

ICPC

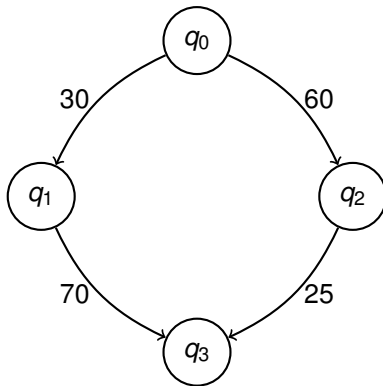
Graphen 3

Tobias, Julian, Jakob, Tobias | 6. Juni 2018

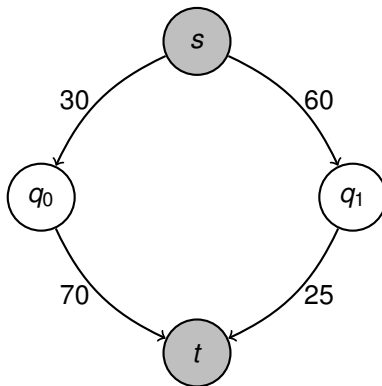
ITI WAGNER, IPD TICHY



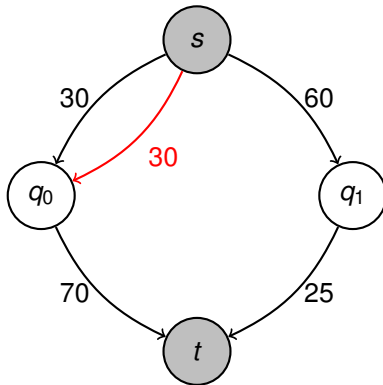
- 1 Tobias
- 2 Jakob
- 3 Julian
- 4 Tobias T



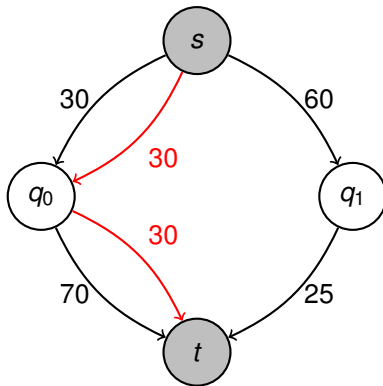
Gegeben gerichteter Graph



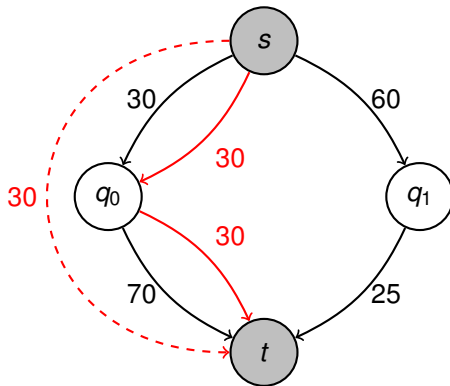
s: source, t:sink



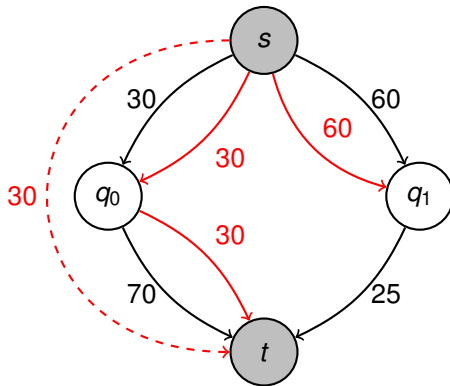
Fluss

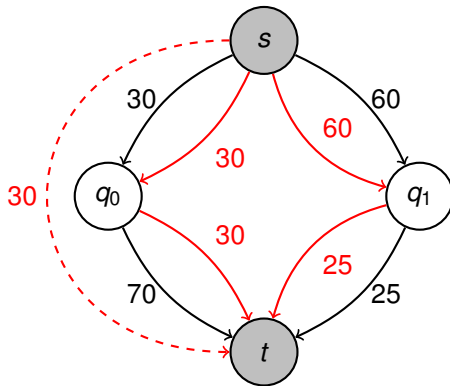


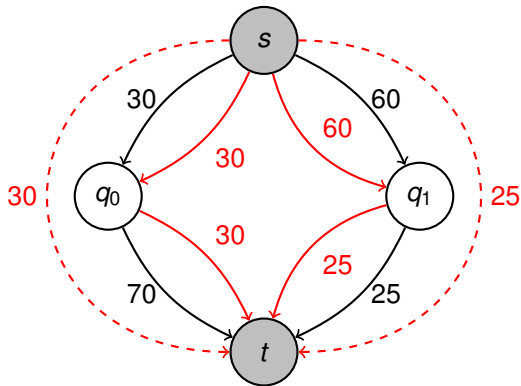
Flusserhaltung

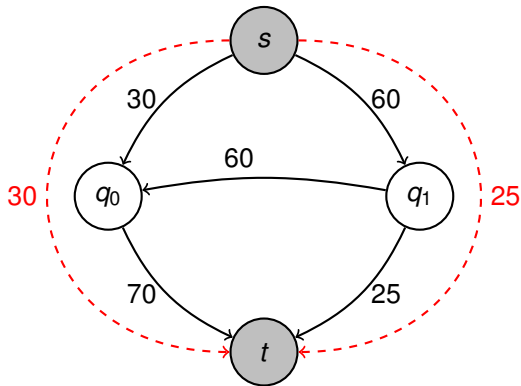


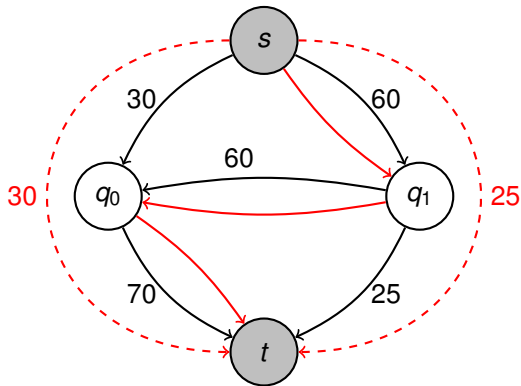
Wert eines s-t-Flusses

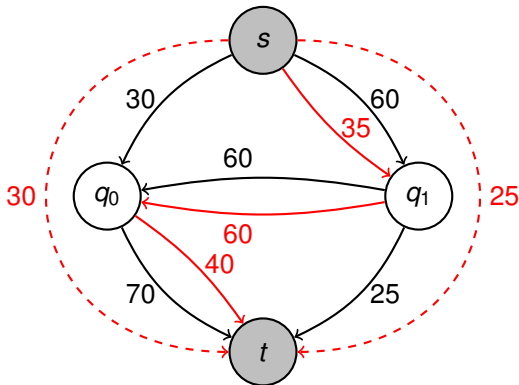




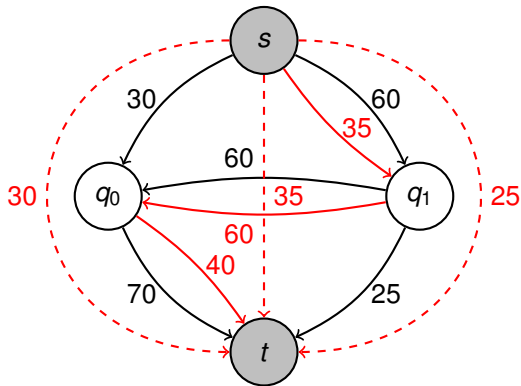








Exzess: Werte entsprechend der Kantenkapazität abzüglich bereits vorhandener Flüsse

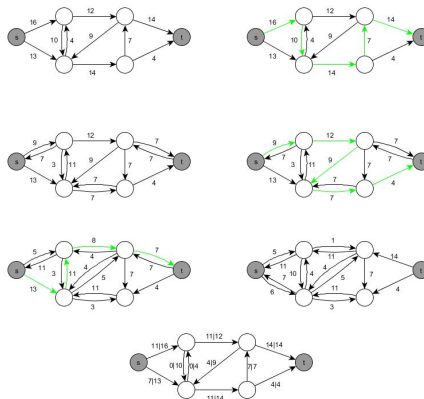


- Schwierigkeit im Erkennen der Aufgaben
Proble UVa xxx

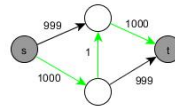
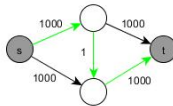
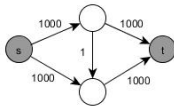
- Schwierigkeit im Erkennen der Aufgaben
- Seit 2013 vermehrtes vorkommen in contests
decider Problem

- Erklärung augmentierender/erweiternder Weg

■ Erklärung Ford-Fulkerson

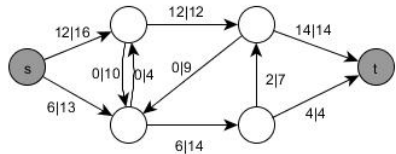
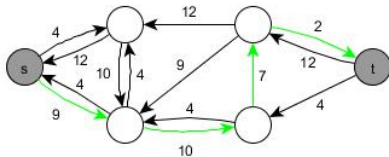
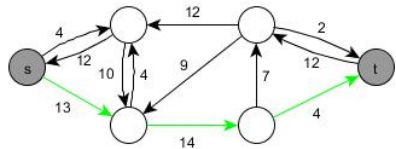
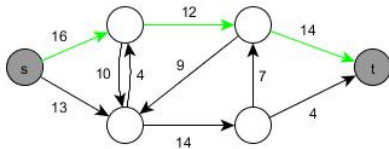


- Laufzeit - nicht benutzen



Edmond-Karp Algorithmus

■ Pseudocode + Beispiel



■ Implementierungsdetails

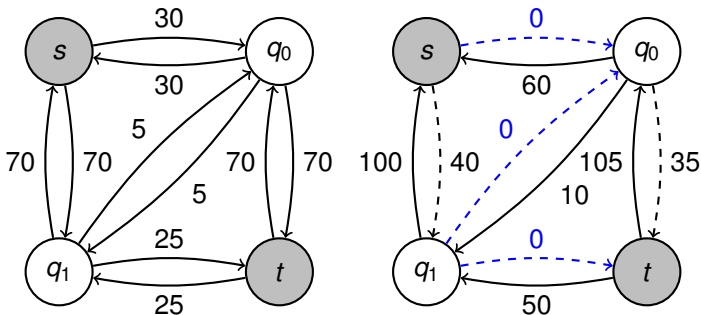
Min-Cut

- Definiere Schnitt $C = (S - \text{Komponente}, T - \text{Komponente})$ als Partition von $V \in G$, wobei $s \in S - \text{Komponente}$ und $t \in T - \text{Komponente}$
- Weiter sei die Schnittmenge
 $c = \{(u, v) \in E \mid u \in S - \text{Komponente} \wedge v \in T - \text{Komponente}\}$
- Wähle c so, dass Max Flow von s nach t 0 ist, für $E' = E \setminus c$

Max-Flow-Min-Cut-Theorem

- Ein maximaler Fluss im Netzwerk hat genau den Wert eines minimalen Schnitts.

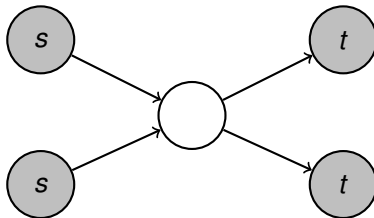
■ Bsp.:



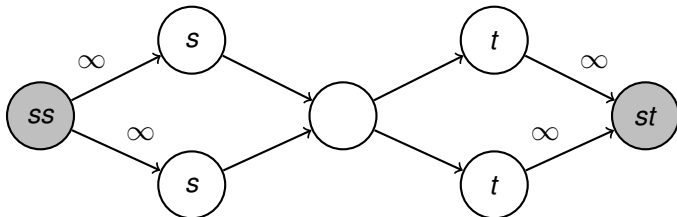
■ Hier

- $C = (\{s, q_1\}, \{t, q_0\})$
- $c = \{(s, q_0), (q_1, q_0), (q_1, t)\}$

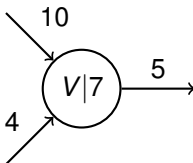
- Gegeben sei folgende Situation:



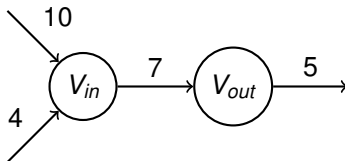
- Problem: Max-Flow Algorithmus kann nur mit einer Quelle und einer Senke arbeiten.
- Lösung: Erstelle Super-Quelle und Super-Senke und verbinde alle Quellen und Senken mit Kantengewicht ∞



- Gegeben sind Knoten mit Kapazität.
- Bsp.:



- Gegeben sind Knoten mit Kapazität.
- Bsp.:



- Erkennen eines Netzwerkfluss-Problems nicht immer einfach
- Was hilft?
 - Übung
 - Übung
 - ...

- Erkennen eines Netzwerkfluss-Problems nicht immer einfach
- Was hilft?
 - Übung
 - Übung
 - ...

- Erkennen eines Netzwerkfluss-Problems nicht immer einfach
- Was hilft?
 - Übung
 - Übung
 - ...

- Situation: Die Titanic ist gesunken. Es soll ermittelt werden wie viele Menschen gerettet werden können.
- Eingabe: X, Y, P mit X, Y Dimension der Fläche ($1 \leq X, Y \leq 30$) und P ($P \leq 10$) die Anzahl von Personen, welche gleichzeitig auf ein Holzbrett können.

Symbol	Bedeutung
*	Menschen auf Treibeis
~	Eiskaltes Wasser
.	Trebbeis
@	Großer Eisberg
#	Großes Holzbrett

- Gegeben sei nun folgende Eingabe:

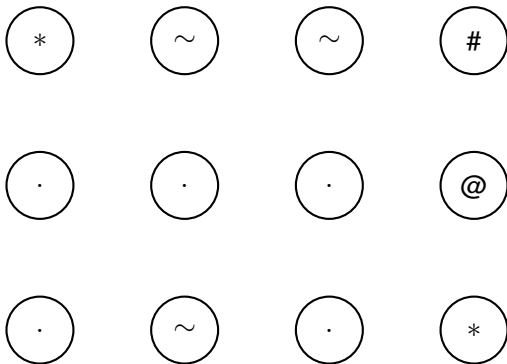
*	~	~	#
.	.	.	@
.	~	.	*

- Wandle in Graphen um...

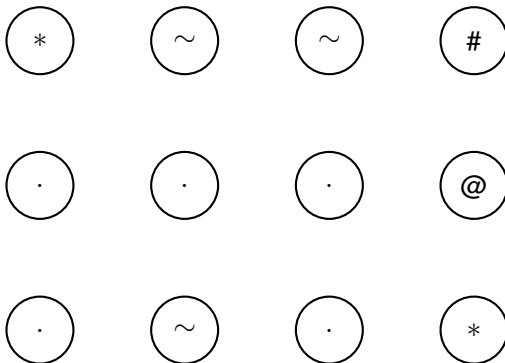
- Gegeben sei nun folgende Eingabe:

*	~	~	#
.	.	.	@
.	~	.	*

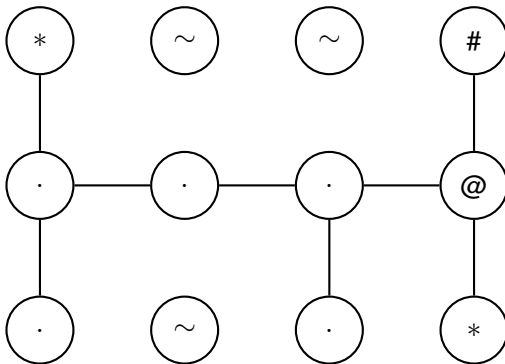
- Wandle in Graphen um...



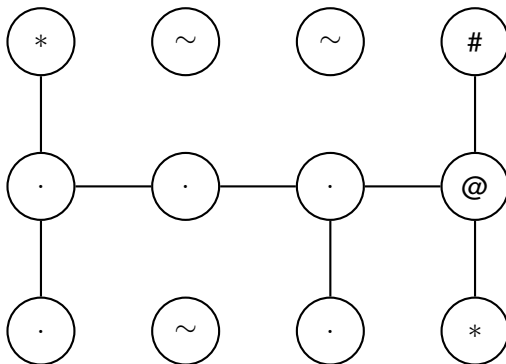
■ Verbinde alle Knoten, über die ein Weg möglich ist...



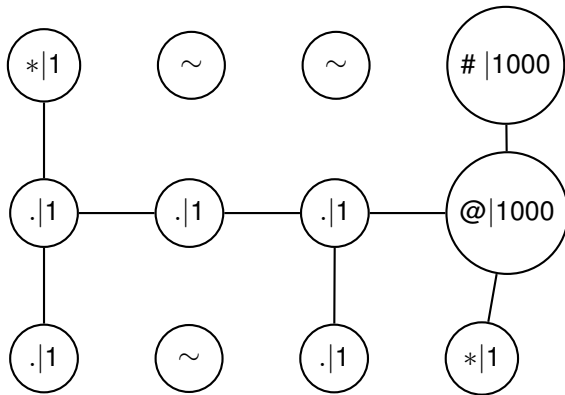
- Verbinde alle Knoten, über die ein Weg möglich ist...



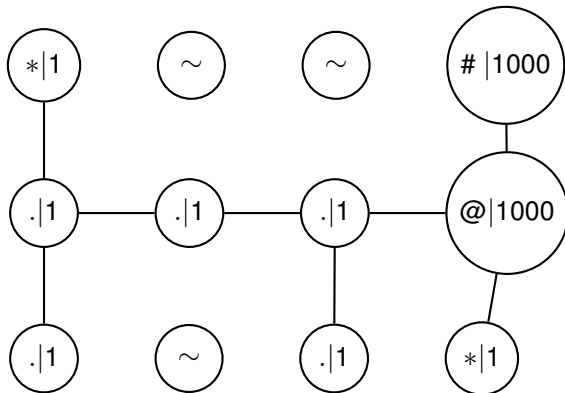
■ Füge Knotengewichte hinzu...



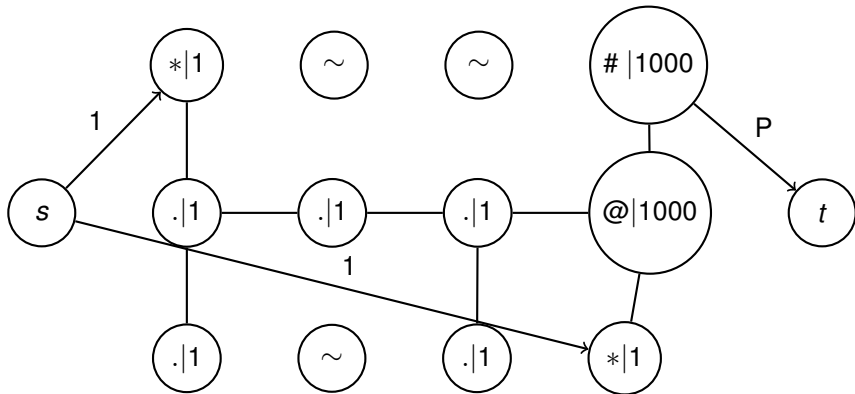
- Füge Knotengewichte hinzu...



■ Verbinde alle Menschen mit s und alle Holzbretter mit t...



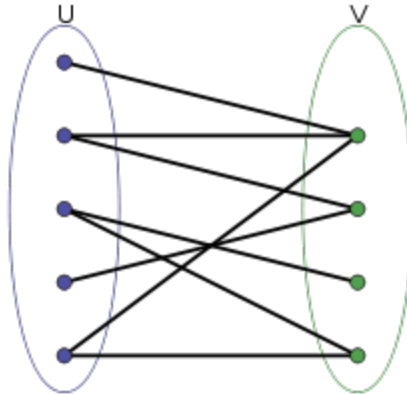
- Verbinde alle Menschen mit s und alle Holzbretter mit t...



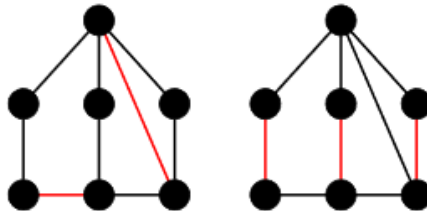
■ Bem.: Knotengewichte müssen noch aufgelöst werden

Bipartiter Graph

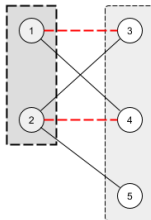
■ Bipartiter Graph



- Definitionen: Matching, maximales Matching, kardinalitätsmaximales Matching, perfektes Matching



- Kurz auf Laufzeit eingehen
- Beispiel: Primzahlen (Competitive Programming 3, Seite 180)
- Definitionen: Max Independent Set, Min Vertex Cover, Königs Theorem: $|\text{Min Vertex Cover}| = |\text{größtes Matching}|$



- Beispiel: Guardian of Decency (Competitive Programming 3, Seite 182)
- (Je nach verbleibender Zeit:) noch mehr Graphentheorie: bipartit \Leftrightarrow keine ungeraden Kreise, ...