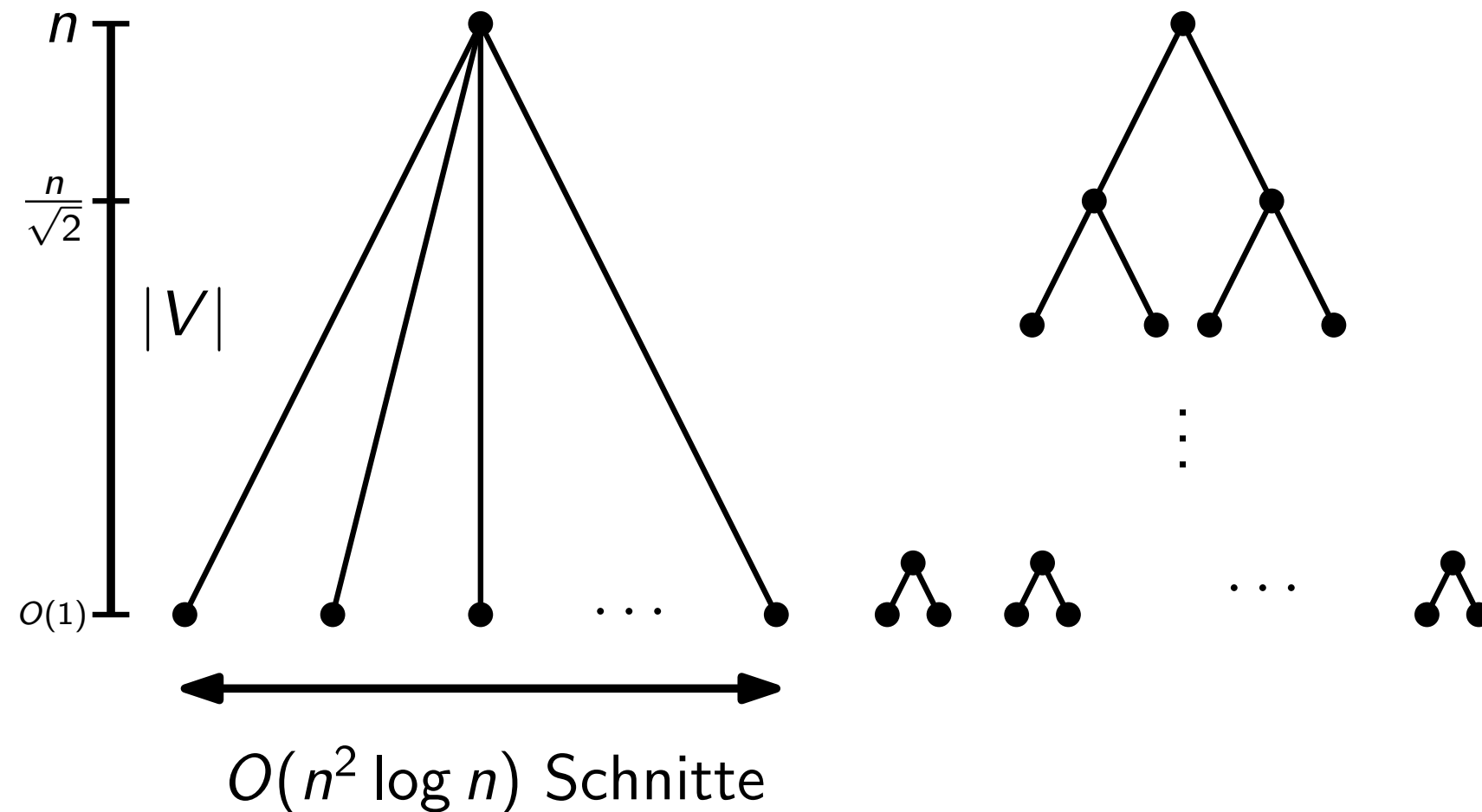


# Intuition FASTCUT vs. CONTRACT

$$\prod_{i=1}^{n-2} \left(1 - \frac{2}{n-i+1}\right) \leftarrow \text{Erfolgswahrscheinlichkeit}$$

CONTRACT (mit Wh.)

FASTCUT (einmal)

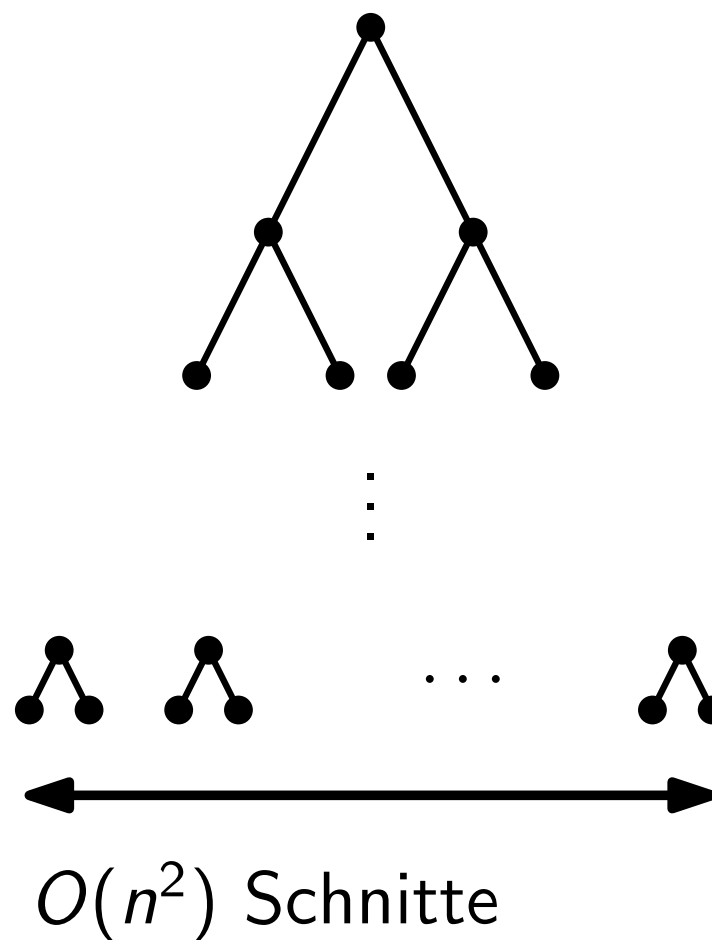
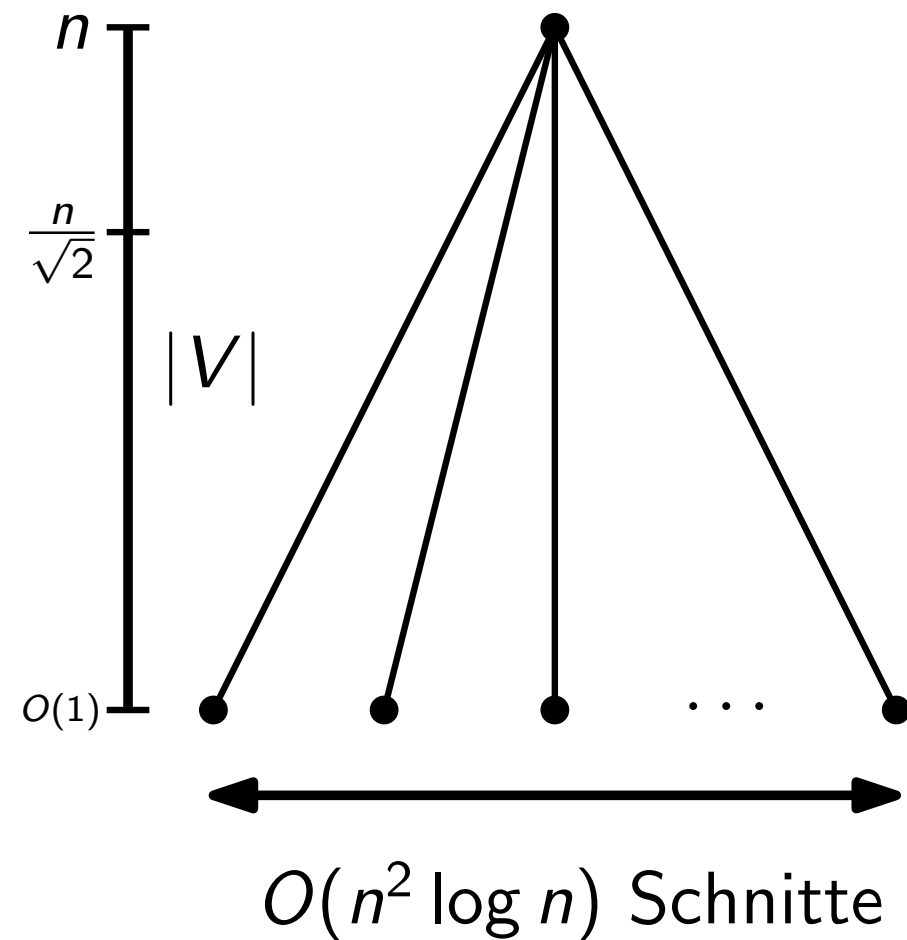


# Intuition FASTCUT vs. CONTRACT

$$\prod_{i=1}^{n-2} \left(1 - \frac{2}{n-i+1}\right) \leftarrow \text{Erfolgswahrscheinlichkeit}$$

CONTRACT (mit Wh.)

FASTCUT (einmal)

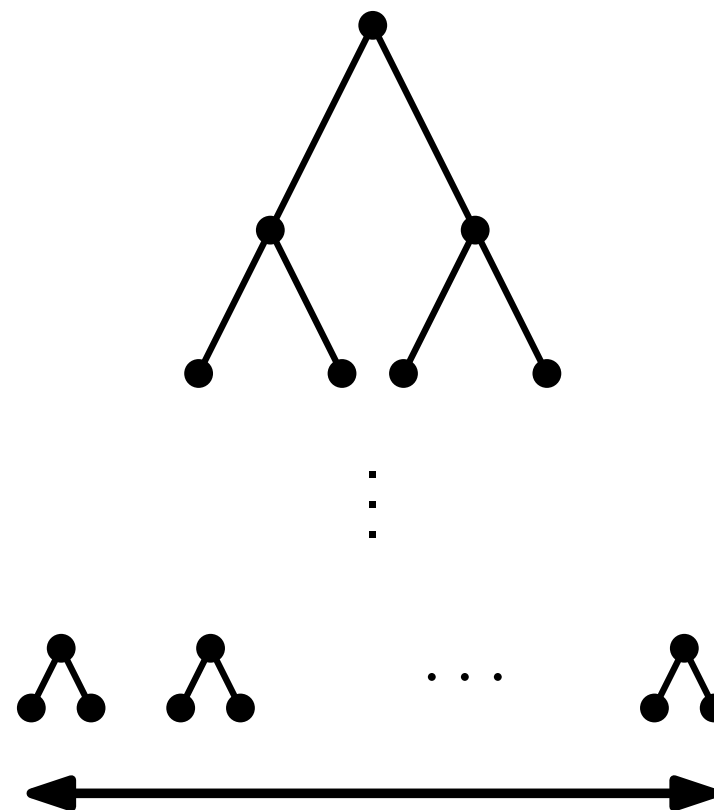
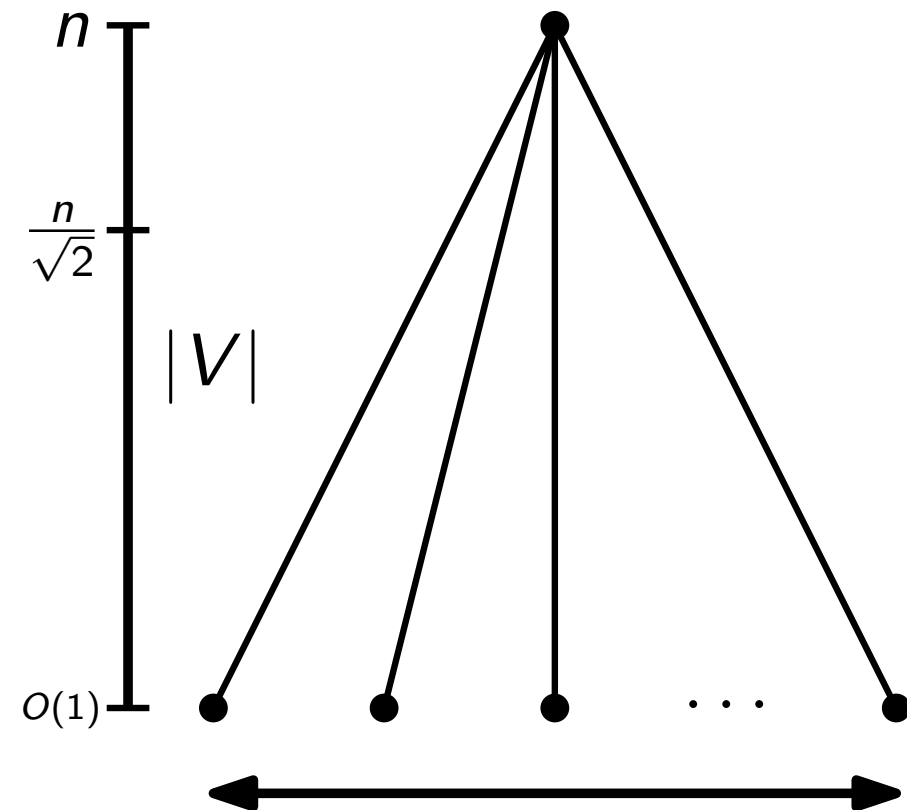


# Intuition FASTCUT vs. CONTRACT

$$\prod_{i=1}^{n-2} \left(1 - \frac{2}{n-i+1}\right) \longleftarrow \text{Erfolgswahrscheinlichkeit}$$

CONTRACT (mit Wh.)

FASTCUT (einmal)



$O(n^2 \log n)$  Schnitte

$O(n^2)$  Schnitte, da  $2^{\log_{\sqrt{2}} n} = n^2$ .

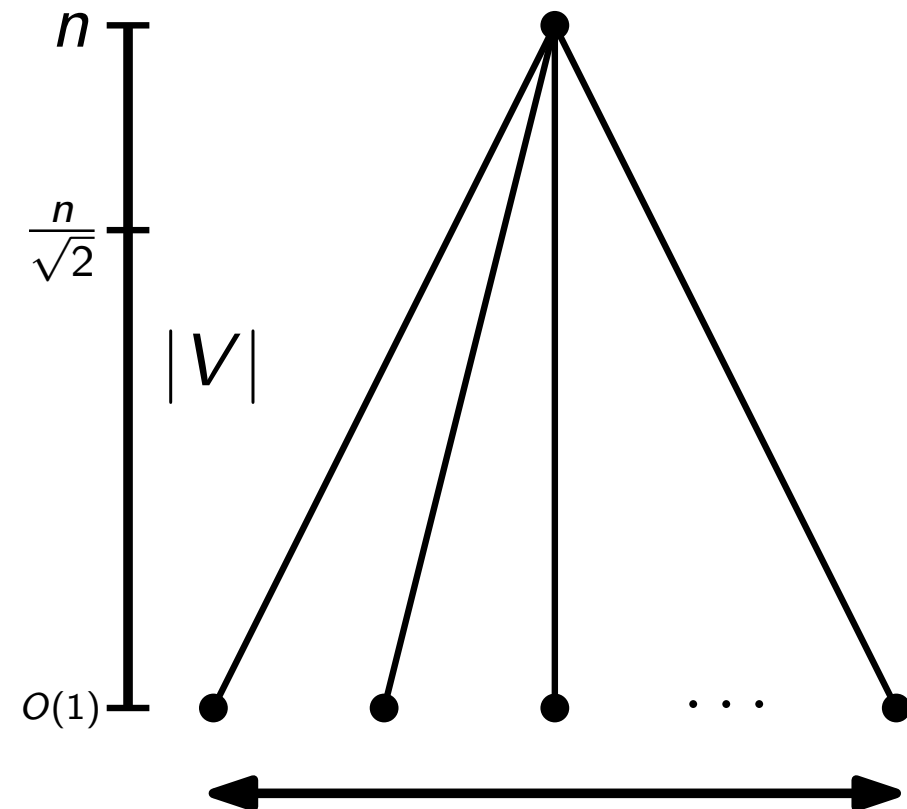
# Intuition FASTCUT vs. CONTRACT

$$\prod_{i=1}^{n-2} \left(1 - \frac{2}{n-i+1}\right) \leftarrow \text{Erfolgswahrscheinlichkeit}$$

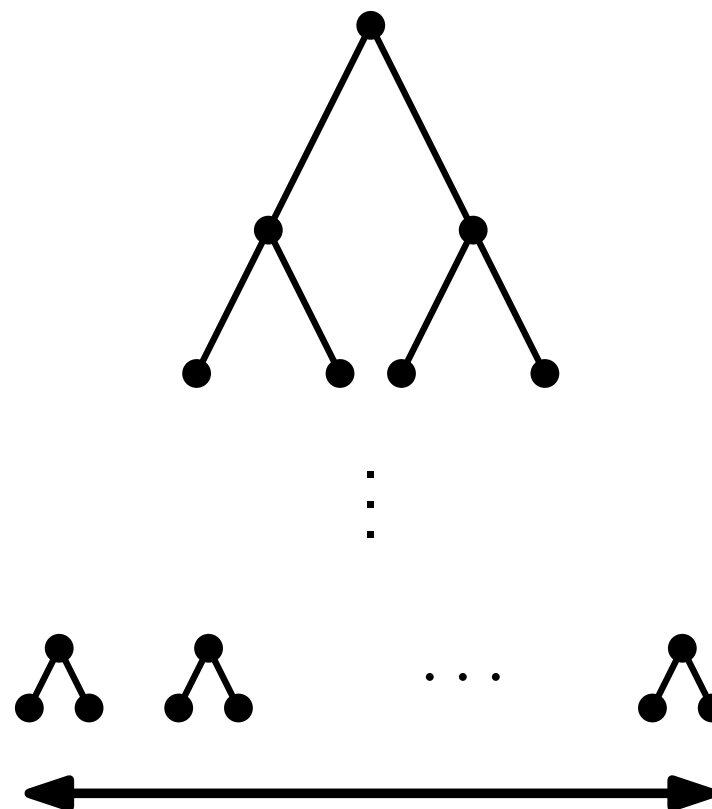
CONTRACT (mit Wh.)

FASTCUT (einmal)

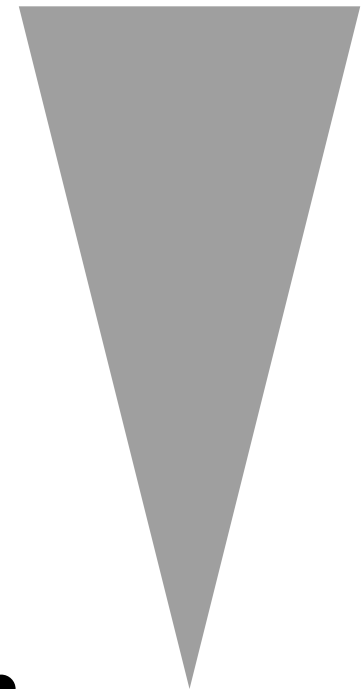
Ersparnis



$O(n^2 \log n)$  Schnitte



$O(n^2)$  Schnitte, da  $2^{\log_{\sqrt{2}} n} = n^2$ .



# Laufzeit

**Satz.** FASTCUT läuft in  $O(n^2 \log n)$  Zeit.

```

FASTCUT(zsghd. Multigraph  $G = (V, E)$ )
   $n \leftarrow |V|$ 
  if  $n \leq 6$  then
    | löse Problem „brute force“
  else
     $t \leftarrow \lceil 1 + n/\sqrt{2} \rceil \approx 0,7 \cdot n$ 
     $G_1 \leftarrow \text{CONTRACT}(G, t)$ 
     $G_2 \leftarrow \text{CONTRACT}(G, t)$ 
     $(S_1, T_1) \leftarrow \text{FASTCUT}(G_1)$ 
     $(S_2, T_2) \leftarrow \text{FASTCUT}(G_2)$ 
    return kleineren von  $(S_1, T_1), (S_2, T_2)$ 

```

# Laufzeit

**Satz.** FASTCUT läuft in  $O(n^2 \log n)$  Zeit.

*Beweis.* Rekurrenz

$$T(n) =$$

```

FASTCUT(zsghd. Multigraph  $G = (V, E)$ )
   $n \leftarrow |V|$ 
  if  $n \leq 6$  then
    | löse Problem „brute force“
  else
    |  $t \leftarrow \lceil 1 + n/\sqrt{2} \rceil \approx 0,7 \cdot n$ 
    |  $G_1 \leftarrow \text{CONTRACT}(G, t)$ 
    |  $G_2 \leftarrow \text{CONTRACT}(G, t)$ 
    |  $(S_1, T_1) \leftarrow \text{FASTCUT}(G_1)$ 
    |  $(S_2, T_2) \leftarrow \text{FASTCUT}(G_2)$ 
    | return kleineren von  $(S_1, T_1), (S_2, T_2)$ 
  
```

# Laufzeit

**Satz.** FASTCUT läuft in  $O(n^2 \log n)$  Zeit.

*Beweis.* Rekurrenz

$$T(n) = 2T\left(\frac{n}{\sqrt{2}}\right) + O(n^2)$$

```

FASTCUT(zsghd. Multigraph  $G = (V, E)$ )
   $n \leftarrow |V|$ 
  if  $n \leq 6$  then
    löse Problem „brute force“
  else
     $t \leftarrow \lceil 1 + n/\sqrt{2} \rceil \approx 0,7 \cdot n$ 
     $G_1 \leftarrow \text{CONTRACT}(G, t)$ 
     $G_2 \leftarrow \text{CONTRACT}(G, t)$ 
     $(S_1, T_1) \leftarrow \text{FASTCUT}(G_1)$ 
     $(S_2, T_2) \leftarrow \text{FASTCUT}(G_2)$ 
    return kleineren von  $(S_1, T_1), (S_2, T_2)$ 

```

# Laufzeit

**Satz.** FASTCUT läuft in  $O(n^2 \log n)$  Zeit.

*Beweis.* Rekurrenz

$$T(n) = 2T\left(\frac{n}{\sqrt{2}}\right) + O(n^2)$$

Erinnerung Mastertheorem (2. Fall):

$$T(n) = aT(n/b) + f(n) \text{ mit } f(n) = \Theta(n^{\log_b a})$$

$$\Rightarrow T(n) = \Theta(\quad).$$

```

FASTCUT(zsghd. Multigraph  $G = (V, E)$ )
   $n \leftarrow |V|$ 
  if  $n \leq 6$  then
    löse Problem „brute force“
  else
     $t \leftarrow \lceil 1 + n/\sqrt{2} \rceil \approx 0,7 \cdot n$ 
     $G_1 \leftarrow \text{CONTRACT}(G, t)$ 
     $G_2 \leftarrow \text{CONTRACT}(G, t)$ 
     $(S_1, T_1) \leftarrow \text{FASTCUT}(G_1)$ 
     $(S_2, T_2) \leftarrow \text{FASTCUT}(G_2)$ 
    return kleineren von  $(S_1, T_1), (S_2, T_2)$ 

```



# Laufzeit

**Satz.** FASTCUT läuft in  $O(n^2 \log n)$  Zeit.

*Beweis.* Rekurrenz

$$T(n) = 2T\left(\frac{n}{\sqrt{2}}\right) + O(n^2)$$

Erinnerung Mastertheorem (2. Fall):

$$T(n) = aT(n/b) + f(n) \text{ mit } f(n) = \Theta(n^{\log_b a})$$

$$\Rightarrow T(n) = \Theta(f(n) \cdot \log n).$$

```

FASTCUT(zsghd. Multigraph  $G = (V, E)$ )
   $n \leftarrow |V|$ 
  if  $n \leq 6$  then
    löse Problem „brute force“
  else
     $t \leftarrow \lceil 1 + n/\sqrt{2} \rceil \approx 0,7 \cdot n$ 
     $G_1 \leftarrow \text{CONTRACT}(G, t)$ 
     $G_2 \leftarrow \text{CONTRACT}(G, t)$ 
     $(S_1, T_1) \leftarrow \text{FASTCUT}(G_1)$ 
     $(S_2, T_2) \leftarrow \text{FASTCUT}(G_2)$ 
    return kleineren von  $(S_1, T_1), (S_2, T_2)$ 

```

# Laufzeit

**Satz.** FASTCUT läuft in  $O(n^2 \log n)$  Zeit.

*Beweis.* Rekurrenz

$$T(n) = 2T\left(\frac{n}{\sqrt{2}}\right) + O(n^2)$$

Erinnerung Mastertheorem (2. Fall):

$$T(n) = aT(n/b) + f(n) \text{ mit } f(n) = \Theta(n^{\log_b a})$$

$$\Rightarrow T(n) = \Theta(f(n) \cdot \log n).$$

In unserem Fall:  $a = 2$ ,  $b = \sqrt{2}$ ,  $f(n) = O(n^2)$ .

```

FASTCUT(zsghd. Multigraph  $G = (V, E)$ )
   $n \leftarrow |V|$ 
  if  $n \leq 6$  then
    löse Problem „brute force“
  else
     $t \leftarrow \lceil 1 + n/\sqrt{2} \rceil \approx 0,7 \cdot n$ 
     $G_1 \leftarrow \text{CONTRACT}(G, t)$ 
     $G_2 \leftarrow \text{CONTRACT}(G, t)$ 
     $(S_1, T_1) \leftarrow \text{FASTCUT}(G_1)$ 
     $(S_2, T_2) \leftarrow \text{FASTCUT}(G_2)$ 
    return kleineren von  $(S_1, T_1)$ ,  $(S_2, T_2)$ 
  
```

# Laufzeit

**Satz.** FASTCUT läuft in  $O(n^2 \log n)$  Zeit.

*Beweis.* Rekurrenz

$$T(n) = 2T\left(\frac{n}{\sqrt{2}}\right) + O(n^2)$$

Erinnerung Mastertheorem (2. Fall):

$$T(n) = aT(n/b) + f(n) \text{ mit } f(n) = \Theta(n^{\log_b a})$$

$$\Rightarrow T(n) = \Theta(f(n) \cdot \log n).$$

In unserem Fall:  $a = 2$ ,  $b = \sqrt{2}$ ,  $f(n) = O(n^2)$ .

Wegen  $\log_{\sqrt{2}} 2 = 2$  folgt die Behauptung. □

```

FASTCUT(zsghd. Multigraph  $G = (V, E)$ )
   $n \leftarrow |V|$ 
  if  $n \leq 6$  then
    löse Problem „brute force“
  else
     $t \leftarrow \lceil 1 + n/\sqrt{2} \rceil \approx 0,7 \cdot n$ 
     $G_1 \leftarrow \text{CONTRACT}(G, t)$ 
     $G_2 \leftarrow \text{CONTRACT}(G, t)$ 
     $(S_1, T_1) \leftarrow \text{FASTCUT}(G_1)$ 
     $(S_2, T_2) \leftarrow \text{FASTCUT}(G_2)$ 
    return kleineren von  $(S_1, T_1)$ ,  $(S_2, T_2)$ 
  
```

# Erfolgswahrscheinlichkeit (I)

**Satz.** FASTCUT liefert mit Wahrscheinlichkeit  $\Omega(1/\log n)$  einen kleinsten Schnitt.

*Beweis.*

# Erfolgswahrscheinlichkeit (I)

**Satz.** FASTCUT liefert mit Wahrscheinlichkeit  $\Omega(1/\log n)$  einen kleinsten Schnitt.

*Beweis.*

Betrachte Graph  $G$  mit  $n$  Knoten bei rekursivem Aufruf.

# Erfolgswahrscheinlichkeit (I)

**Satz.** FASTCUT liefert mit Wahrscheinlichkeit  $\Omega(1/\log n)$  einen kleinsten Schnitt.

*Beweis.*

Betrachte Graph  $G$  mit  $n$  Knoten bei rekursivem Aufruf.

Seien  $G_1, G_2$  die Graphen nach Anw. von CONTRACT( $H, t$ ).

# Erfolgswahrscheinlichkeit (I)

**Satz.** FASTCUT liefert mit Wahrscheinlichkeit  $\Omega(1/\log n)$  einen kleinsten Schnitt.

*Beweis.*

Betrachte Graph  $G$  mit  $n$  Knoten bei rekursivem Aufruf.

Seien  $G_1, G_2$  die Graphen nach Anw. von CONTRACT( $H, t$ ).

Wahrscheinlichkeit, dass kleinster Schnitt aus  $G$  in  $G_1$  „überlebt“ ist nach  $(\star)$  mindestens

$$\frac{t(t-1)}{n(n-1)} =$$

# Erfolgswahrscheinlichkeit (I)

**Satz.** FASTCUT liefert mit Wahrscheinlichkeit  $\Omega(1/\log n)$  einen kleinsten Schnitt.

*Beweis.*

Betrachte Graph  $G$  mit  $n$  Knoten bei rekursivem Aufruf.

Seien  $G_1, G_2$  die Graphen nach Anw. von CONTRACT( $H, t$ ).

Wahrscheinlichkeit, dass kleinster Schnitt aus  $G$  in  $G_1$  „überlebt“ ist nach ( $\star$ ) mindestens

$$\frac{t(t-1)}{n(n-1)} = \frac{\lceil 1 + n/\sqrt{2} \rceil (\lceil 1 + n/\sqrt{2} \rceil - 1)}{n(n-1)} \geq$$



# Erfolgswahrscheinlichkeit (I)

**Satz.** FASTCUT liefert mit Wahrscheinlichkeit  $\Omega(1/\log n)$  einen kleinsten Schnitt.

*Beweis.*

Betrachte Graph  $G$  mit  $n$  Knoten bei rekursivem Aufruf.

Seien  $G_1, G_2$  die Graphen nach Anw. von CONTRACT( $H, t$ ).

Wahrscheinlichkeit, dass kleinster Schnitt aus  $G$  in  $G_1$  „überlebt“ ist nach  $(\star)$  mindestens

$$\frac{t(t-1)}{n(n-1)} = \frac{\lceil 1 + n/\sqrt{2} \rceil (\lceil 1 + n/\sqrt{2} \rceil - 1)}{n(n-1)} \geq \frac{(n/\sqrt{2})^2}{n^2} =$$

# Erfolgswahrscheinlichkeit (I)

**Satz.** FASTCUT liefert mit Wahrscheinlichkeit  $\Omega(1/\log n)$  einen kleinsten Schnitt.

*Beweis.*

Betrachte Graph  $G$  mit  $n$  Knoten bei rekursivem Aufruf.

Seien  $G_1, G_2$  die Graphen nach Anw. von CONTRACT( $H, t$ ).

Wahrscheinlichkeit, dass kleinster Schnitt aus  $G$  in  $G_1$  „überlebt“ ist nach  $(\star)$  mindestens

$$\frac{t(t-1)}{n(n-1)} = \frac{\lceil 1 + n/\sqrt{2} \rceil (\lceil 1 + n/\sqrt{2} \rceil - 1)}{n(n-1)} \geq \frac{(n/\sqrt{2})^2}{n^2} = \frac{1}{2}.$$

# Erfolgswahrscheinlichkeit (II)

Seien  $G_1, G_2$  die Graphen nach  $2 \times \text{CONTRACT}(G, t)$ .

WK, dass kleinster Schnitt in  $G_1$  überlebt, ist  $\geq 1/2$ .

# Erfolgswahrscheinlichkeit (II)

Seien  $G_1, G_2$  die Graphen nach  $2 \times \text{CONTRACT}(G, t)$ .

WK, dass kleinster Schnitt in  $G_1$  überlebt, ist  $\geq 1/2$ .

`FASTCUT` findet kleinsten Schnitt in  $G$ , wenn ein solcher in  $G_1$  oder  $G_2$  überlebt *und* der rekursive Aufruf im entsprechenden Graphen  $G_i$  ebenfalls einen kleinsten Schnitt findet.

# Erfolgswahrscheinlichkeit (II)

Seien  $G_1, G_2$  die Graphen nach  $2 \times \text{CONTRACT}(G, t)$ .

WK, dass kleinster Schnitt in  $G_1$  überlebt, ist  $\geq 1/2$ .

`FASTCUT` findet kleinsten Schnitt in  $G$ , wenn ein solcher in  $G_1$  oder  $G_2$  überlebt *und* der rekursive Aufruf im entsprechenden Graphen  $G_i$  ebenfalls einen kleinsten Schnitt findet.

Sei  $\tau = \Theta(\log n)$  die Tiefe der Rekursion von `FASTCUT`( $G$ ).

# Erfolgswahrscheinlichkeit (II)

Seien  $G_1, G_2$  die Graphen nach  $2 \times \text{CONTRACT}(G, t)$ .

WK, dass kleinster Schnitt in  $G_1$  überlebt, ist  $\geq 1/2$ .

`FASTCUT` findet kleinsten Schnitt in  $G$ , wenn ein solcher in  $G_1$  oder  $G_2$  überlebt *und* der rekursive Aufruf im entsprechenden Graphen  $G_i$  ebenfalls einen kleinsten Schnitt findet.

Sei  $\tau = \Theta(\log n)$  die Tiefe der Rekursion von `FASTCUT`( $G$ ).

$\Rightarrow$  untere Schranke  $p(\tau)$  für die Erfolgs-WK auf Tiefe  $\tau$ :

$$p(\tau + 1) =$$

$$\text{wobei } p(0) = 1.$$

# Erfolgswahrscheinlichkeit (II)

Seien  $G_1, G_2$  die Graphen nach  $2 \times \text{CONTRACT}(G, t)$ .

WK, dass kleinster Schnitt in  $G_1$  überlebt, ist  $\geq 1/2$ .

FASTCUT findet kleinsten Schnitt in  $G$ , wenn ein solcher in  $G_1$  oder  $G_2$  überlebt *und* der rekursive Aufruf im entsprechenden Graphen  $G_i$  ebenfalls einen kleinsten Schnitt findet.

Sei  $\tau = \Theta(\log n)$  die Tiefe der Rekursion von FASTCUT( $G$ ).

$\Rightarrow$  untere Schranke  $p(\tau)$  für die Erfolgs-WK auf Tiefe  $\tau$ :

$$p(\tau + 1) = 1 - \underbrace{\left( \quad \right)^2}_{\text{Fehler-WK auf Tiefe } \tau + 1} = \quad \text{wobei } p(0) = 1.$$

# Erfolgswahrscheinlichkeit (II)

Seien  $G_1, G_2$  die Graphen nach  $2 \times \text{CONTRACT}(G, t)$ .

WK, dass kleinster Schnitt in  $G_1$  überlebt, ist  $\geq 1/2$ .

FASTCUT findet kleinsten Schnitt in  $G$ , wenn ein solcher in  $G_1$  oder  $G_2$  überlebt *und* der rekursive Aufruf im entsprechenden Graphen  $G_i$  ebenfalls einen kleinsten Schnitt findet.

Sei  $\tau = \Theta(\log n)$  die Tiefe der Rekursion von FASTCUT( $G$ ).

$\Rightarrow$  untere Schranke  $p(\tau)$  für die Erfolgs-WK auf Tiefe  $\tau$ :

$$p(\tau + 1) = 1 - \underbrace{\left( \underbrace{\text{Fehler-WK}_{\text{FASTCUT}(G_1)}} \right)^2}_{\text{Fehler-WK auf Tiefe } \tau + 1} = \quad \text{wobei } p(0) = 1.$$



# Erfolgswahrscheinlichkeit (II)

Seien  $G_1, G_2$  die Graphen nach  $2 \times \text{CONTRACT}(G, t)$ .

WK, dass kleinster Schnitt in  $G_1$  überlebt, ist  $\geq 1/2$ .

FASTCUT findet kleinsten Schnitt in  $G$ , wenn ein solcher in  $G_1$  oder  $G_2$  überlebt *und* der rekursive Aufruf im entsprechenden Graphen  $G_i$  ebenfalls einen kleinsten Schnitt findet.

Sei  $\tau = \Theta(\log n)$  die Tiefe der Rekursion von FASTCUT( $G$ ).

$\Rightarrow$  untere Schranke  $p(\tau)$  für die Erfolgs-WK auf Tiefe  $\tau$ :

$$p(\tau + 1) = 1 - \underbrace{\left(1 - \frac{1}{2}p(\tau)\right)^2}_{\text{Fehler-WK auf Tiefe } \tau + 1} = \underbrace{\text{Fehler-WK FASTCUT}(G_1)}_{\text{Fehler-WK auf Tiefe } \tau + 1} \quad \text{wobei } p(0) = 1.$$

# Erfolgswahrscheinlichkeit (II)

Seien  $G_1, G_2$  die Graphen nach  $2 \times \text{CONTRACT}(G, t)$ .

WK, dass kleinster Schnitt in  $G_1$  überlebt, ist  $\geq 1/2$ .

FASTCUT findet kleinsten Schnitt in  $G$ , wenn ein solcher in  $G_1$  oder  $G_2$  überlebt *und* der rekursive Aufruf im entsprechenden Graphen  $G_i$  ebenfalls einen kleinsten Schnitt findet.

Sei  $\tau = \Theta(\log n)$  die Tiefe der Rekursion von FASTCUT( $G$ ).

$\Rightarrow$  untere Schranke  $p(\tau)$  für die Erfolgs-WK auf Tiefe  $\tau$ :

Fehler-WK FASTCUT( $G_1$ )

$$p(\tau + 1) = 1 - \underbrace{\left(1 - \frac{1}{2}p(\tau)\right)^2}_{\text{Fehler-WK auf Tiefe } \tau + 1} = p(\tau) - \frac{p(\tau)^2}{4}, \text{ wobei } p(0) = 1.$$

Fehler-WK auf Tiefe  $\tau + 1$

# Erfolgswahrscheinlichkeit (III)

Sei  $p(\tau)$  untere Schranke für die Erfolgs-WK:

$$p(\tau + 1) = p(\tau) - \frac{p(\tau)^2}{4} \quad \text{mit } p(0) = 1$$

# Erfolgswahrscheinlichkeit (III)

Sei  $p(\tau)$  untere Schranke für die Erfolgs-WK:

$$p(\tau + 1) = p(\tau) - \frac{p(\tau)^2}{4} \quad \text{mit } p(0) = 1$$

Setze  $q(\tau) = 4/p(\tau) - 1$ . (Äquivalent:  $p(\tau) = 4/(q(\tau) + 1)$ )

# Erfolgswahrscheinlichkeit (III)

Sei  $p(\tau)$  untere Schranke für die Erfolgs-WK:

$$p(\tau + 1) = p(\tau) - \frac{p(\tau)^2}{4} \quad \text{mit } p(0) = 1$$

Setze  $q(\tau) = 4/p(\tau) - 1$ . (Äquivalent:  $p(\tau) = 4/(q(\tau) + 1)$ )

Einsetzen ergibt  $q(0) = 3$  und

# Erfolgswahrscheinlichkeit (III)

Sei  $p(\tau)$  untere Schranke für die Erfolgs-WK:

$$p(\tau + 1) = p(\tau) - \frac{p(\tau)^2}{4} \quad \text{mit } p(0) = 1$$

Setze  $q(\tau) = 4/p(\tau) - 1$ . (Äquivalent:  $p(\tau) = 4/(q(\tau) + 1)$ )

Einsetzen ergibt  $q(0) = 3$  und

$$q(\tau + 1) = \frac{4}{p(\tau + 1)} - 1 = \frac{4}{p(\tau) - p(\tau)^2/4} - 1$$

# Erfolgswahrscheinlichkeit (III)

Sei  $p(\tau)$  untere Schranke für die Erfolgs-WK:

$$p(\tau + 1) = p(\tau) - \frac{p(\tau)^2}{4} \quad \text{mit } p(0) = 1$$

Setze  $q(\tau) = 4/p(\tau) - 1$ . (Äquivalent:  $p(\tau) = 4/(q(\tau) + 1)$ )

Einsetzen ergibt  $q(0) = 3$  und

$$\begin{aligned} q(\tau + 1) &= \frac{4}{p(\tau + 1)} - 1 = \frac{4}{p(\tau) - p(\tau)^2/4} - 1 \\ &= \frac{1}{1/(q(\tau) + 1) - 1/(q(\tau) + 1)^2} - 1 \end{aligned}$$

# Erfolgswahrscheinlichkeit (III)

Sei  $p(\tau)$  untere Schranke für die Erfolgs-WK:

$$p(\tau + 1) = p(\tau) - \frac{p(\tau)^2}{4} \quad \text{mit } p(0) = 1$$

Setze  $q(\tau) = 4/p(\tau) - 1$ . (Äquivalent:  $p(\tau) = 4/(q(\tau) + 1)$ )

Einsetzen ergibt  $q(0) = 3$  und

$$\begin{aligned} q(\tau + 1) &= \frac{4}{p(\tau + 1)} - 1 = \frac{4}{p(\tau) - p(\tau)^2/4} - 1 \\ &= \frac{1}{1/(q(\tau) + 1) - 1/(q(\tau) + 1)^2} - 1 \\ &= \frac{(q(\tau) + 1)^2 - q(\tau)}{q(\tau)} = q(\tau) + 1 + \frac{1}{q(\tau)} \end{aligned}$$



# Erfolgswahrscheinlichkeit (IV)

Es gilt  $q(\tau + 1) = q(\tau) + 1 + \frac{1}{q(\tau)}$  mit  $q(0) = 3$ .

Zeige per Induktion, dass  $\tau < q(\tau) < \tau + h_{\tau-1} + c$  für  $\tau \geq 2$  und genügend große Konstante  $c > 0$ .

# Erfolgswahrscheinlichkeit (IV)

Es gilt  $q(\tau + 1) = q(\tau) + 1 + \frac{1}{q(\tau)}$  mit  $q(0) = 3$ .

Zeige per Induktion, dass  $\tau < q(\tau) < \tau + h_{\tau-1} + c$  für  $\tau \geq 2$  und genügend große Konstante  $c > 0$ .

*i*-te harmonische Zahl  
 $h_i := \sum_{j=1}^i \frac{1}{j} = \Theta(\log i)$

# Erfolgswahrscheinlichkeit (IV)

Es gilt  $q(\tau + 1) = q(\tau) + 1 + \frac{1}{q(\tau)}$  mit  $q(0) = 3$ .

Zeige per Induktion, dass  $\tau < q(\tau) < \tau + h_{\tau-1} + c$  für  $\tau \geq 2$  und genügend große Konstante  $c > 0$ .

*Induktionsanfang* ( $\tau = 2$ ):  
 $\leq q(2) \leq c$

*$i$ -te harmonische Zahl  
 $h_i := \sum_{j=1}^i \frac{1}{j} = \Theta(\log i)$*

# Erfolgswahrscheinlichkeit (IV)

Es gilt  $q(\tau + 1) = q(\tau) + 1 + \frac{1}{q(\tau)}$  mit  $q(0) = 3$ .

Zeige per Induktion, dass  $\tau < q(\tau) < \tau + h_{\tau-1} + c$  für  $\tau \geq 2$  und genügend große Konstante  $c > 0$ .

*Induktionsanfang* ( $\tau = 2$ ):

$$3 = q(0) \leq q(2) \leq c$$

*$i$ -te harmonische Zahl  
 $h_i := \sum_{j=1}^i \frac{1}{j} = \Theta(\log i)$*

# Erfolgswahrscheinlichkeit (IV)

Es gilt  $q(\tau + 1) = q(\tau) + 1 + \frac{1}{q(\tau)}$  mit  $q(0) = 3$ .

Zeige per Induktion, dass  $\tau < q(\tau) < \tau + h_{\tau-1} + c$  für  $\tau \geq 2$  und genügend große Konstante  $c > 0$ .

*Induktionsanfang* ( $\tau = 2$ ):

$$2 < 3 = q(0) \leq q(2) \leq c$$

*i*-te harmonische Zahl  
 $h_i := \sum_{j=1}^i \frac{1}{j} = \Theta(\log i)$

# Erfolgswahrscheinlichkeit (IV)

Es gilt  $q(\tau + 1) = q(\tau) + 1 + \frac{1}{q(\tau)}$  mit  $q(0) = 3$ .

Zeige per Induktion, dass  $\tau < q(\tau) < \tau + h_{\tau-1} + c$  für  $\tau \geq 2$  und genügend große Konstante  $c > 0$ .

*Induktionsanfang* ( $\tau = 2$ ):

$$2 < 3 = q(0) \leq q(2) \leq c$$

*Induktionsschritt* ( $\tau \rightarrow \tau + 1$ ):

$$q(\tau + 1) \geq q(\tau) + 1$$

*i*-te harmonische Zahl  
 $h_i := \sum_{j=1}^i \frac{1}{j} = \Theta(\log i)$

# Erfolgswahrscheinlichkeit (IV)

Es gilt  $q(\tau + 1) = q(\tau) + 1 + \frac{1}{q(\tau)}$  mit  $q(0) = 3$ .

Zeige per Induktion, dass  $\tau < q(\tau) < \tau + h_{\tau-1} + c$  für  $\tau \geq 2$  und genügend große Konstante  $c > 0$ .

*Induktionsanfang* ( $\tau = 2$ ):

$$2 < 3 = q(0) \leq q(2) \leq c$$

*Induktionsschritt* ( $\tau \rightarrow \tau + 1$ ):

$$q(\tau + 1) \geq q(\tau) + 1$$

*i*-te harmonische Zahl  
 $h_i := \sum_{j=1}^i \frac{1}{j} = \Theta(\log i)$

# Erfolgswahrscheinlichkeit (IV)

Es gilt  $q(\tau + 1) = q(\tau) + 1 + \frac{1}{q(\tau)}$  mit  $q(0) = 3$ .

Zeige per Induktion, dass  $\tau < q(\tau) < \tau + h_{\tau-1} + c$  für  $\tau \geq 2$  und genügend große Konstante  $c > 0$ .

*Induktionsanfang* ( $\tau = 2$ ):

$$2 < 3 = q(0) \leq q(2) \leq c$$

*Induktionsschritt* ( $\tau \rightarrow \tau + 1$ ):

$$q(\tau + 1) \geq q(\tau) + 1 \stackrel{\text{IV}}{>}$$

*i*-te harmonische Zahl  
 $h_i := \sum_{j=1}^i \frac{1}{j} = \Theta(\log i)$



# Erfolgswahrscheinlichkeit (IV)

Es gilt  $q(\tau + 1) = q(\tau) + 1 + \frac{1}{q(\tau)}$  mit  $q(0) = 3$ .

Zeige per Induktion, dass  $\tau < q(\tau) < \tau + h_{\tau-1} + c$  für  $\tau \geq 2$  und genügend große Konstante  $c > 0$ .

*Induktionsanfang* ( $\tau = 2$ ):

$$2 < 3 = q(0) \leq q(2) \leq c$$

*Induktionsschritt* ( $\tau \rightarrow \tau + 1$ ):

$$q(\tau + 1) \geq q(\tau) + 1 \stackrel{\text{IV}}{>} \tau + 1$$

*i*-te harmonische Zahl  
 $h_i := \sum_{j=1}^i \frac{1}{j} = \Theta(\log i)$

# Erfolgswahrscheinlichkeit (IV)

Es gilt  $q(\tau + 1) = q(\tau) + 1 + \frac{1}{q(\tau)}$  mit  $q(0) = 3$ .

Zeige per Induktion, dass  $\tau < q(\tau) < \tau + h_{\tau-1} + c$  für  $\tau \geq 2$  und genügend große Konstante  $c > 0$ .

*Induktionsanfang* ( $\tau = 2$ ):

$$2 < 3 = q(0) \leq q(2) \leq c$$

*Induktionsschritt* ( $\tau \rightarrow \tau + 1$ ):

$$q(\tau + 1) \geq q(\tau) + 1 \stackrel{\text{IV}}{>} \tau + 1$$

$$q(\tau + 1) \stackrel{\text{s.o.}}{=} q(\tau) + 1 + \frac{1}{q(\tau)}$$

*i-te harmonische Zahl*  
 $h_i := \sum_{j=1}^i \frac{1}{j} = \Theta(\log i)$

# Erfolgswahrscheinlichkeit (IV)

Es gilt  $q(\tau + 1) = q(\tau) + 1 + \frac{1}{q(\tau)}$  mit  $q(0) = 3$ .

Zeige per Induktion, dass  $\tau < q(\tau) < \tau + h_{\tau-1} + c$  für  $\tau \geq 2$  und genügend große Konstante  $c > 0$ .

*Induktionsanfang* ( $\tau = 2$ ):

$$2 < 3 = q(0) \leq q(2) \leq c$$

*Induktionsschritt* ( $\tau \rightarrow \tau + 1$ ):

$$q(\tau + 1) \geq q(\tau) + 1 \stackrel{\text{IV}}{>} \tau + 1$$

$$q(\tau + 1) \stackrel{\text{IV}}{=} q(\tau) + 1 + \frac{1}{q(\tau)} <$$

S.O.

*i*-te harmonische Zahl  
 $h_i := \sum_{j=1}^i \frac{1}{j} = \Theta(\log i)$

# Erfolgswahrscheinlichkeit (IV)

Es gilt  $q(\tau + 1) = q(\tau) + 1 + \frac{1}{q(\tau)}$  mit  $q(0) = 3$ .

Zeige per Induktion, dass  $\tau < q(\tau) < \tau + h_{\tau-1} + c$  für  $\tau \geq 2$  und genügend große Konstante  $c > 0$ .

*Induktionsanfang* ( $\tau = 2$ ):

$$2 < 3 = q(0) \leq q(2) \leq c$$

*i-te harmonische Zahl*  
 $h_i := \sum_{j=1}^i \frac{1}{j} = \Theta(\log i)$

*Induktionsschritt* ( $\tau \rightarrow \tau + 1$ ):

$$q(\tau + 1) \geq q(\tau) + 1 \stackrel{\text{IV}}{>} \tau + 1$$

$$q(\tau + 1) \stackrel{\text{S.O.}}{=} q(\tau) + 1 + \frac{1}{q(\tau)} \stackrel{\text{IV}}{<} (\tau + h_{\tau-1} + c) + 1 + \frac{1}{\tau} =$$

# Erfolgswahrscheinlichkeit (IV)

Es gilt  $q(\tau + 1) = q(\tau) + 1 + \frac{1}{q(\tau)}$  mit  $q(0) = 3$ .

Zeige per Induktion, dass  $\tau < q(\tau) < \tau + h_{\tau-1} + c$  für  $\tau \geq 2$  und genügend große Konstante  $c > 0$ .

*Induktionsanfang* ( $\tau = 2$ ):

$$2 < 3 = q(0) \leq q(2) \leq c$$

*i-te harmonische Zahl*  
 $h_i := \sum_{j=1}^i \frac{1}{j} = \Theta(\log i)$

*Induktionsschritt* ( $\tau \rightarrow \tau + 1$ ):

$$q(\tau + 1) \geq q(\tau) + 1 \stackrel{\text{IV}}{>} \tau + 1$$

$$q(\tau + 1) \stackrel{\text{S.O.}}{=} q(\tau) + 1 + \frac{1}{q(\tau)} \stackrel{\text{IV}}{<} (\tau + h_{\tau-1} + c) + 1 + \frac{1}{\tau} = (\tau + 1) + h_{\tau} + c$$

# Erfolgswahrscheinlichkeit (IV)

Es gilt  $q(\tau + 1) = q(\tau) + 1 + \frac{1}{q(\tau)}$  mit  $q(0) = 3$ .

Zeige per Induktion, dass  $\tau < q(\tau) < \tau + h_{\tau-1} + c$  für  $\tau \geq 2$  und genügend große Konstante  $c > 0$ .

*Induktionsanfang* ( $\tau = 2$ ):

$$2 < 3 = q(0) \leq q(2) \leq c$$

*i-te harmonische Zahl*  
 $h_i := \sum_{j=1}^i \frac{1}{j} = \Theta(\log i)$

*Induktionsschritt* ( $\tau \rightarrow \tau + 1$ ):

$$q(\tau + 1) \geq q(\tau) + 1 \stackrel{\text{IV}}{>} \tau + 1$$

$$q(\tau + 1) \stackrel{\text{S.O.}}{=} q(\tau) + 1 + \frac{1}{q(\tau)} \stackrel{\text{IV}}{<} (\tau + h_{\tau-1} + c) + 1 + \frac{1}{\tau} = (\tau + 1) + h_{\tau} + c$$

$$\Rightarrow p(\tau) = \frac{4}{q(\tau)+1} = \Omega\left(\frac{1}{\log n}\right)$$

□

# Erfolgswahrscheinlichkeit (IV)

Es gilt  $q(\tau + 1) = q(\tau) + 1 + \frac{1}{q(\tau)}$  mit  $q(0) = 3$ .

Zeige per Induktion, dass  $\tau < q(\tau) < \tau + h_{\tau-1} + c$  für  $\tau \geq 2$  und genügend große Konstante  $c > 0$ .

*Induktionsanfang* ( $\tau = 2$ ):

$$2 < 3 = q(0) \leq q(2) \leq c$$

*i-te harmonische Zahl*  
 $h_i := \sum_{j=1}^i \frac{1}{j} = \Theta(\log i)$

*Induktionsschritt* ( $\tau \rightarrow \tau + 1$ ):

$$q(\tau + 1) \geq q(\tau) + 1 \stackrel{\text{IV}}{>} \tau + 1$$

$$q(\tau + 1) \stackrel{\text{S.O.}}{=} q(\tau) + 1 + \frac{1}{q(\tau)} \stackrel{\text{IV}}{<} (\tau + h_{\tau-1} + c) + 1 + \frac{1}{\tau} = (\tau + 1) + h_{\tau} + c$$

$$\Rightarrow p(\tau) = \frac{4}{q(\tau)+1} = \Omega\left(\frac{1}{\log n}\right)$$

□

$\tau = \Theta(\log n)$  und  $q(\tau) \in O(\tau)$

# Erfolgswahrscheinlichkeit (IV)

Es gilt  $q(\tau + 1) = q(\tau) + 1 + \frac{1}{q(\tau)}$  mit  $q(0) = 3$ .

Zeige per Induktion, dass  $\tau < q(\tau) < \tau + h_{\tau-1} + c$  für  $\tau \geq 2$  und genügend große Konstante  $c > 0$ .

*Induktionsanfang* ( $\tau = 2$ ):

$$2 < 3 = q(0) \leq q(2) \leq c$$

*Induktionsschritt* ( $\tau \rightarrow \tau + 1$ ):

$$q(\tau + 1) \geq q(\tau) + 1 \stackrel{\text{IV}}{>} \tau + 1$$

$$q(\tau + 1) \stackrel{\text{S.O.}}{=} q(\tau) + 1 + \frac{1}{q(\tau)} \stackrel{\text{IV}}{<} (\tau + h_{\tau-1} + c) + 1 + \frac{1}{\tau} = (\tau + 1) + h_{\tau} + c$$

$$\Rightarrow p(\tau) = \frac{4}{q(\tau)+1} = \Omega\left(\frac{1}{\log n}\right)$$

$$\tau = \Theta(\log n) \text{ und } q(\tau) \in O(\tau)$$

*i*-te harmonische Zahl  
 $h_i := \sum_{j=1}^i \frac{1}{j} = \Theta(\log i)$

**Bem.** Wenn wir unten nur z.B.  $q(\tau) \leq 27\tau$  brauchen – warum zeigen wir oben viel mehr?

□



# Verringerung der Fehlerwahrscheinlichkeit

Wiederhole FASTCUT  $\log^2 n$  mal.

# Verringerung der Fehlerwahrscheinlichkeit

Wiederhole FASTCUT  $\log^2 n$  mal.

Die Erfolgswahrscheinlichkeit pro Ausführung ist  $\geq C / \log n$  für ein  $C > 0$ .

# Verringerung der Fehlerwahrscheinlichkeit

Wiederhole FASTCUT  $\log^2 n$  mal.

Die Erfolgswahrscheinlichkeit pro Ausführung ist  $\geq C / \log n$  für ein  $C > 0$ .

Gesamtfehlerwahrscheinlichkeit:

$$\left(1 - \frac{C}{\log n}\right)^{\log^2 n} \leq e^{-C \log n} = \frac{1}{n^C}$$

$$1 + x \leq e^x$$

für alle  $x \in \mathbb{R}$

# Verringerung der Fehlerwahrscheinlichkeit

Wiederhole FASTCUT  $\log^2 n$  mal.

Die Erfolgswahrscheinlichkeit pro Ausführung ist  $\geq C / \log n$  für ein  $C > 0$ .

Gesamtfehlerwahrscheinlichkeit:

$$\left(1 - \frac{C}{\log n}\right)^{\log^2 n} \leq e^{-C \log n} = \frac{1}{n^C}$$

$$1 + x \leq e^x$$

für alle  $x \in \mathbb{R}$

**Satz.** Es gibt einen randomisierten Algorithmus für MINCUT mit  $O(n^2 \log^3 n)$  Laufzeit, der mit hoher Wahrscheinlichkeit einen kleinsten Schnitt ermittelt.

# Und noch schneller!

**Satz.** Es gibt einen randomisierten Algorithmus für MINCUT mit  $O(m \log^3 n)$  Laufzeit, der mit hoher Wahrscheinlichkeit einen kleinsten Schnitt ermittelt.

*Beweis.* [Karger, STOC'96]



# Und noch schneller!

**Satz.** Es gibt einen randomisierten Algorithmus für MINCUT mit  $O(m \log^3 n)$  Laufzeit, der mit hoher Wahrscheinlichkeit einen kleinsten Schnitt ermittelt.

*Beweis.* [Karger, STOC'96]



Diese Laufzeit ist auch für Graphen mit  $m = o(n^2)$  Kanten fast linear!

# Und noch schneller!

**Satz.** Es gibt einen randomisierten Algorithmus für MINCUT mit  $O(m \log^3 n)$  Laufzeit, der mit hoher Wahrscheinlichkeit einen kleinsten Schnitt ermittelt.

*Beweis.* [Karger, STOC'96]



Diese Laufzeit ist auch für Graphen mit  $m = o(n^2)$  Kanten fast linear!

Zum Vergleich:

# Und noch schneller!

**Satz.** Es gibt einen randomisierten Algorithmus für MINCUT mit  $O(m \log^3 n)$  Laufzeit, der mit hoher Wahrscheinlichkeit einen kleinsten Schnitt ermittelt.

*Beweis.* [Karger, STOC'96]



Diese Laufzeit ist auch für Graphen mit  $m = o(n^2)$  Kanten fast linear!

Zum Vergleich:

**Satz.** Es gibt einen einfachen deterministischen Algorithmus für MINCUT mit  $O(mn + n^2 \log n)$  Laufzeit.

[Stoer & Wagner, JACM'97]



# Und noch schneller!

**Satz.** Es gibt einen randomisierten Algorithmus für MINCUT mit  $O(m \log^3 n)$  Laufzeit, der mit hoher Wahrscheinlichkeit einen kleinsten Schnitt ermittelt.

*Beweis.* [Karger, STOC'96]



Diese Laufzeit ist auch für Graphen mit  $m = o(n^2)$  Kanten fast linear!

Zum Vergleich:

**Satz.** Es gibt einen einfachen deterministischen Algorithmus für MINCUT mit  $O(mn + n^2 \log n)$  Laufzeit.

[Stoer & Wagner, JACM'97]

... ist in LEDA (C++) und jgraphT (Java) implementiert.