

Algorithmen & Wahrscheinlichkeit

Übungsstunde 12

Minitest

- Passwort: cut
- Zeit: 10 minutes
- Anzahl Fragen: 10
- 0.2 Punkte pro richtig beantwortete Frage

Programm

- Minitest
- Theory Recap
- Aufgaben

Minitest 6

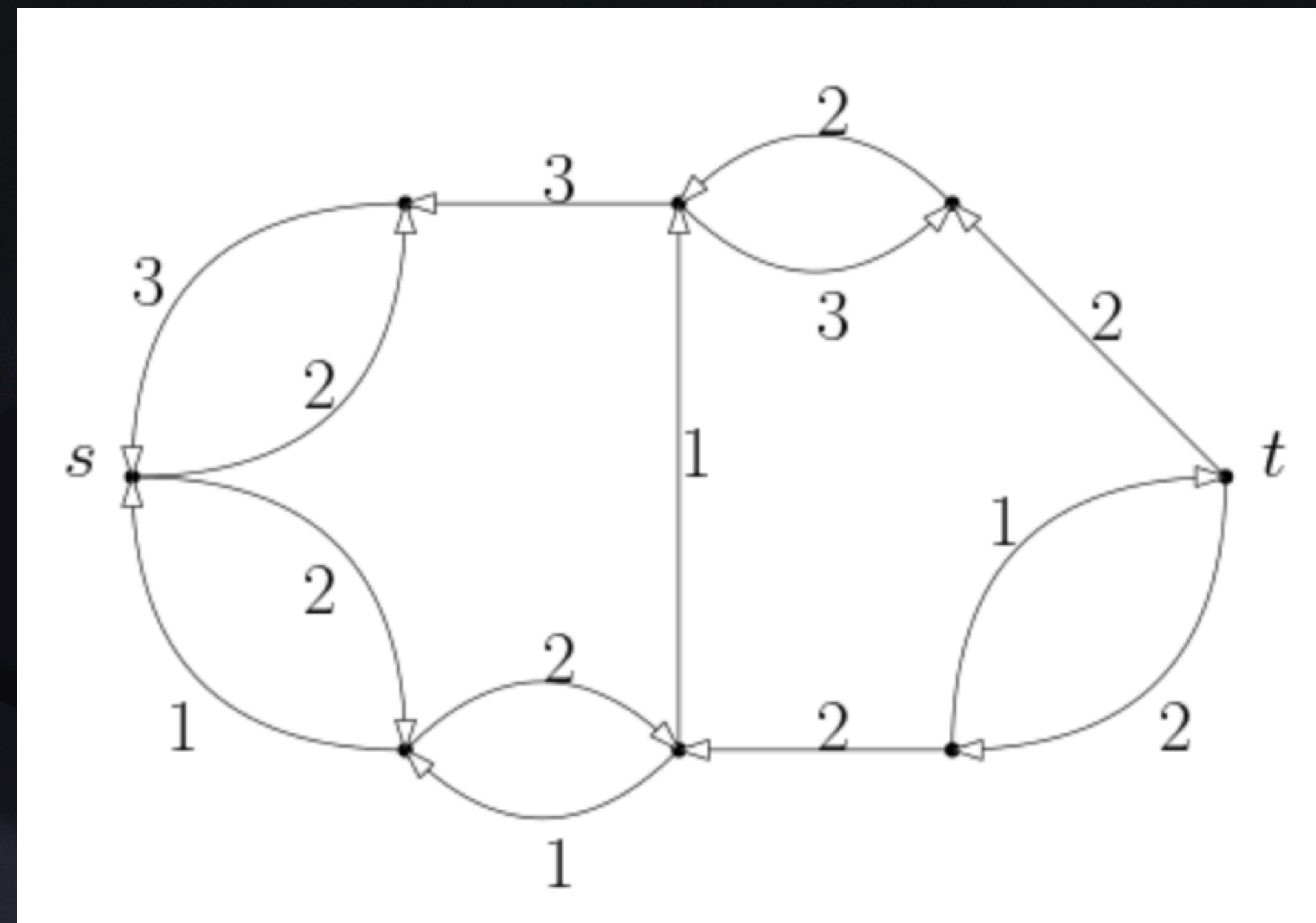
Teil 1

- Für jeden 2-färbbaren Graph G können wir mithilfe eines Flussproblems herausfinden ob G ein perfektes Matching hat.
- Sei $N = (V, A, c, s, t)$ ein Netzwerk. Wenn c nur ganzzahlige Werte annimmt, so ist jeder maximale Fluss in N ganzzahlig.
- Sei $N = (V, A, c, s, t)$ ein Netzwerk ohne engeren gerichtete Kanten und f ein Fluss, so dass $\text{val}(f) > 0$. Sei N_f das Restnetzwerk. Dann enthält N_f einen (gerichteten) Weg von t nach s .

Mintest 6

Teil 2

- Sei $N = (V, A, c, s, t)$ ein Netzwerk ohne entgegen gerichtete Kanten und f ein Fluss. Das Restnetzwerk N_f ist wie folgt gegeben, ist der Fluss maximal?



Minitest 6

Teil 3

- Sei $N = (V, A, c, s, t)$ ein Netzwerk ohne engeren gerichtete Kanten. Sei zudem die Kapazität jeder Kante höchstens U . Dann berechnet der Ford-Fulkerson Algorithmus in Zeit $O(mnU)$ einen maximalen Fluss.
- Sei $N = (V, A, c, s, t)$ ein Netzwerk, f ein Fluss in N und (S, T) ein s - t -Schnitt in N . Dann gilt $\text{val}(f) \geq \text{cap}(S, T)$
- Sei f ein maximaler Fluss in einem Netzwerk $N = (V, A, c, s, t)$ ein Netzwerk ohne engeren gerichtete Kanten. Dann gibt es keine zwei Knoten u, v in V , sodass sowohl die Kante (u, v) als auch die Kante (v, u) im Restnetzwerk N_f vorkommt.

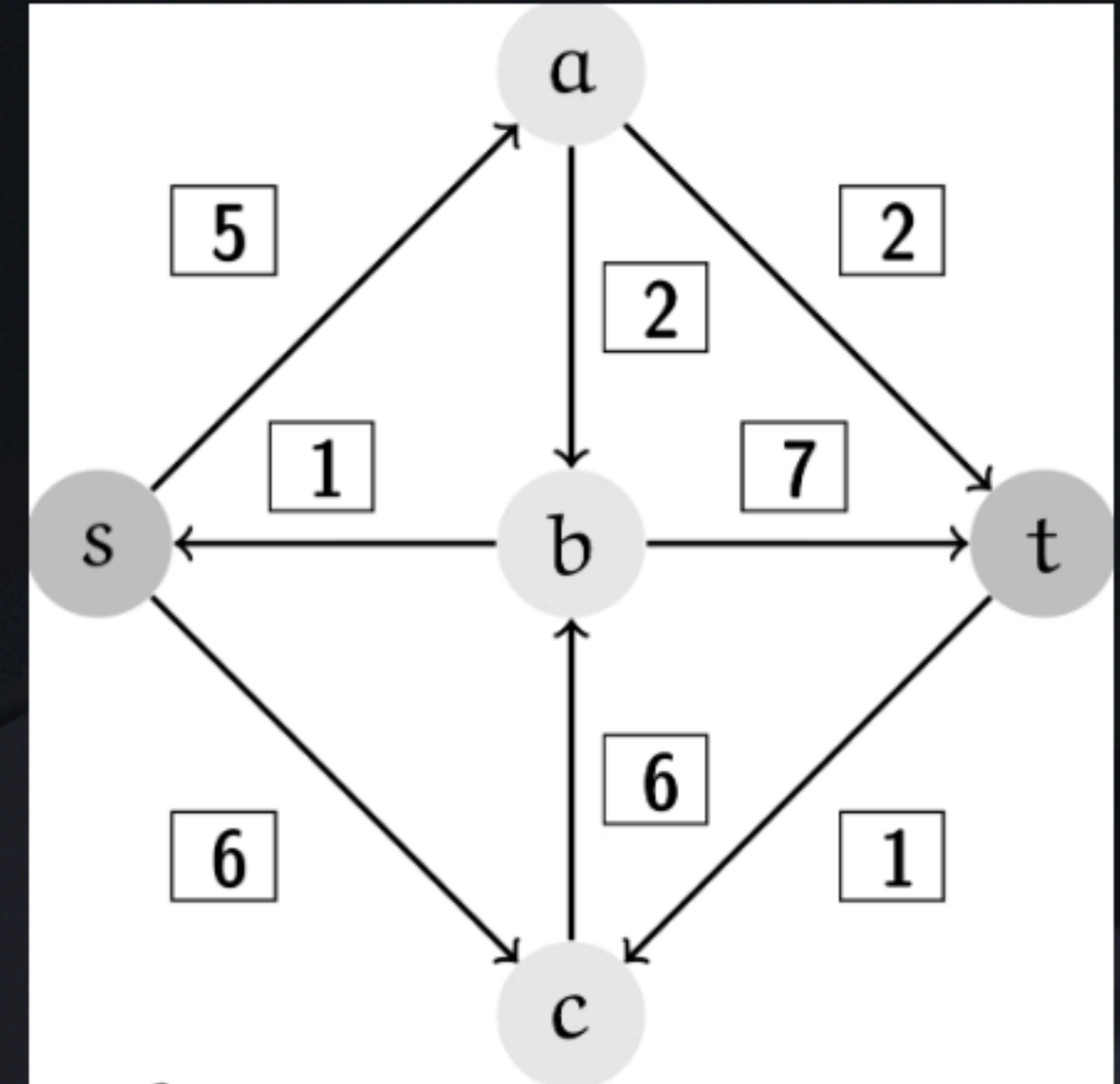
Mintest 6

Teil 4

- Das MIN-CUT-Problem für einen gegebenen Multigraphen (V, E) kann gelöst werden, indem man einen Knoten s in V fest wählt und minimale s - t -Schnitte für alle t in $V \setminus \{s\}$ berechnet.
- Wenn e eine Kante eines Multigraphen G ist, dann gilt immer $\mu(G/e) \leq \mu(G)$, wobei $\mu(G)$ die Kardinalität eines minimalen Schnitts in G bezeichnet.
- Sei G ein Multigraph. Wenn es einen minimalen Schnitt C von G mit e nicht in C gibt, dann gilt $\mu(G/e) = \mu(G)$, wobei $\mu(G)$ die Kardinalität eines minimalen Schnitts in G bezeichnet.

Flüsse in Netzwerken

- Ein Netzwerk ist ein Tupel $N = (V, A, c, s, t)$, wobei gilt:
 - (V, A) ist ein gerichteter Graph
 - $s \in V$, die Quelle
 - $t \in V \setminus \{s\}$, die Senke
 - $c: A \rightarrow \mathbb{R}_0^+$, die Kapazitätsfunktion



Restnetzwerke

- Sei $N = (V, A, c, s, t)$ ein Netzwerk ohne entgegen gerichtete Kanten und sei f ein Fluss in N . Das restnetzwerk $N_f := (V, A_f, r_f, s, t)$ ist wie folgt definiert:
 - Ist $e \in A$ mit $f(e) < c(e)$, dann ist e auch eine Kante in A_f , mit $r_f(e) := c(e) - f(e)$
 - Ist $e \in A$ mit $f(e) > 0$, dann ist e^{opp} in A_f , mit $r_f(e^{\text{opp}}) = f(e)$
 - Nur oben beschriebene Kanten sind in A_f
- $r_f(e)$, $e \in A_f$, nennen wir die restkapazität der Kante e

Ford-Fulkerson

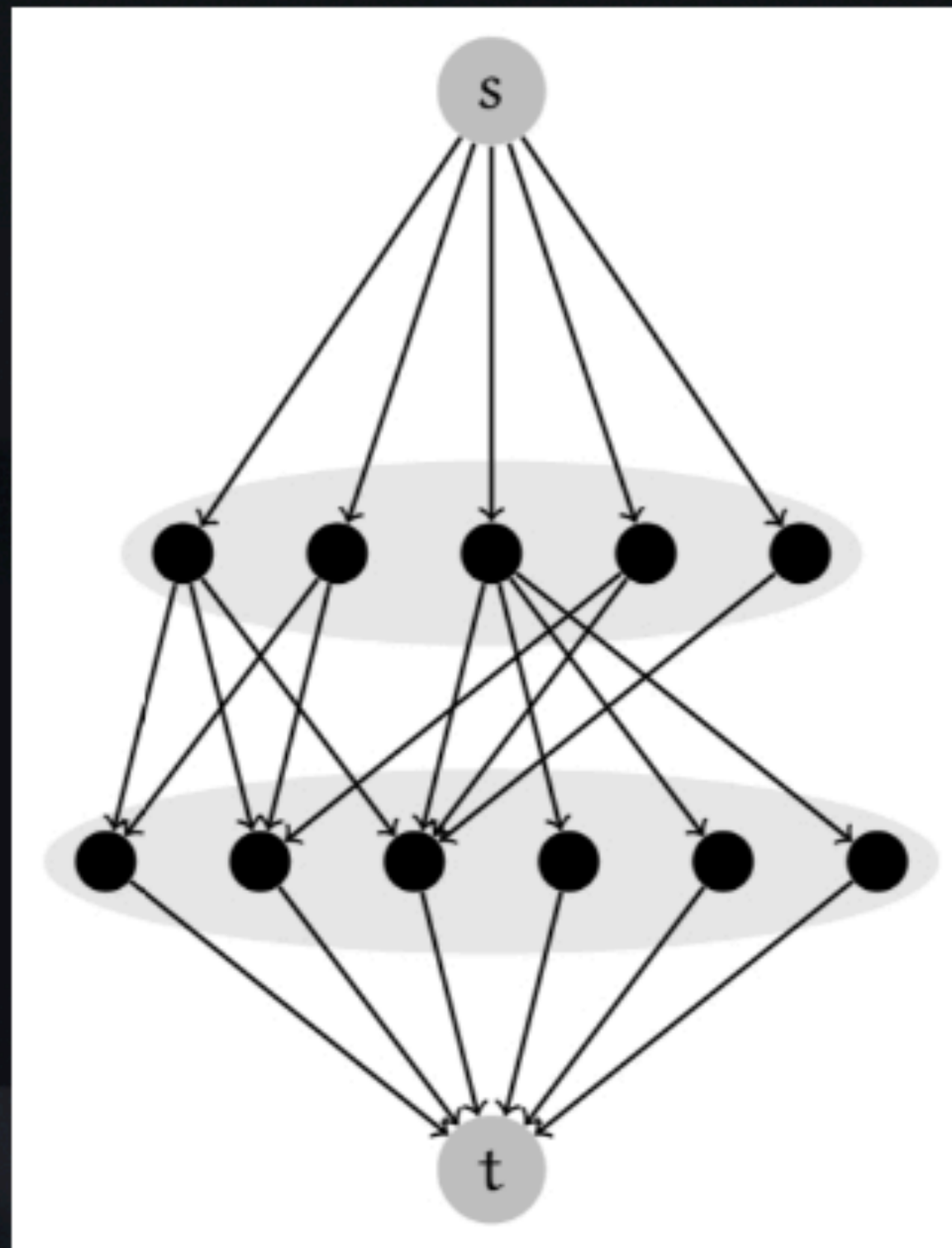
FORD-FULKERSON(V, A, c, s, t)

1: $f \leftarrow 0$	▷ Fluss konstant 0
2: while $\exists$ s-t-Pfad P in (V, A_f) do	▷ augmentierender Pfad
3: Erhöhe den Fluss entlang P	▷ wie in Beweis zu Satz 3.11
4: return f	▷ maximaler Fluss

- In ganzzahligem Netzwerk $O(mnU)$, wobei U maximales Kantenkapazität darstellt
- Capacity Scaling in $O(mn \cdot (1 + \log(U)))$
- Dynamic Trees in $(O(mn \cdot \log(n)))$

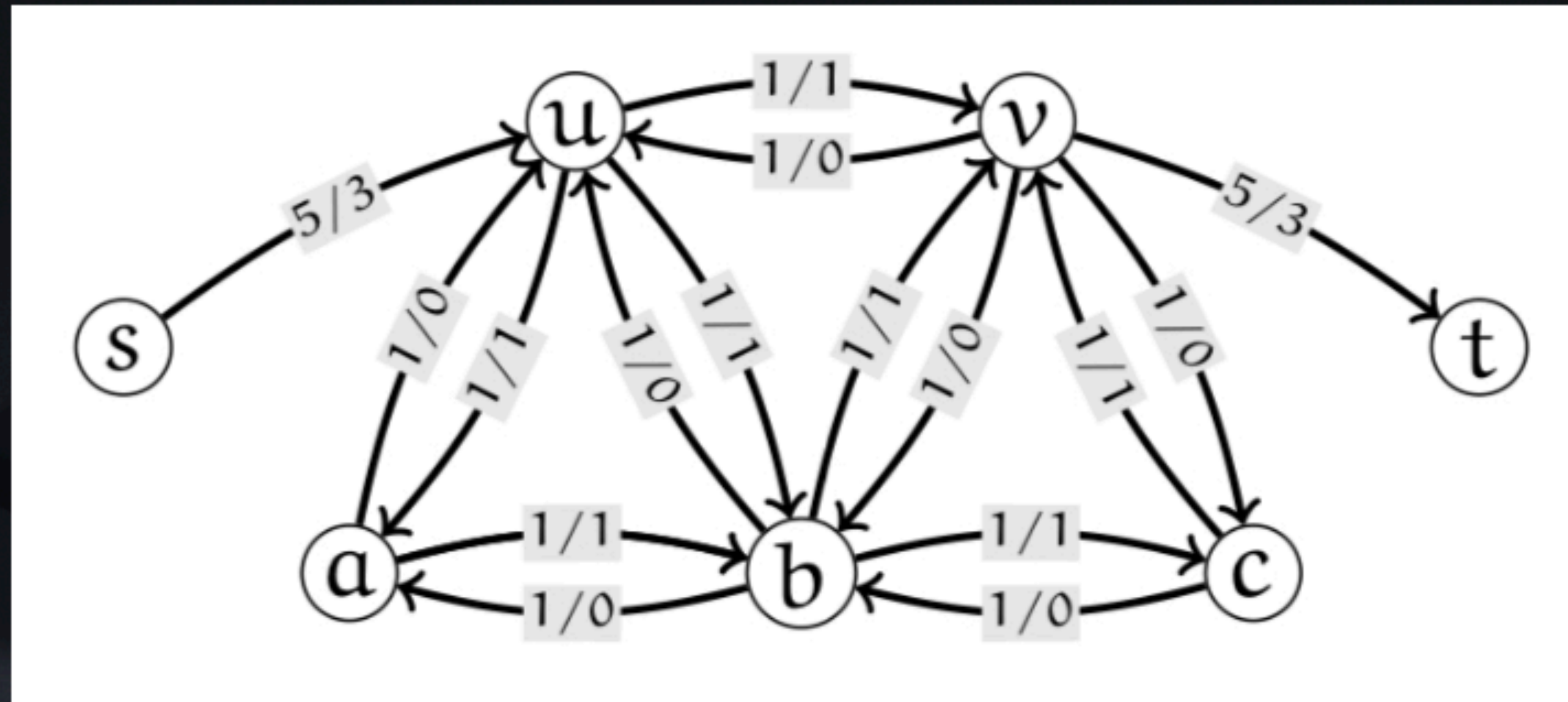
Bipartites Matching

- $A := (\{s\} \times U) \cup \{(u, w) \in U \times W \mid \{u, w\} \in E\} \cup (W \times \{t\})$



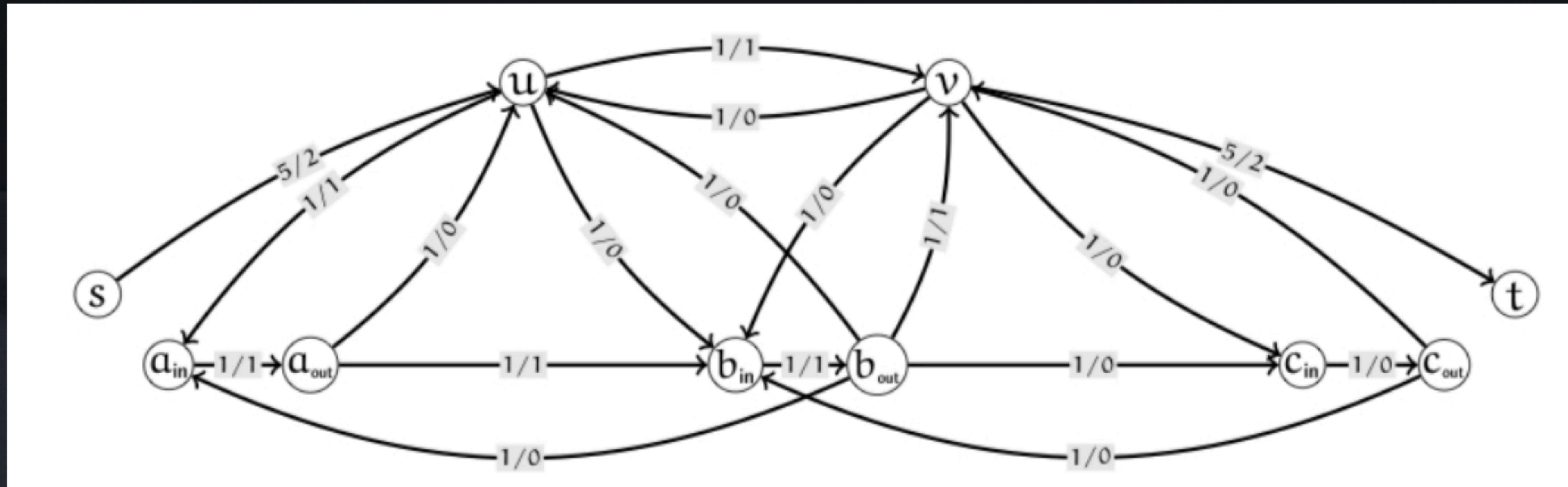
Kantendisjunkte Pfade

- Wir ersetzen jede Kante $\{x,y\}$ durch zwei gerichtete Kanten (x,y) und (y,x)
- Wir fügen s und t hinzu und verbinden diese mit u und v mit Kapazität = #Knoten



Knotendisjunkte Pfade

- Wir konstruieren das Netzwerk für Kantendisjunkte Pfade
- Wir ersetzen jeden Knoten $x \in V \setminus \{u, v\}$ durch x_{in} und x_{out} und verbinden Eingangskanten mit x_{in} und Ausgangskanten mit x_{out} und eine Kante mit Kapazität 1 zwischen x_{in} und x_{out}



Bildsegmentierung

- Problem: Gegeben ein Bild, welche Pixel sind im Vordergrund, welche im Hintergrund?
- Wir definieren:
 - $\alpha_p :=$ Erwartung p in Vordergrund
 - $\beta_p :=$ Erwartung p im Hintergrund
 - $\gamma_e :=$ Strafe, falls $e = \{u,v\}$ mit u,v nicht in gleicher Partition

Bildsegmentierung

- $q(A, B) := \sum_{p \in A} \alpha_p + \sum_{p \in B} \beta_p + \sum_{e \in E, |e \cap A|=1} \gamma_e$

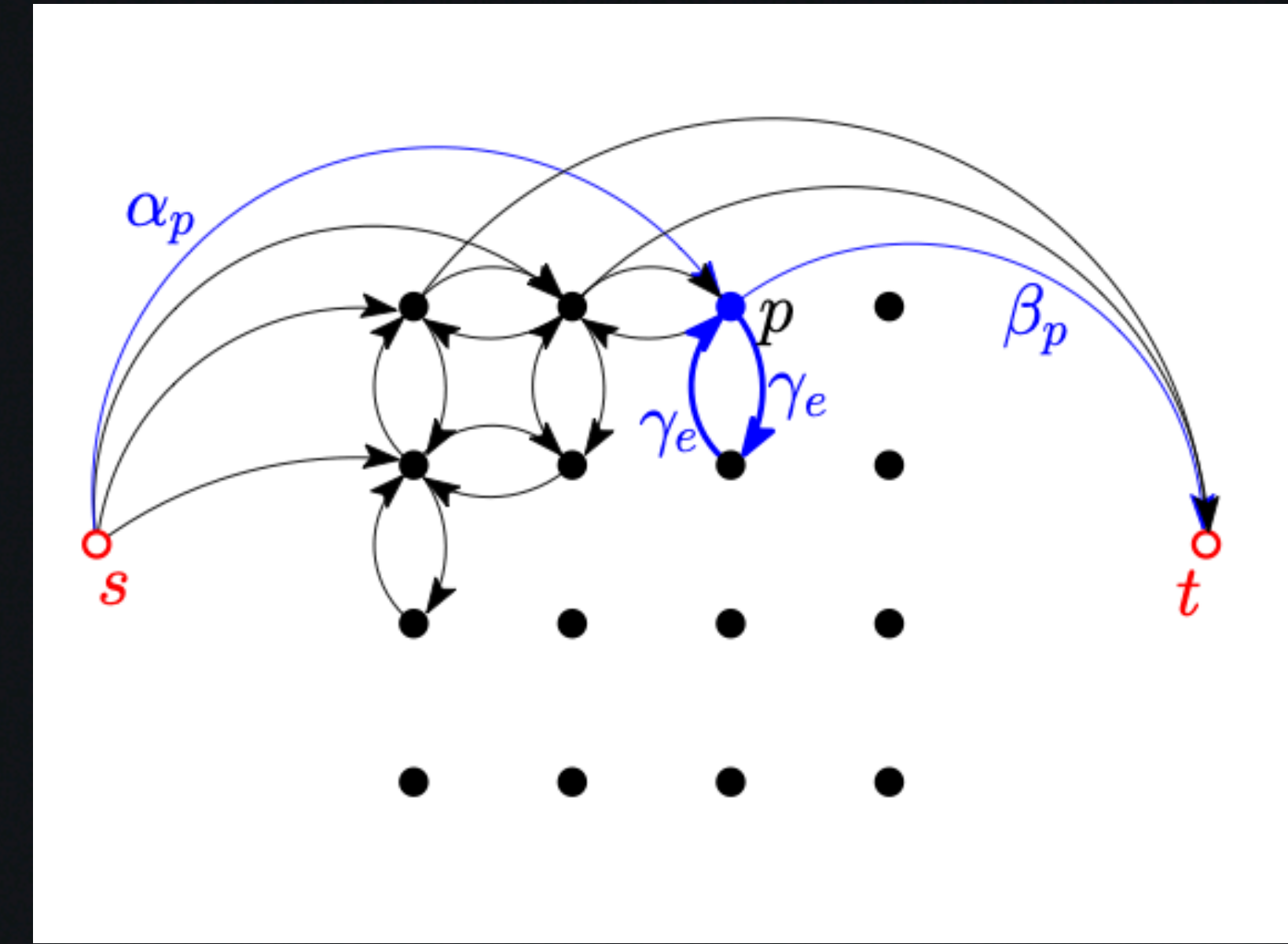
- $q(A, B) = Q - \sum_{p \in A} \alpha_p + \sum_{p \in B} \beta_p + \sum_{e \in E, |e \cap A|=1} \gamma_e$ wobei $Q := \sum_{p \in P} (\alpha_p + \beta_p)$

- $q'(A, B) := \sum_{p \in A} \beta_p + \sum_{p \in B} \alpha_p + \sum_{e \in E, |e \cap A|=1} \gamma_e$

Bildsegmentierung

Netzwerkkonstruktion

- Wir fügen s und t hinzu
- Wir verbinden s (gerichtet) mit allen Pixeln in P mit Kapazität α_p
- Wir verbinden t (gerichtet) mit allen Pixeln in P mit Kapazität β_p
- Benachbarte Pixel in P verbinden wir mit je 2 gerichteten Kanten mit je Kapazität γ_e



Bildsegmentierung

Kapazität s-t-Schnitt in N

- Kanten (s, p) mit $p \in B$. Ihr Beitrag zur Kapazität $cap(S, T)$ ist $\sum_{p \in B} \alpha_p$
- Kanten (p, t) mit $p \in A$, die $\sum_{p \in A} \beta_p$ zur Kapazität des Schnitts beitragen.
- Kanten (p, p') des Netzwerks in $A \times B$ mit Beitrag $\sum_{(p, p')} \gamma_{(p, p')}$,
wobei die Summe über alle Kanten des Netzwerks aus $A \times B$ geht.

Exercise S12.1 – *Integrality and Matching*

We have already seen Frobenius' Theorem: *Let $k \geq 1$. Every k -regular bipartite graph contains a perfect matching.* The proof we have seen uses Hall's theorem. In this exercise, we want to find another proof using our knowledge about flows.

Let $G = (A \dot{\cup} B, E)$ be a k -regular graph for $k \geq 1$.

- (a) Describe how to model bipartite matching on G as a flow problem in a network $N = (V, A, c, s, t)$.
- (b) Construct explicitly a (not necessarily integer!) flow f on N with $\text{val}(f) = n$. Conclude that $\text{maxflow}(N) = n$.

Hint: you don't have to construct an integral flow. Make sure to define the flow value on all edges of your network N !

- (c) Using the result from (b), show that there exists an integer valued flow f' with $\text{val}(f') = n$. Using f' , prove that G contains a perfect matching M .

InClass Exercise

Aufgabe 1 – Meisterschaft

Die Mannschaften aus *Bedigliora*, *Caslano*, *Novaggio*, *Pura* und *Sessa* spielen um die Meisterschaft. Während der Saison muss jede Mannschaft sechs mal gegen jede andere Mannschaft antreten, und bei jedem Spiel wird genau ein Punkt an den Sieger der Begegnung vergeben (es gibt immer einen Sieger und einen Verlierer). Die aktuelle Tabelle ist rechts angegeben. Dabei hat *Bedigliora* noch 5 ausstehende Spiele gegen *Caslano*, 3 gegen *Sessa* und 6 gegen *Novaggio*. *Caslano* hat noch 2 ausstehende Spiele gegen *Sessa* und 5 gegen *Novaggio*. *Sessa* muss noch 3 mal gegen *Novaggio* antreten.

Mannschaft	Punkte
Caslano	8
Novaggio	7
Bedigliora	6
Sessa	3
Pura	0

Pura verpflichtet mit sofortiger Wirkung einen Superstar, um doch noch die Meisterschaft zu gewinnen. Aber können sie das überhaupt noch? *Gibt es einen weiteren Saisonverlauf, bei dem Pura am Ende auf einem (womöglich geteilten) ersten Platz landet?*

Modellieren Sie das Problem als Flussproblem in einem Netzwerk.

Hinweis: Wie viele Spiele müsste *Pura* noch gewinnen, und wie viele dürfen die anderen Mannschaften noch gewinnen? Benutzen Sie diese Informationen, um die Netzwerkkapazitäten geeignet festzulegen.

InClass Exercise

Aufgabe 1 – *Arbeitsgruppen*

Zu einem Kongress werden Mitarbeiter von m Firmen gesandt. Dabei sendet die i -te Firma c_i viele Mitarbeiter. Während des Kongresses sollen die anwesenden Personen in bis zu r verschiedene Arbeitsgruppen mit jeweils höchstens 5 Mitgliedern aufgeteilt werden, wobei keine zwei Mitarbeiter derselben Firma in derselben Arbeitsgruppe sein sollen.

Wir sollen eine solche Aufteilung mit Hilfe von Fluss-Algorithmen finden.

- (a) Definieren Sie dazu ein geeignetes Netzwerk $N = (V, A, c, s, t)$ und zeigen Sie, dass es genau dann eine mögliche Aufteilung wie oben gibt, wenn $\text{maxflow}(N)$ eine gewisse (welche?) Eigenschaft hat.
- (b) Angenommen, wir erhalten als Eingabe die Zahlen $m, r, c_1, \dots, c_m$ sowie die Listen $L_1, \dots, L_m$ mit den Namen aller Teilnehmer des Kongresses. Wir können davon ausgehen, dass keine zwei Teilnehmer den gleichen Namen haben. Geben Sie einen Algorithmus an, der dies als Eingabe verwendet, und als Ausgabe entweder r Listen, $G_1, \dots, G_r$ gibt, wobei G_i die Namen aller Teilnehmer der i -ten Gruppe in einer gültigen Zuordnung enthält, oder antwortet „Zuordnung nicht möglich“.

MinCut

- Ford Fulkerson $O(nmU)$ oder $O(nm\log(n)) \rightarrow O(n^4\log(n))$
- Kantenkontraktionen $O(n^4\log(n))$ Fehlerw'keit $\leq 1/n$
- Bootstrapping $O(n^3\log(n))$
- Limit $O(n^2\text{polylog}(n))$

Minimale Schnitte in Graphen

Kantenkontraktion

- Problem: Gegeben ein Multigraph G , bestimme $\mu(G)$
- Idee: bei $e = \{u,v\}$ verschmelzen wir u und v zu einem Knoten mit
$$\deg_{G/e} x_{u,v} = \deg_G(u) + \deg_G(v) - 2k$$
- $\mu(G/e) \geq \mu(G)$ und falls es in G einen minimalen Schnitt C mit $e \notin C$ gibt, dann gilt
$$\mu(G/e) = \mu(G)$$
- Wir kontrahieren randomisiert bis nur noch 2 Knoten übrig sind und returnen diesen Schnitt
Laufzeit: $O(n^2)$

Erfolgsw'keit

- Sei $G = (V, E)$ ein Multigraph mit n Knoten. Falls e gleichverteilt zufällig unter den Kanten in G gewählt wird, dann gilt: $Pr[\mu(G) = \mu(G/e)] \geq 1 - \frac{2}{n}$
- $Pr[\mu(G) = \mu(G/e)] \geq Pr[e \notin C] = 1 - \frac{|C|}{|E|} \geq 1 - \frac{k}{kn/2} = 1 - \frac{2}{n}$
- $q(G) :=$ W'keit, dass $CUT(G)$ den Wert $\mu(G)$ ausgibt
- Für alle $n \geq 3$ gilt $q(n) \geq (1 - 2/n) \cdot q(n-1)$
- Algorithmus hat Laufzeit $O(\lambda n^4)$

Bootstrapping

- **Idee:** Letzte Schritte von rand. Algorithmus sind am gefährlichsten, also diese deterministisch ausführen

- Laufzeit: $O(n(n-t) + z(t))$

- Erfolgsw'keit $\geq \frac{n-2}{n} \cdot \frac{n-3}{n-1} \cdots \frac{t}{t+2} \cdot \frac{t-1}{t+1} \cdot p^*(t) = \frac{t(t-1)}{n(n-1)} \cdot p^*(t)$

- wir wählen: $z(t) = O(t^4)$ $p_3(t) = 1 - e^{-1}$ und erhalten für $t = n^{1/2}$ eine Laufzeit: $O(\lambda n^3)$

Kleinsten Umschliessender Kreis

- **Gegeben:** endliche Punktmenge $P \subseteq \mathbb{R}^2$
- **Gesucht:** Kreis mit kleinstem Radius, der P umschliesst
- Für jede Punktmenge $P \subseteq \mathbb{R}^2$, $|P| \geq 3$ gibt es eine Teilmenge $Q \subseteq P$
s.d. $|Q| = 3 \wedge C(Q) = C(P)$

Kleinstes Umschliessender Kreis

Algorithmus

RANDOMISED_PRIMITIVEVERSION(P)

```
1: repeat forever  
2:   wähle  $Q \subseteq P$  mit  $|Q| = 3$  zufällig und gleichverteilt  
3:   bestimme  $C(Q)$   
4:   if  $P \subseteq C^\bullet(Q)$  then  
5:     return  $C(Q)$ 
```

- Laufzeit: $T \sim \text{Geo}(1/\binom{n}{3}) \Rightarrow O(n \cdot n^3)$

Kleinstes Umschliessender Kreis

Algorithmus

RANDOMISED_CLEVERVERSION(P)

```
1: repeat forever  
2:   wähle  $Q \subseteq P$  mit  $|Q| = 11$  zufällig und gleichverteilt  
3:   bestimme  $C(Q)$   
4:   if  $P \subseteq C^\bullet(Q)$  then  
5:     return  $C(Q)$   
6:   verdoppele alle Punkte von  $P$  ausserhalb von  $C(Q)$ 
```

- Laufzeit: $O(n \cdot T) = O(n \cdot \log(n))$

Kleinsten Umschliessender Kreis

Beweis Lemma 3.28

$$\begin{aligned}\mathbb{E}[Y] &= \frac{1}{\binom{n}{r}} \sum_{R \in \binom{P}{r}} \sum_{p \in P \setminus R} \text{out}(p, R) \\ &= \frac{1}{\binom{n}{r}} \sum_{R \in \binom{P}{r}} \sum_{p \in P \setminus R} \text{essential}(p, R \cup \{p\}) \\ &= \frac{1}{\binom{n}{r}} \sum_{Q \in \binom{P}{r+1}} \underbrace{\sum_{p \in Q} \text{essential}(p, Q)}_{\leq 3} \\ &\leq \frac{1}{\binom{n}{r}} \sum_{Q \in \binom{P}{r+1}} 3 = 3 \cdot \frac{\binom{n}{r+1}}{\binom{n}{r}} = 3 \cdot \frac{n-r}{r+1}\end{aligned}$$

1. gehe über alle r -elementige Subsets durch und zähle die Punkte ausserhalb, am Ende bestimme den Durchschnitt
2. $\text{out}(p, R) = 1 \iff \text{essential}(p, R \cup \{p\}) = 1$
3. im Schritt davor haben wir uns sowieso schon Mengen mit $r+1$ Elementen angeschaut
4. es kann höchstens 3 essentielle Punkte geben
letzte Umformungen sind einfach Arithmetik

Kleinstes Umschliessender Kreis

Laufzeit - Upper Bound

- $X_k :=$ Anzahl Punkte nach k Iterationen

$$\mathbb{E}[X_k] = \sum_{i=0}^{\infty} \mathbb{E}[X_k | X_{k-1} = i] \cdot \Pr[X_{k-1} = i]$$

2. Abschätzung durch Lemma 3.28

$$\leq \sum_{i=0}^{\infty} \left(1 + \frac{3}{r+1}\right) \cdot i \cdot \Pr[X_{k-1} = i]$$

→ höchstens $\frac{3i}{r+1}$ ausserhalb

$$= \left(1 + \frac{3}{r+1}\right) \cdot \sum_{i=0}^{\infty} i \cdot \Pr[X_{k-1} = i]$$

$$= \left(1 + \frac{3}{r+1}\right) \cdot \mathbb{E}[X_{k-1}]$$

$$\Rightarrow \mathbb{E}[X_k] \leq \left(1 + \frac{3}{r+1}\right)^k \cdot n$$

1. Def. bedingter Erwartungswert

3. die Klammer rausziehen

4. Def. Erwartungswert

5. Teleskopieren

Kleinsten Umschliessender Kreis

Laufzeit - Lower Bound

- $X_k :=$ Anzahl Punkte nach k Iterationen
- $T :=$ Anzahl der Runden/Durchläufe der Schleife

$$\begin{aligned}\mathbb{E}[X_k] &= \mathbb{E}[X_k | T \geq k] \cdot \Pr[T \geq k] + \mathbb{E}[X_k | T < k] \cdot \Pr[T < k] \\ &\geq \mathbb{E}[X_k | T \geq k] \cdot \Pr[T \geq k] \\ &\geq 2^{k/3} \cdot \Pr[T \geq k]\end{aligned}$$

1. Def. bedingter Erwartungswert
2. Erwartungswert und Wahr-keit nicht negativ
3. mindestens einer der Punkte ist in mindesten $k/3$ der Runden ausserhalb
→ hat mindestens $2^{k/3}$ Kopien

Kleinsten Umschliessender Kreis

Laufzeit

$$2^{k/3} \cdot \Pr[T \geq k] \leq \mathbb{E}[X_k] \leq \left(1 + \frac{3}{r+1}\right)^k \cdot n \quad \Rightarrow \quad \Pr[T \geq k] \leq \min \left\{ 1, \left(\frac{1 + \frac{3}{12}}{2^{1/3}} \right)^k \cdot n \right\} \leq \min\{1, 0.995^k \cdot n\}$$

Mit $k_0 := -\log_{0.995} n$

$$\begin{aligned} \mathbb{E}[T] &= \sum_{k \geq 1} \Pr[T \geq k] \\ &\leq \sum_{k=1}^{k_0} 1 + \sum_{k > k_0} 0.995^k \cdot n \\ &= \sum_{k=1}^{k_0} 1 + \sum_{k' \geq 1} 0.995^{k'} \cdot 0.995^{k_0} \cdot n \\ &= k_0 + \mathcal{O}(1) \leq 200 \ln n + \mathcal{O}(1) \end{aligned}$$

1. Def. Erwartungswert, T nichtnegativ

2. 0.995^k dominiert erst ab $k > k_0$

3. $k = k_0 + k'$

4. $0.995^{k_0} \cdot n = 1$

$\sum_{k' \geq 1} 0.995^{k'}$ ist eine geom. Reihe