Softwarepraktikum

3. Frontalveranstaltung

17.11.2017

ES Real-Time Systems Lab
Prof. Dr. rer. nat. Andy Schürr
Dept. of Electrical Engineering and Information Technology
Dept. of Computer Science (adjunct Professor)

Dr. Malte Lochau

Malte.Lochau@es.tu-darmstadt.de

www.es.tu-darmstadt.de

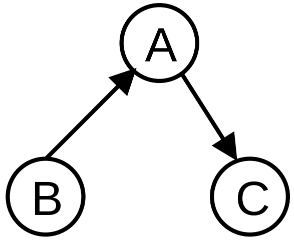
Gliederung

TECHNISCHE UNIVERSITÄT DARMSTADT

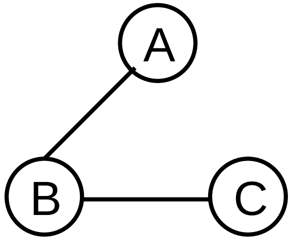
- Definition und Eigenschaften von Graphen
- Flüsse in Netzwerken
- Residualnetzwerke
- Aufgabenblock 3

Graphen

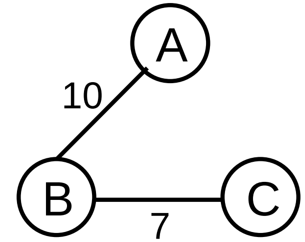
- Ein Graph besteht aus einer Menge von Knoten und einer Menge von Kanten
- Kanten verbinden jeweils einen Start- und einen Zielknoten miteinander
- Kanten können gerichtet oder ungerichtet sein
- Kanten können gewichtet oder ungewichtet sein



Graph mit gerichteten Kanten



Graph mit ungerichteten Kanten

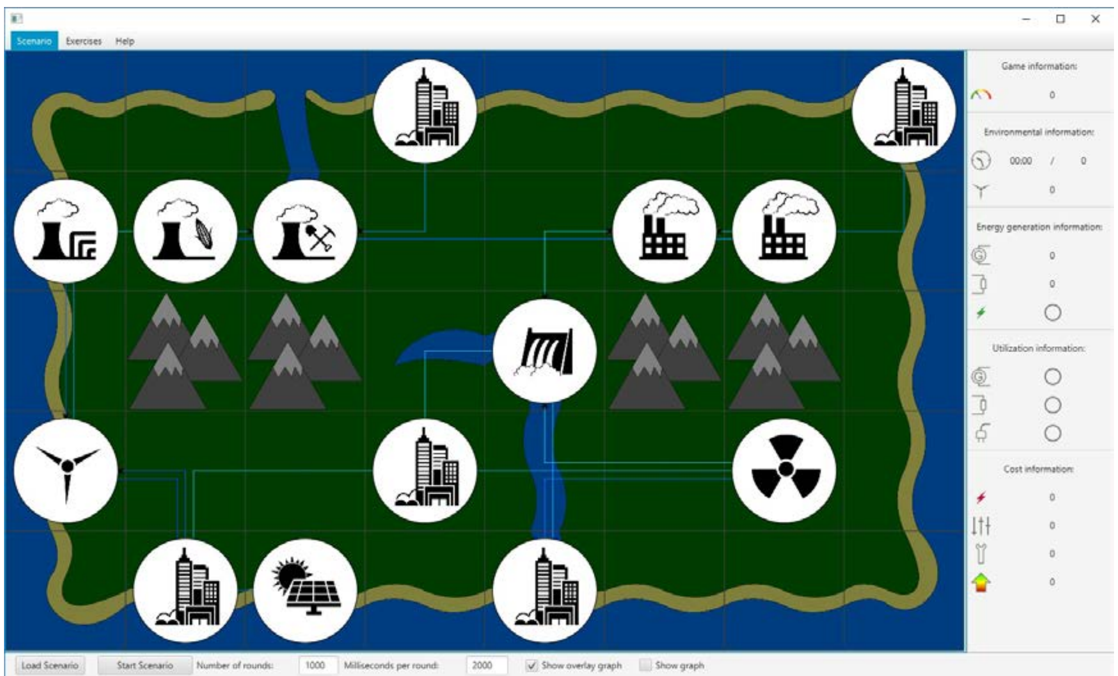


Graph mit ungerichteten und gewichteten Kanten

Beispiel: Flussnetze

EVS

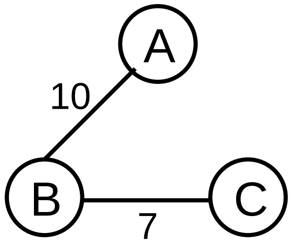
- Produzenten, Konsumenten und Umspannwerke sind Knoten
- Leitungen sind **ungerichtete, gewichtete** Kanten
- Kantengewichte sind z.B. Leitungskapazitäten und Energieflüsse



Definition von Graphen

Ein gewichteter, ungerichteter Graph G ist ein Tripel $G = (V, E, w)$

- V ist eine endliche, nicht-leere Menge (Knotenmenge)
- $E \subseteq V \times V$ ist eine symmetrische Relation (Kantenmenge)
- w ist eine Funktion (Kantengewichte), z.B. $w : E \rightarrow \mathbb{Z}^+$
  Es gilt: $w(v, v') = w(v', v)$ für alle $(v, v') \in E$



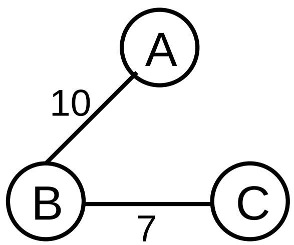
$$V = \{ A, B, C \}$$

$$E = \{ (A, B), (B, A), (B, C), (C, B) \}$$

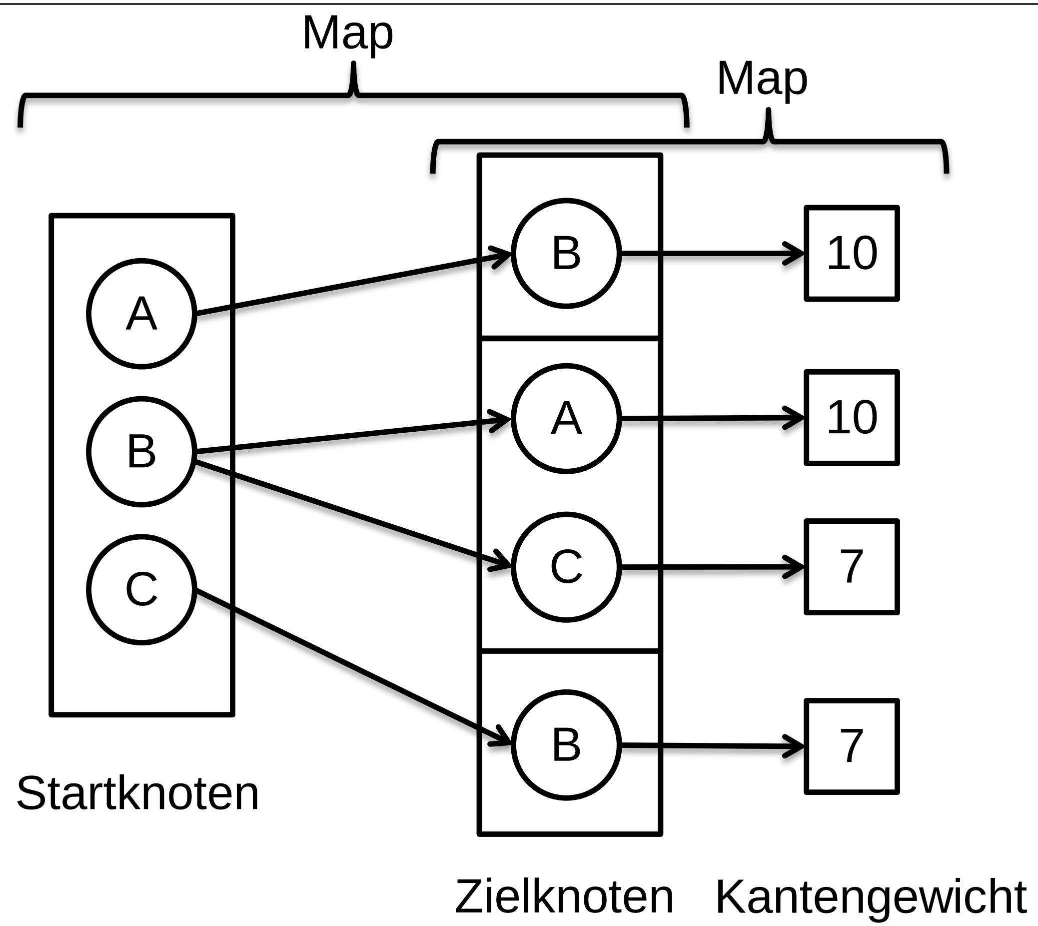
$$w(e) = \begin{cases} 10, \text{ falls e = (A,B) oder e = (B,A)} \\ 7, \text{ falls e = (B,C) oder e = (C,B)} \end{cases}$$

Datenstruktur zur Repräsentation von Graphen

TECHNISCHE UNIVERSITÄT DARMSTADT



- Äußere Map:
  - Keys: Startknoten
  - Values: innere Map
- Innere Map:
  - Keys: Zielknoten der abgehenden Kanten
  - Values: Kantengewichte (bzw. Kantenobjekte)



Gliederung

- Definition und Eigenschaften von Graphen
- Flüsse in Netzwerken
- Residualnetzwerke
- Aufgabenblock 3

Definition: Fluss-Netzwerke und Flüsse

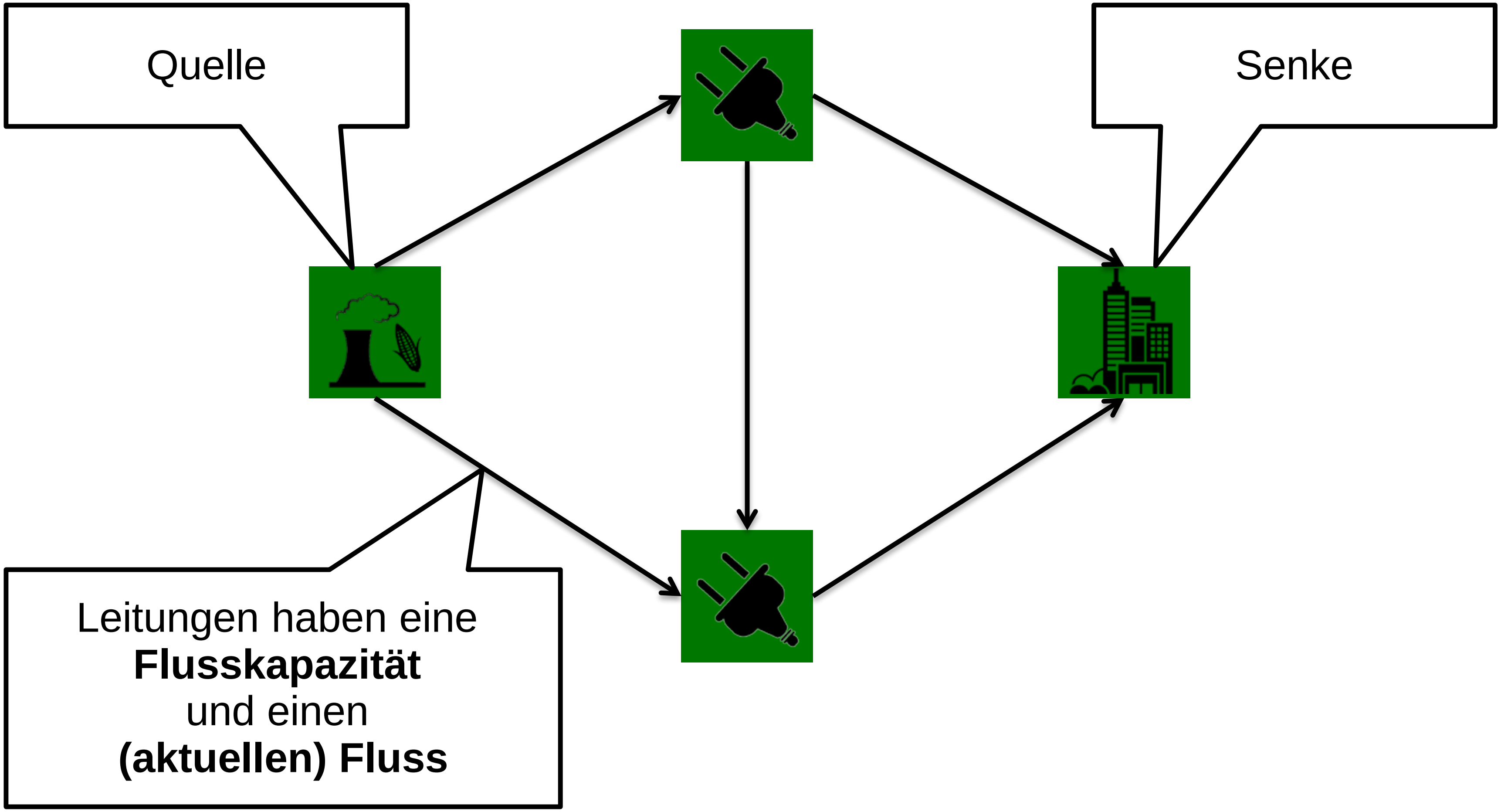
Ein **Fluss-Netzwerk** ist ein Tupel $N = (G, s, t)$

- $G = (V, E, w)$ ist ein gerichteter, gewichteter Graph, wobei w eine Gewichtsfunktion ist, die jeder Kante ein Tupel wie folgt zuordnet
- $w(e) = (f(e), c(e))$ mit $f(e)$ Fluss über die Kante e,
  $c(e)$ Kapazität der Kante e
- $s \in V$ ist der Startknoten
- $t \in V$ ist der Zielknoten

Für einen **Fluss** $f: E \rightarrow \mathbb{Z}^+$ in einem Fluss-Netzwerk N gilt

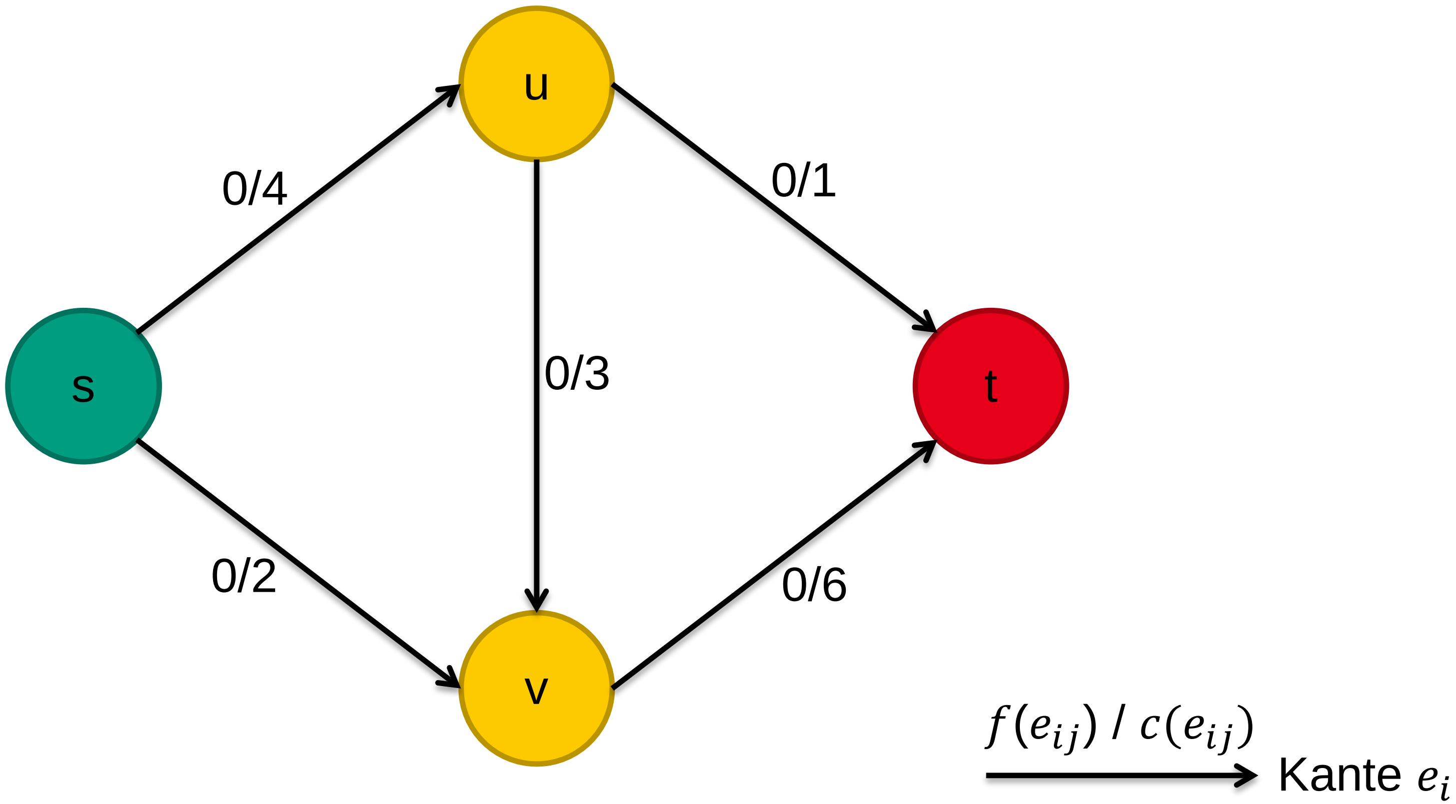
- Für alle $e \in E$: $0 \leq f(e) \leq c(e)$
  (**Kapazitätsbeschränkung**)
- Für alle $x \in V \setminus \{s, t\}$: $\sum_{\{e\prime=(x,x')\in E\}} f(e') = \sum_{\{e\prime\prime=(x'',x)\in E\}} f(e'')$
  (**Flusserhalt**)

Beispiel: Fluss-Netzwerk



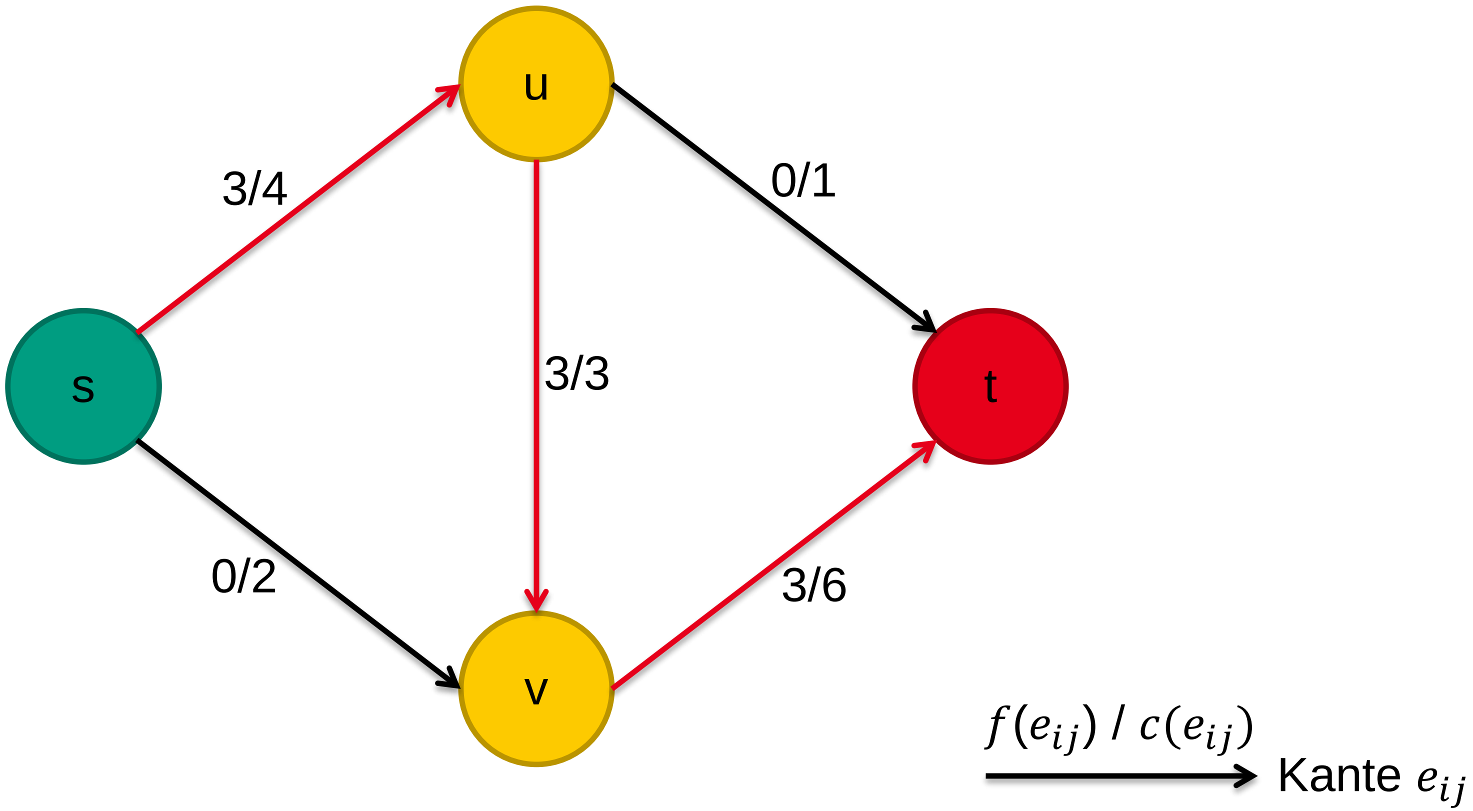
Maximaler Fluss in Flussgraphen

- Darstellung eines Flusses in einem Fluss-Netzwerk
- Mit e_{ij} bezeichnen wir jeweils Kanten von Knoten i nach Knoten j



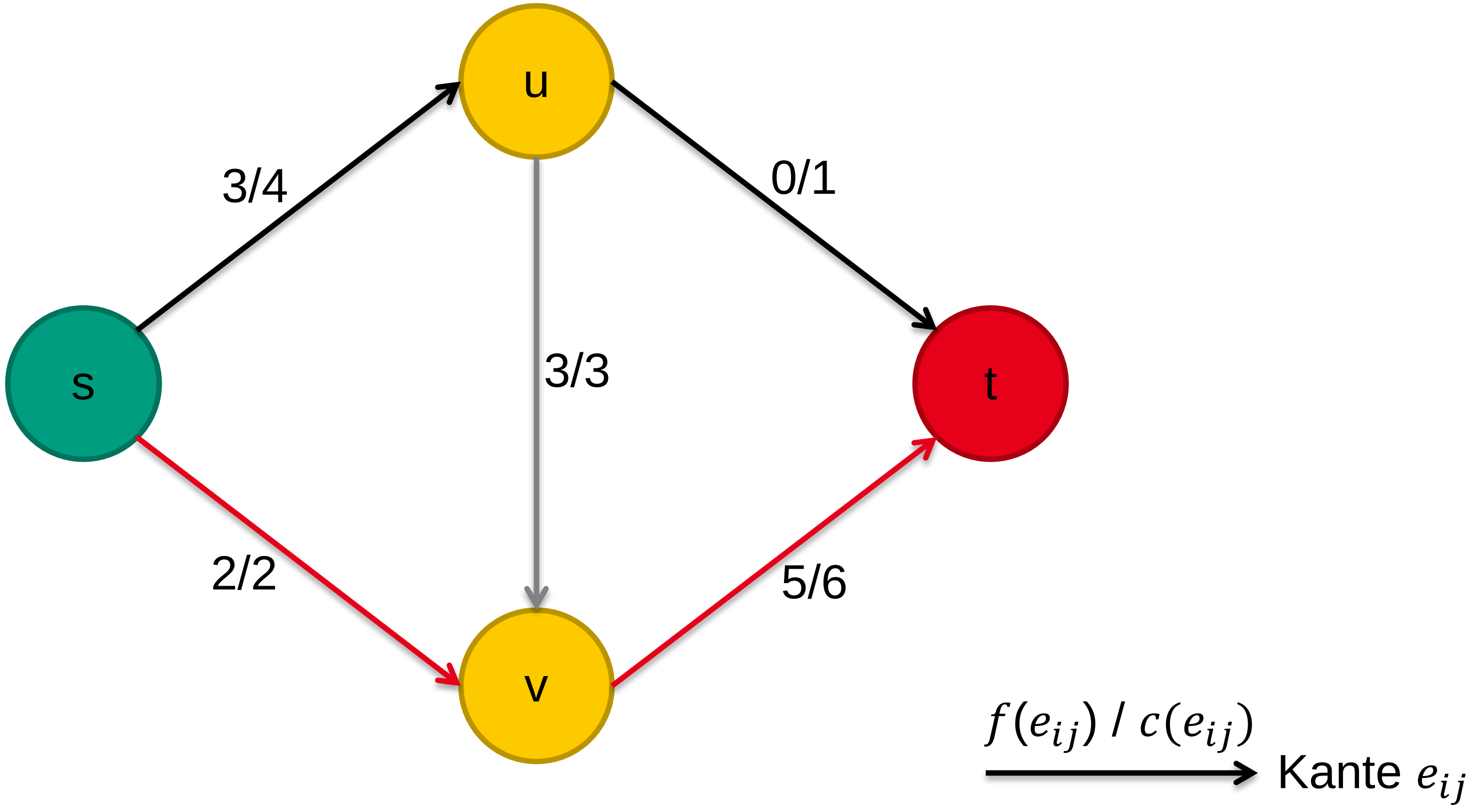
Maximaler Fluss in Flussgraphen

- Rote Kanten repräsentieren eine Flusserhöhung an diesen Kanten
- Die Flusserhöhung erfolgt auf einem Pfad von Knoten s zu Knoten t

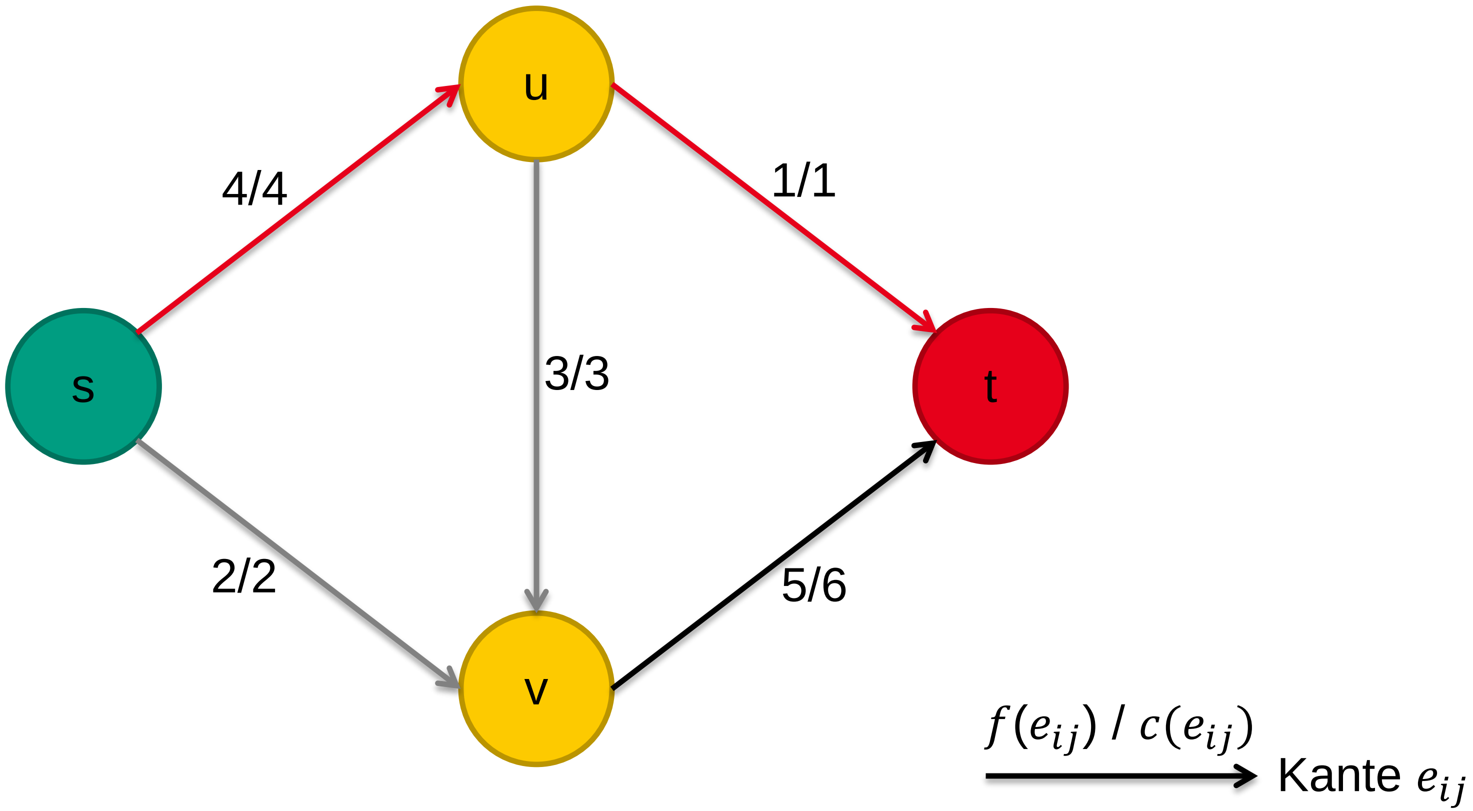


Maximaler Fluss in Flussgraphen

- Graue Kanten zeigen an, dass keine weitere Flusserhöhung entlang der Kante mehr möglich ist.

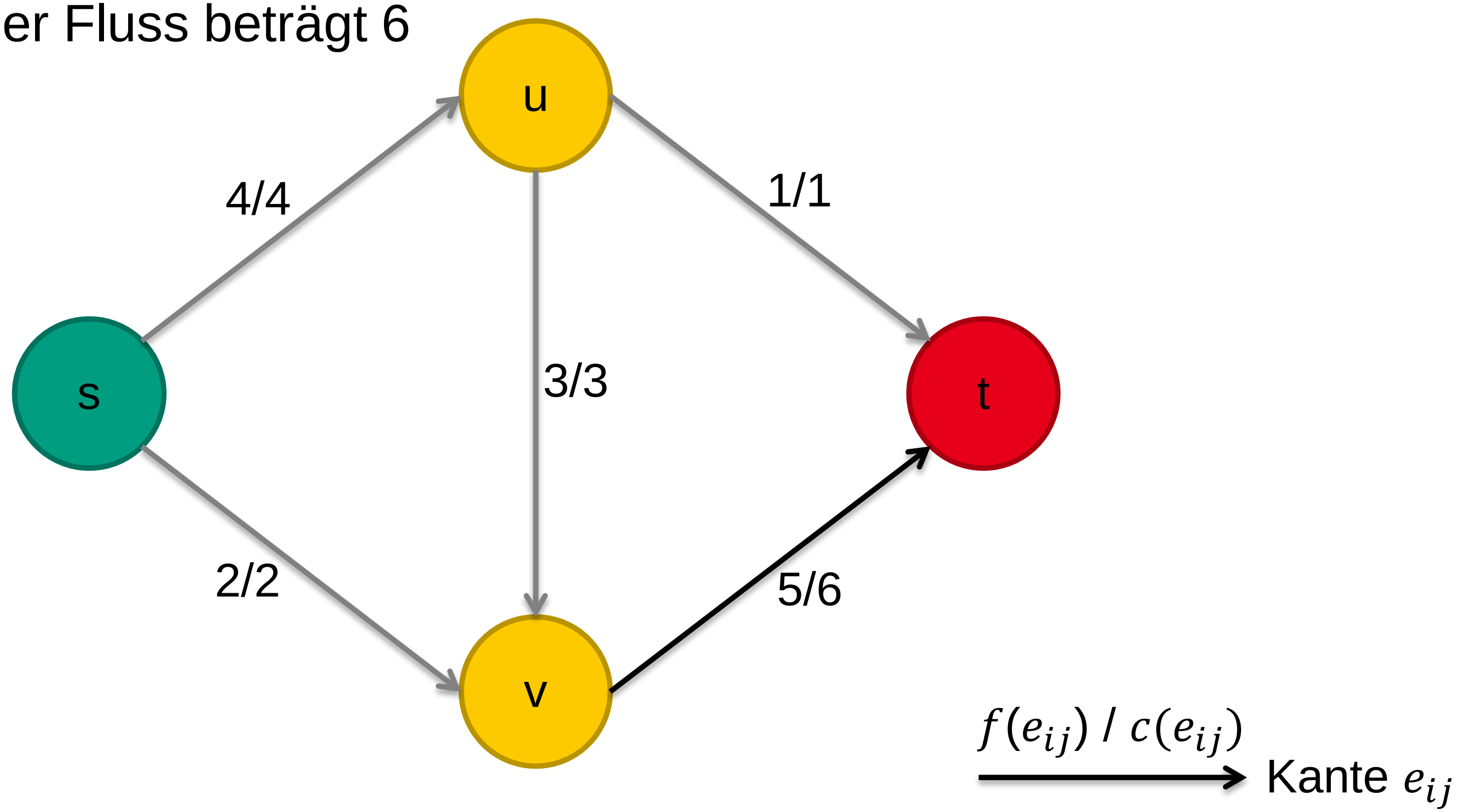


Maximaler Fluss in Flussgraphen



Maximaler Fluss in Flussgraphen

- Der maximale Fluss ist erreicht, wenn kein Pfad von Knoten s nach Knoten t eine Restkapazität besitzt
- Maximaler Fluss beträgt 6



Maximaler Fluss in Flussgraphen

Für den maximalen Fluss zwischen einem Start- und einem Zielknoten gilt:

Der maximale Fluss in einem Flussgraphen ist durch die niedrigste Kanten-kapazität entlang der möglichen Wege vom Start- zum Zielknoten beschränkt.

Gliederung

- Definition und Eigenschaften von Graphen
- Flüsse in Netzwerken
- Residualnetzwerke
- Aufgabenblock 3

Residualgraph

- Der im Praktikum zur Berechnung des maximalen Flusses verwendete *Ford-Fulkerson-Algorithmus* funktioniert in Kombination mit einem **Residualgraph**

- Die **ungerichteten** Flussgraphen des Energienetzes müssen dafür zunächst in **gerichtete (symmetrische)** Flussgraphen überführt werden

- Für den gerichteten symmetrischen Flussgraphen gilt:
  - Eine ungerichtete Kante wird dargestellt durch zwei gegenläufige Kanten mit der jeweils gleichen Kapazität
  - nur eine der beiden Kanten besitzt einen Fluss größer Null

Definition: Residualgraphen

- Ein **Residualgraph** $G_f = (V, E_f, c_f)$ beschreibt mögliche verbleibende Flusswege durch einen Flussgraphen $G = (V, E, c)$ mit dem bisherigen Fluss f

  - V ist die Knotenmenge von G
  - E_f enthält diejenigen Kanten von G, deren Kapazität noch nicht durch den Fluss f ausgelastet sind, mit ihrer **Restkapazität** (Residualkapazität):

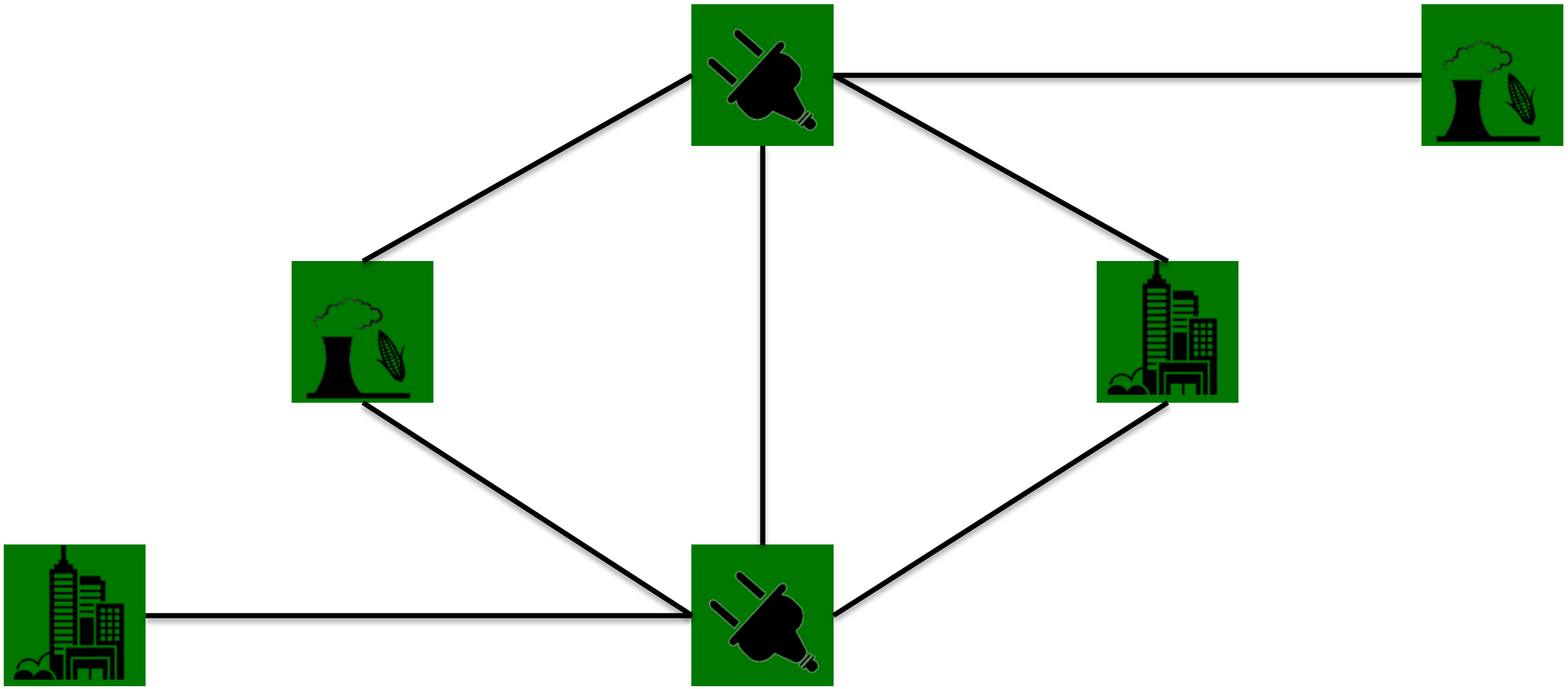
$$\text{Falls } f(e_{ij}) < c(e_{ij}), \text{ dann } e_{ij}^{r} \in E_f \text{ mit } c_f(e_{ij}^{r}) = c(e_{ij}) - f(e_{ij})$$

  - sowie zusätzliche **Rückwärtskanten,** deren Kapazität dem Fluss auf der zugehörigen Vorwärtskante entspricht:

$$\text{Falls } f(e_{ij}) > 0, \text{ dann } e_{ji}^{r\prime} = (j, i) \in E_f \text{ mit } c_f(e_{ji}^{r\prime}) = f(e_{ij})$$

Maximale Flüsse im EVS-Szenario

- Es gibt **mehrere Quellen und Senken** (siehe AB4)
- Im EVS-Szenario sind Leitungen **ungerichtet**

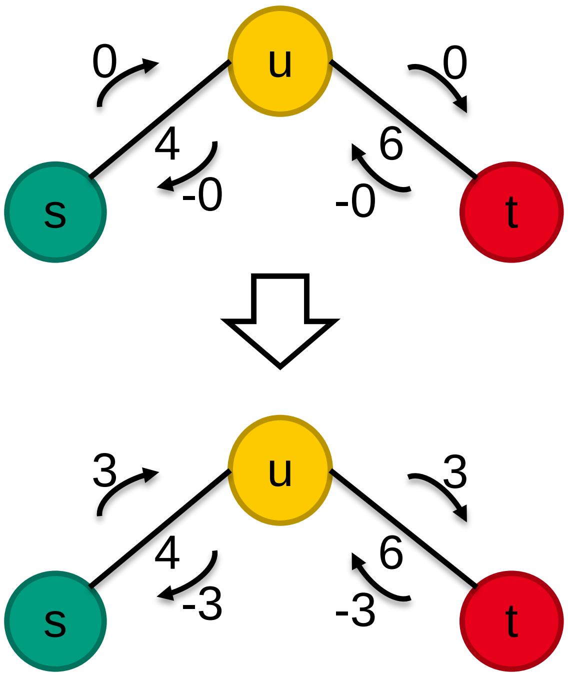


Umwandlung: ungerichteter Flussgraph in gerichteten symmetrischen Flussgraph

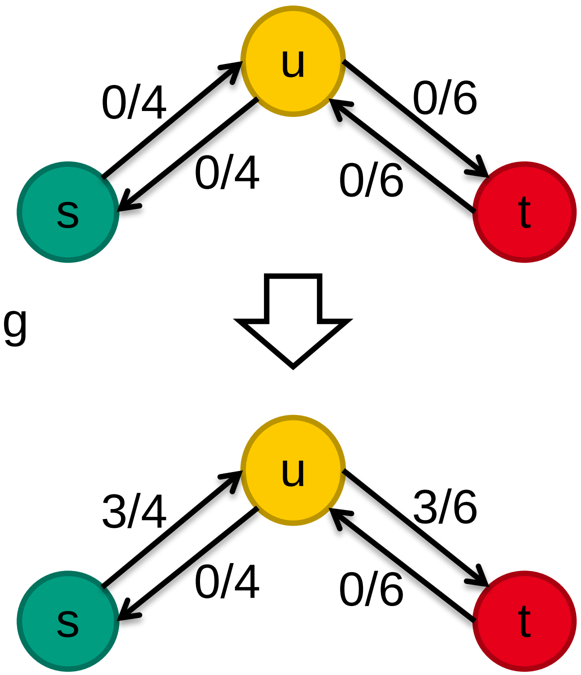
- Ungerichtete Kanten im ungerichteten Flussgraphen werden im gerichteten symmetrischen Flussgraphen als Paare gegenläufiger Kanten mit gleicher Kapazität dargestellt
- Flusserhöhungen in ungerichteten Flussgraphen sind immer symmetrisch: Eine Erhöhung des Flusses auf einer Kante entspricht einer gleichgroßen Verringerung des Flusses auf der zugehörigen rücklaufenden Kante
- Flüsse können in ungerichteten Flussgraphen nicht repräsentiert werden, daher sind diese im Folgenden als zusätzliche Pfeile angedeutet, deren Kantenbeschriftung dem Fluss entspricht

Umwandlung: ungerichteter Flussgraph in gerichteten symmetrischen Flussgraph

- ungerichteter Flussgraph
- gerichteter symmetrischer Flussgraph



Flusserhöhung



$c(e_{ij})$ $f(e_{ij})$

$f(e_{ij})$ / $c(e_{ij})$ $\longrightarrow$ Kante e_{ij}

Umwandlung: gerichteter symmetrischer Flussgraph in symmetrischen Residualgraph

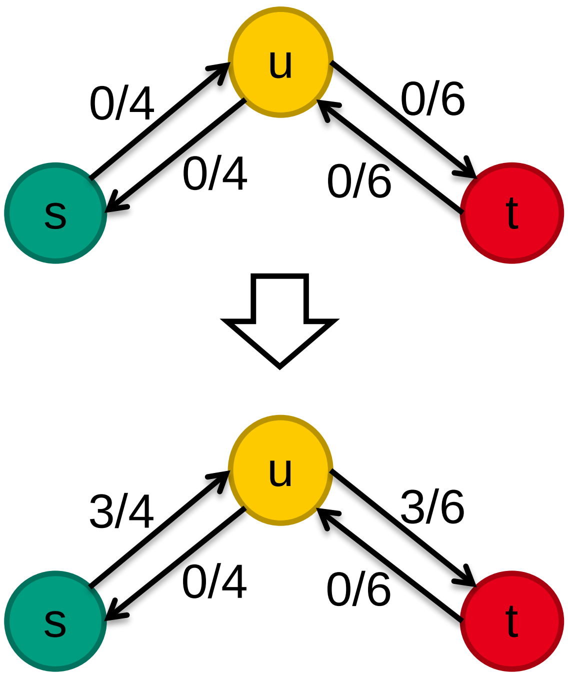
- Um die Anwendung des Ford-Fulkerson Algorithmus zu ermöglichen, müssen die gerichteten symmetrischen Flussgraphen in symmetrische Residualgraphen umgewandelt werden

- Kanten des gerichteten symmetrischen Flussgraphen werden im symmetrischen Residualgraphen durch eine Vorwärtskante mit verbliebener Restkapazität $c_f(e_{ij}^r)$ und eine Rückwärtskante $c_f(e_{ij}^{r\prime})$ dargestellt

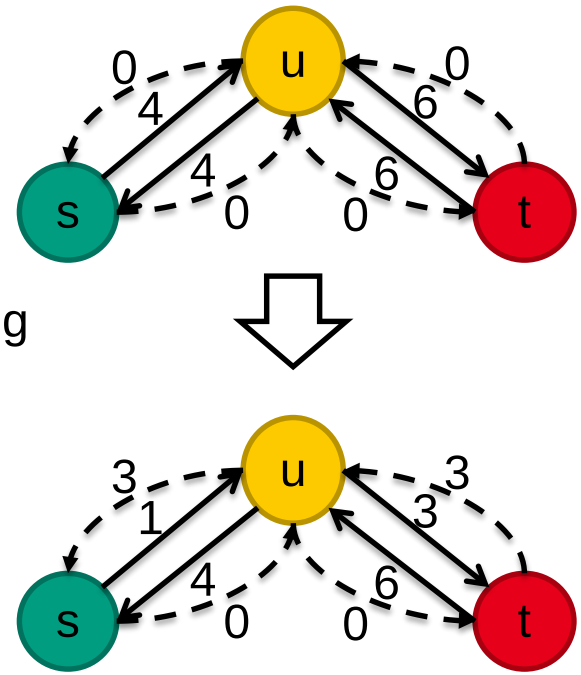
- Die Kapazität der Rückwärtskante entspricht dem Fluss entlang der Kante im Flussgraphen $c_f\left(e_{ij}^{r\prime}\right) = f(e_{ij})$ und die Kapazität der Vorwärtskante $c_f(e_{ij}^r) = c\left(e_{ij}\right) - f(e_{ij})$ der Differenz von Kapazität und Fluss der Kante im Flussgraph

Umwandlung: gerichteter symmetrischer Flussgraph in symmetrischen Residualgraph

- gerichteter symmetrischer Flussgraph
- symmetrischer Residualgraph



Flusserhöhung



$f(e_{ij})$ / $c(e_{ij})$

Kante e_{ij}

$c_f(e^{r}_{ij})$

$c_f(e^{r\prime}_{ij})$

Umwandlung: gerichteter symmetrischer Flussgraph in vereinfachten symmetrischen Residualgraph

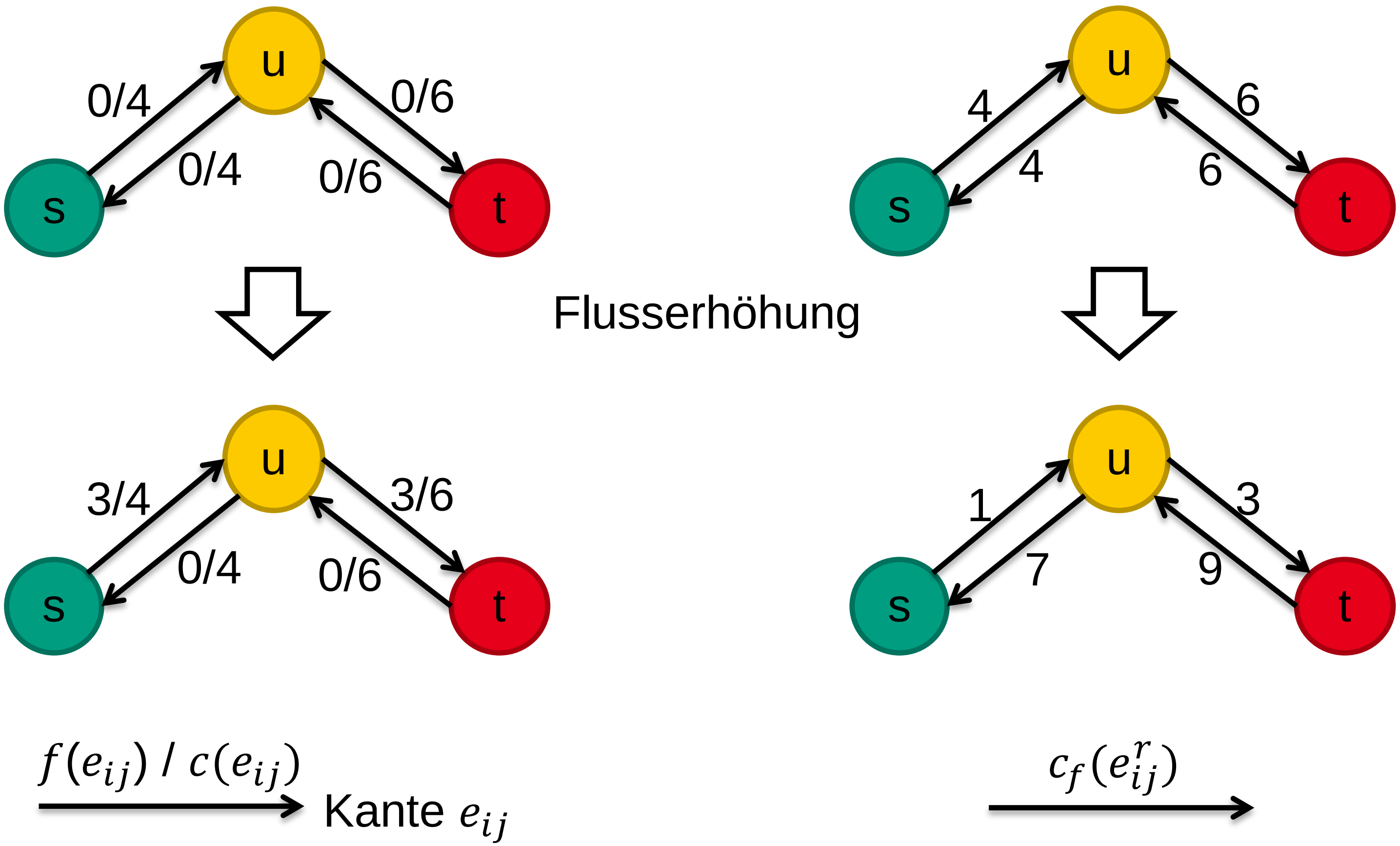
- Im vorliegenden Fall gerichteter symmetrischer Flussgraphen können mithilfe von zwei Kanten im Residualgraphen alle notwendigen Informationen repräsentiert werden

- Kanten des gerichteten symmetrischen Flussgraphen werden im vereinfachten symmetrischen Residualgraphen jeweils durch eine Kapazität dargestellt, wobei $\mathrm{c}(e_{ij}) = [c_f(e_{ij}^r) + c_f(e_{ji}^r)]/2$ gilt

- Die Kapazitäten der Kanten des Residualgraphen bestimmen sich durch: $c_f(e_{ij}^r) = if\ f(e_{ij}) > 0\ then\ c(e_{ij}) - f(e_{ij})\ else\ c(e_{ij}) + f(e_{ji})$

Umwandlung: gerichteter symmetrischer Flussgraph in vereinfachten symmetrischen Residualgraph

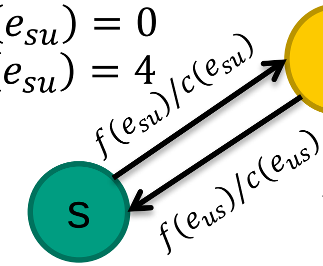
- gerichteter symmetrischer Flussgraph
- vereinfachter symmetrischer Residualgraph



Umwandlung: gerichteter symmetrischer Flussgraph in vereinfachten symmetrischen Residualgraph

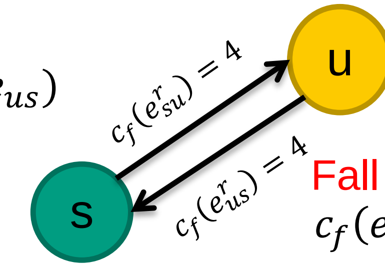
- gerichteter symmetrischer Flussgraph
- vereinfachter symmetrischer Residualgraph

$f(e_{su}) = 0$
$c(e_{su}) = 4$



$f(e_{us}) = 0$
$c(e_{us}) = 4$

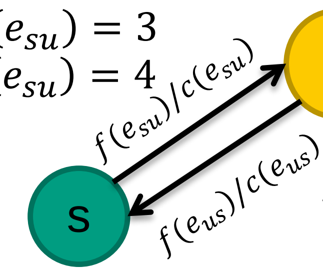
Fall 1: $f(e_{su}) = 0$
$c_f(e_{su}^r) = c(e_{su}) + f(e_{us})$



Fall 1: $f(e_{us}) = 0$
$c_f(e_{us}^r) = c(e_{us}) + f(e_{su})$

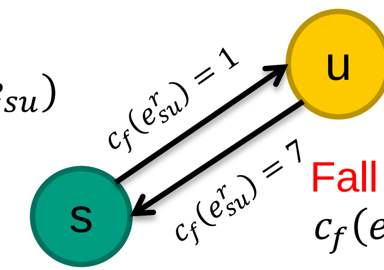
Flusserhöhung

$f(e_{su}) = 3$
$c(e_{su}) = 4$



$f(e_{us}) = 0$
$c(e_{us}) = 4$

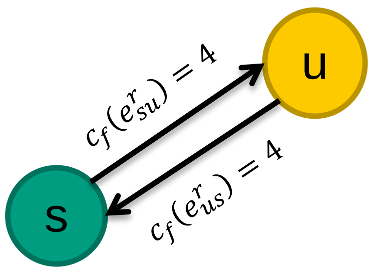
Fall 2: $f(e_{su}) > 0$
$c_f(e_{su}^r) = c(e_{su}) - f(e_{su})$



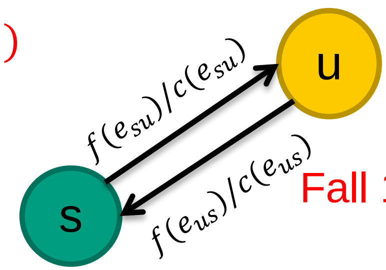
Fall 1: $f(e_{us}) = 0$
$c_f(e_{us}^r) = c(e_{us}) + f(e_{su})$

Umwandlung: vereinfachter symmetrischer Residualgraph in gerichteten symmetrischen Flussgraph

- vereinfachter symmetrischer Residualgraph
- gerichteter symmetrischer Flussgraph

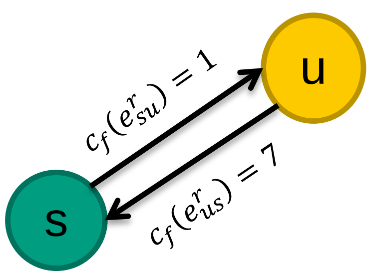


Fall 1: $c_f(e^r_{su}) \geq c_f(e^r_{us})$
$f(e_{su}) = 0$

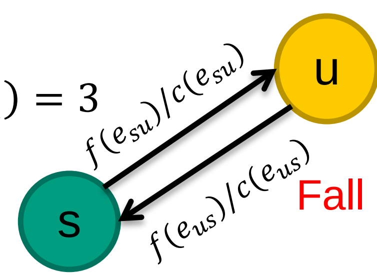


Fall 1: $c_f(e^r_{us}) \geq c_f(e^r_{su})$
$f(e_{us}) = 0$

Flusserhöhung



Fall 2: $c_f(e^r_{su}) < c_f(e^r_{us})$
$f(e_{su}) = c_f(e^r_{us}) - c(e_{su}) = 3$



Fall 1: $c_f(e^r_{us}) \geq c_f(e^r_{su})$
$f(e_{us}) = 0$

Für alle Kapazitäten gilt:
$$c(e_{ij}) = [c_f(e^r_{ij}) + c_f(e^r_{ji})]/2$$

Zusammenfassung der Umwandlung

- Hierbei ist e_{ij} die Kante im Flussgraphen von i nach j und e_{ij}^r die Kante im Residualgraphen von i nach j

- Berechnung der Kantenkapazitäten des Residualgraphen aus dem Flussgraphen:
  - $c_f(e_{ij}^r) = if\ f(e_{ij}) > 0\ then\ c(e_{ij}) - f(e_{ij}) else\ c(e_{ij}) + f(e_{ji})$

- Berechnung der Flüsse und Kapazitäten des Flussgraphen aus dem Residualgraphen:
  - $f(e_{ij}) = if\ c_f(e_{ij}^r) < c_f(e_{ji}^r)\ then\ c_f(e_{ji}^r) - c(e_{ij})\ else\ 0$
  - $c(e_{ij}) = [c_f(e_{ij}^r) + c_f(e_{ji}^r)]/2$

Maximaler Fluss in gerichteten symmetrischen Flussgraphen und vereinfachten symmetrischen Residualgraphen

- Im folgenden Beispiel wird der maximale Fluss in einem gerichteten symmetrischen Flussgraphen und dem äquivalenten vereinfachten symmetrischen Residualgraphen bestimmt

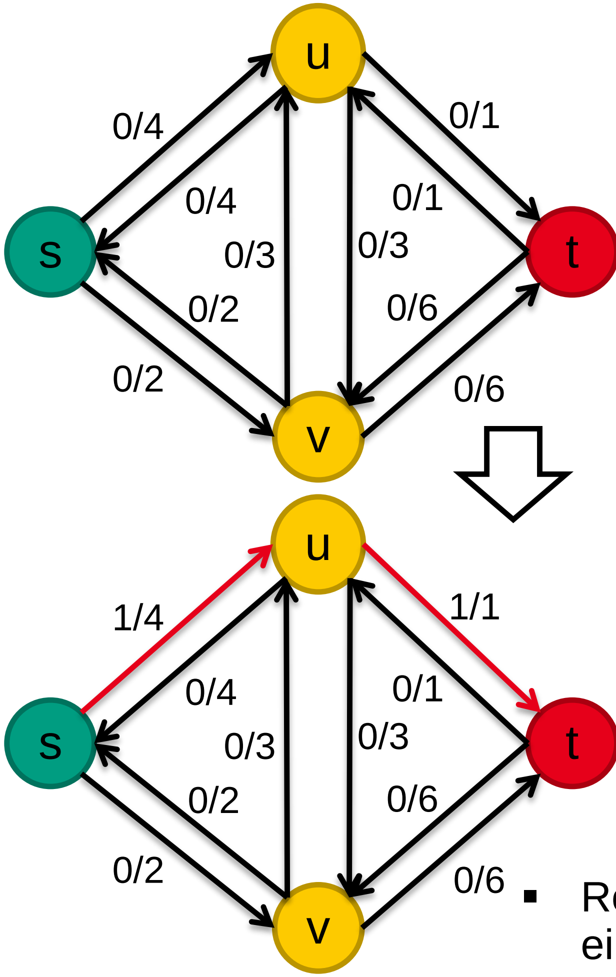
- Ausgehend vom Startknoten s wird dabei der Fluss entlang der Kanten mit verbliebener Restkapazität immer weiter erhöht, bis keine weitere Flusserhöhung möglich ist

- Die Berechnung der Kapazitäten und Flüsse der Kanten erfolgt nach den zuvor genannten Formeln

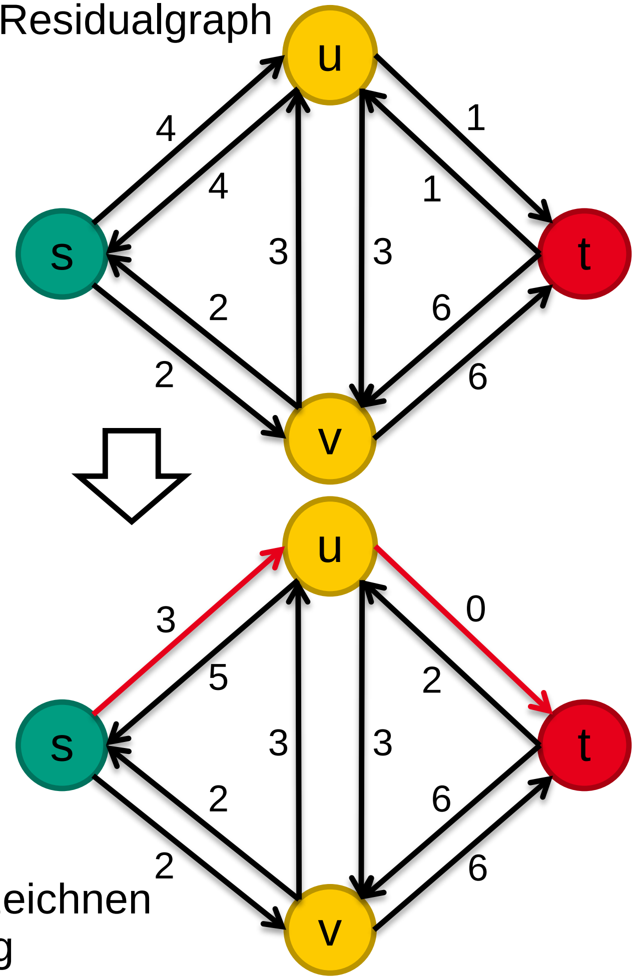
Maximaler Fluss in gerichteten symmetrischen Flussgraphen und vereinfachten symmetrischen Residualgraphen

TECHNISCHE UNIVERSITÄT DARMSTADT

- gerichteter symmetrischer Flussgraph
- vereinfachter symmetrischer Residualgraph



Flusserhöhung

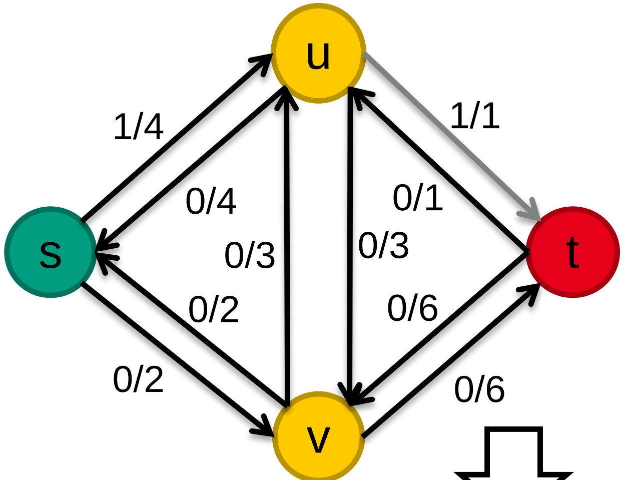


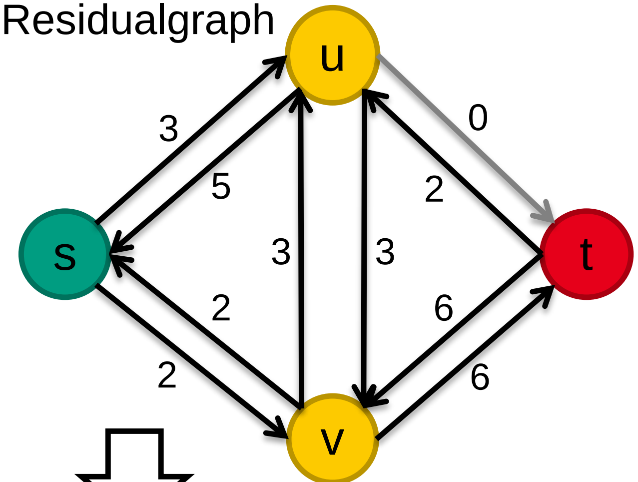
- Rote Kanten kennzeichnen eine Flusserhöhung

Maximaler Fluss in gerichteten symmetrischen Flussgraphen und vereinfachten symmetrischen Residualgraphen

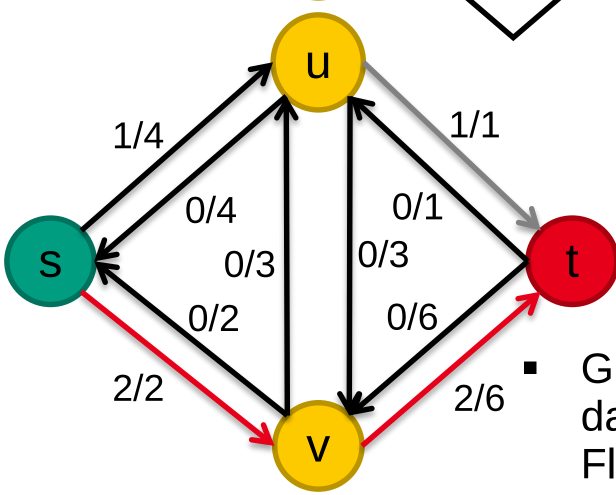
TECHNISCHE UNIVERSITÄT DARMSTADT

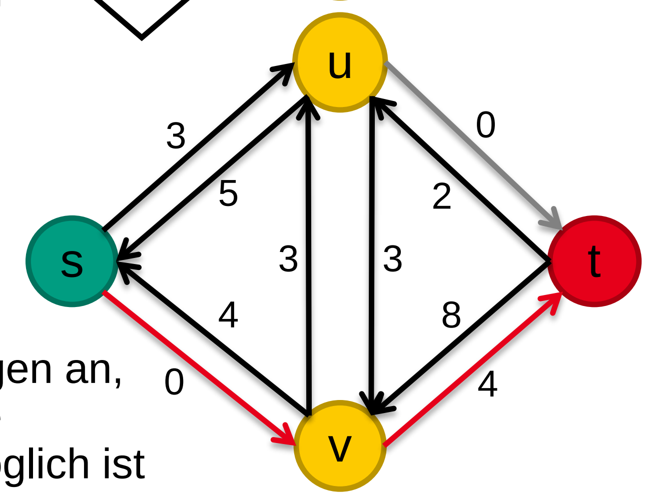
- gerichteter symmetrischer Flussgraph
- vereinfachter symmetrischer Residualgraph





Flusserhöhung



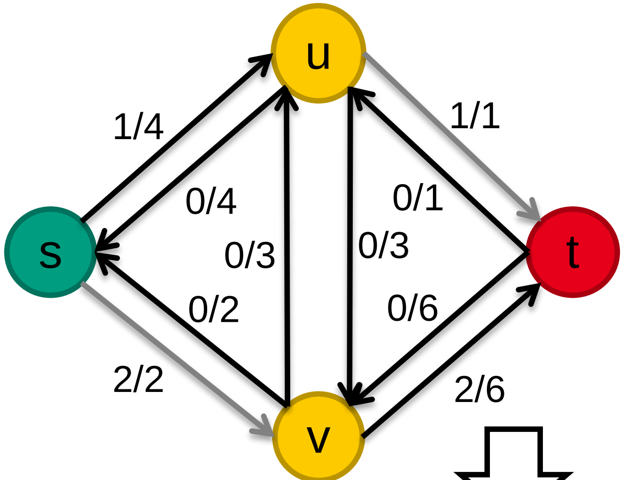


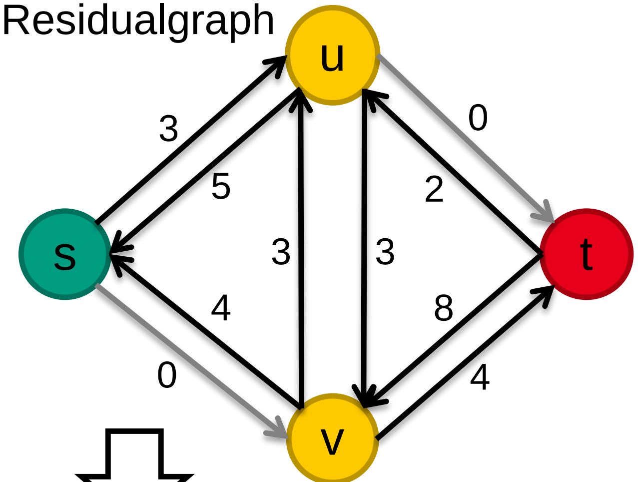
- Graue Kanten zeigen an, dass keine weitere Flusserhöhung möglich ist

Maximaler Fluss in gerichteten symmetrischen Flussgraphen und vereinfachten symmetrischen Residualgraphen

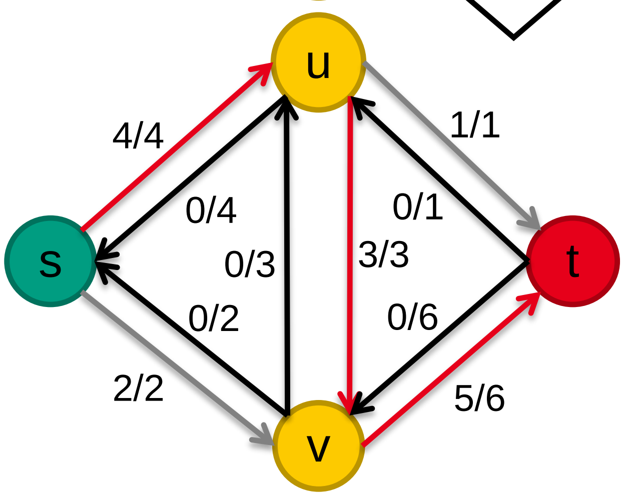
TECHNISCHE UNIVERSITÄT DARMSTADT

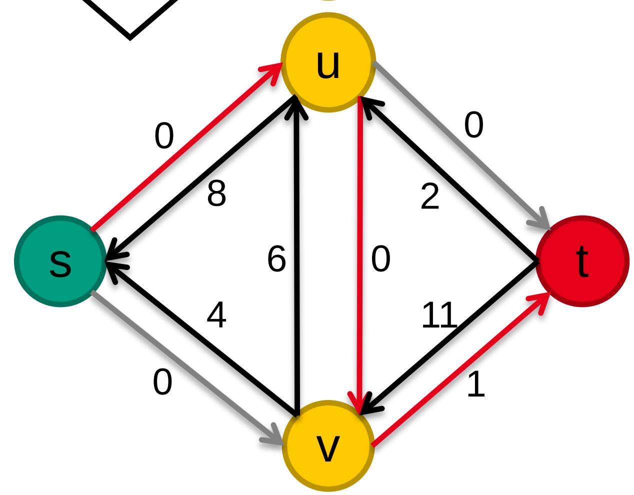
- gerichteter symmetrischer Flussgraph
- vereinfachter symmetrischer Residualgraph





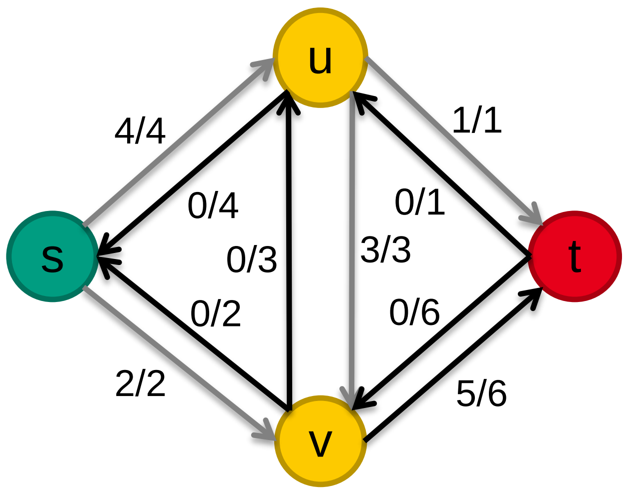
Flusserhöhung



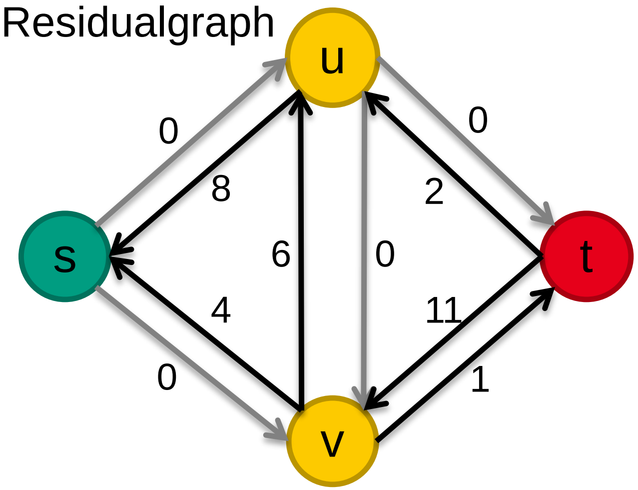


Maximaler Fluss in gerichteten symmetrischen Flussgraphen und vereinfachten symmetrischen Residualgraphen

- gerichteter symmetrischer Flussgraph



- vereinfachter symmetrischer Residualgraph



- Es ist keine weitere Flusserhöhung möglich, da alle vom Startknoten s abgehenden Kanten ihre Kapazität voll ausgeschöpft haben

- Der maximale Fluss von 6 ist damit erreicht

Gliederung

- Definition und Eigenschaften von Graphen
- Flüsse in Netzwerken
- Residualnetzwerke
- Aufgabenblock 3

Aufgabenblock 3

3.1. Generische Klasse **`FlowEdge`** implementieren

3.2. Generische Klasse **`FlowGraph`** implementieren

3.3. Testen von **`FlowGraph`**

3.4. Generische Klasse **`ResidualEdge`** implementieren

3.5. Generische Klasse **`ResidualGraph`** implementieren