Algorithmen und Wahrscheinlichkeit

Woche 3

Ilya Maier

Bemerkungen

- Wenn man Latex ausprobieren möchte:
  - overleaf.com
  - Texifier
  - es gibt ein Template und eine Beispiel Datei auf meiner Webseite

- Pigeonhole principle + bipartiter Graph

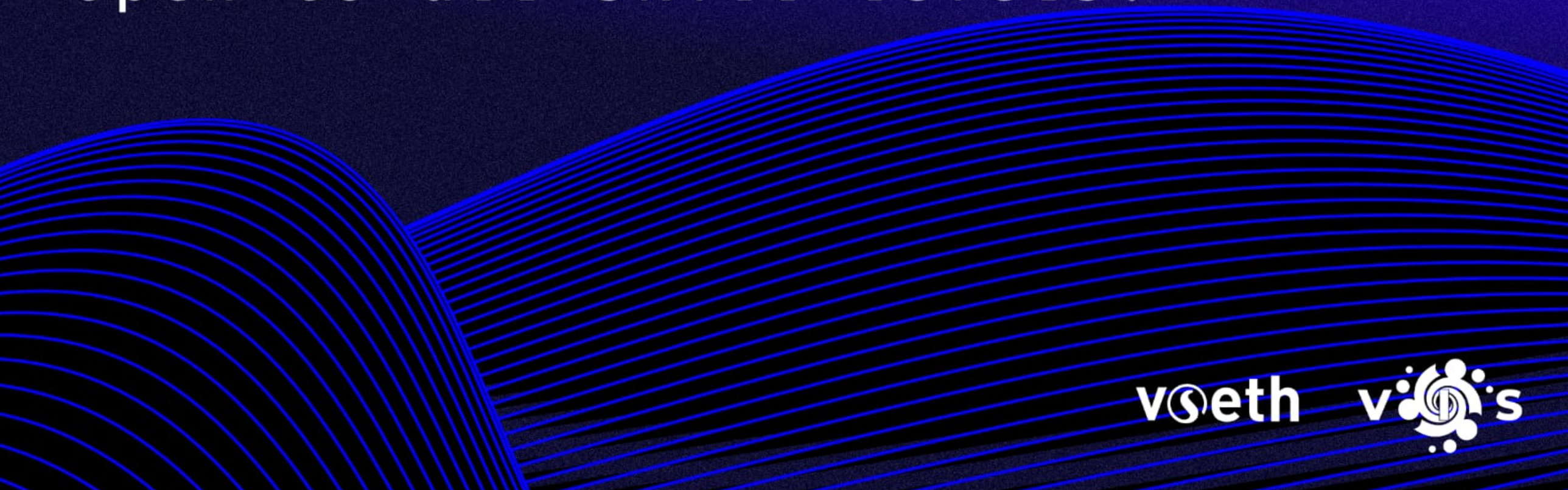
Announcements

- nächste ÜS online -> Link auf meiner Webseite

CPC Meetups Kickoff

19:00 11.03.2024 CAB H56

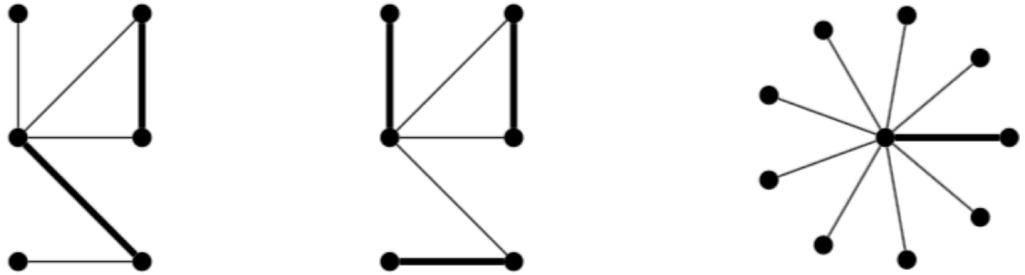
Open to all skill levels!



Matchings - Definitionen

Matching: Eine Kantenmenge $M \subseteq E$ in einem Graphen $G = (V, E)$,
wobei kein Knote in V inzident zu mehr als einer Kante in M ist (inzident zu 0 oder 1 Kante)

Überdeckt: Ein Knote v ist bedeckt von Matching M, falls v inzident zu einer Kante in M ist.



Perfektes Matching: Ein Matching M, sodass alle v überdeckt sind.

Inklusionsmaximales Matching*(maximal)*: Ein Matching M, sodass $\neg \exists M' \subseteq E : M \subseteq M' \wedge |M'| > |M|$

Kardinalitätsmaximales Matching*(maximum)*: Ein Matching M, sodass $\neg \exists M' \subseteq E : |M'| > |M|$

$\Longrightarrow$ Ein KM ist ein IM

Matchings - Sätze

Satz: Für ein IM M_{ink} und ein KM M_{kar} gilt: $|M_{ink}| \geq \frac{|M_{kar}|}{2}$

Beweis: Für jede Kante e in M_{kar} muss M_{ink} mindestens ein Endpunkt von e bedecken, sonst $M_{ink} \cup e$ ist ein Matching!! (Widerspruch)

$\implies |M_{kar}| \leq$ #Endpunkte in $M_{ink} = 2\,|M_{ink}|$

Satz von Hall (Heiratssatz)

Ein bipartiter Graph $G = (A \uplus B, E)$ hat ein Matching M der Kardinalität $|M| = |A|$ **gdw** $\forall X \subseteq A : |X| \leq |\mathcal{N}(X)|$

Korollar (Frobenius)

$\forall k$: jeder k**-reguläre bipartite Graph** hat ein **perfektes Matching**

Beweis: $\forall X \subseteq A : k\,|X| =$ #Kanten von X nach $\mathcal{N}(X) \leq k\,|\mathcal{N}(X)|$
$\implies$: $\forall X \subseteq A : |X| \leq |\mathcal{N}(X)|$

Matchings - Matching Algorithmen

Greedy

Wähle zufällig eine Kante und lösche sie und die inzidenten Kanten bis $|E| = \varnothing$

$\Longrightarrow$ Findet **ein inklusionsmaximales Matching** in $O(|E|)$

Gabows Algorithmus

In 2^k**-regulären bipartiten Graphen** kann man in Zeit $O(|E|)$ ein **perfektes Matching** bestimmen

→ 1. Finde eine Eulertour 2. Entferne jede zweite Kante →2^{k-1}-regulärer bipartiter Graph 3. Iteriere

Cole, Ost, Schirras Algorithmus

In k**-regulären bipartiten Graphen** kann man in Zeit $O(|E|)$ ein **perfektes Matching** bestimmen

Hopcroft-Karp

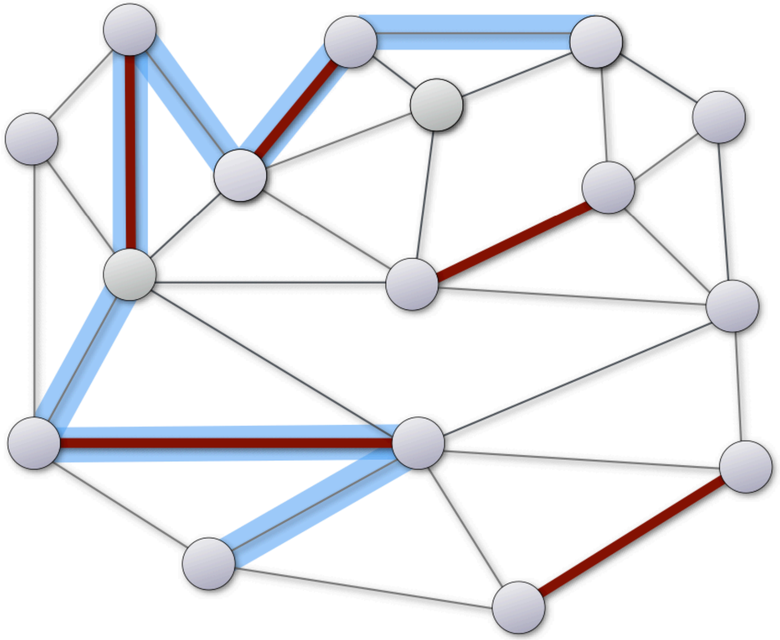
In **bipartiten Graphen** kann man in Zeit $O(\sqrt{|V|} \cdot |E|)$ ein **maximales Matching** bestimmen

Matchings - Augmentierende Pfade

M-augmentierender Pfad:

1) abwechselnd Kanten aus nicht M und M
2) beginnt und endet in einem von M unüberdeckten Knoten

Das Vergrößern von M mit einem M-augmentierenden Pfad P: $M' = M \oplus P$



Matchings - Augmentierende Pfade

M-augmentierender Pfad:

1) abwechselnd Kanten aus nicht M und M
2) beginnt und endet in einem von M unüberdeckten Knoten

Das Vergrößern von M mit einem M-augmentierenden Pfad P: $M' = M \oplus P$

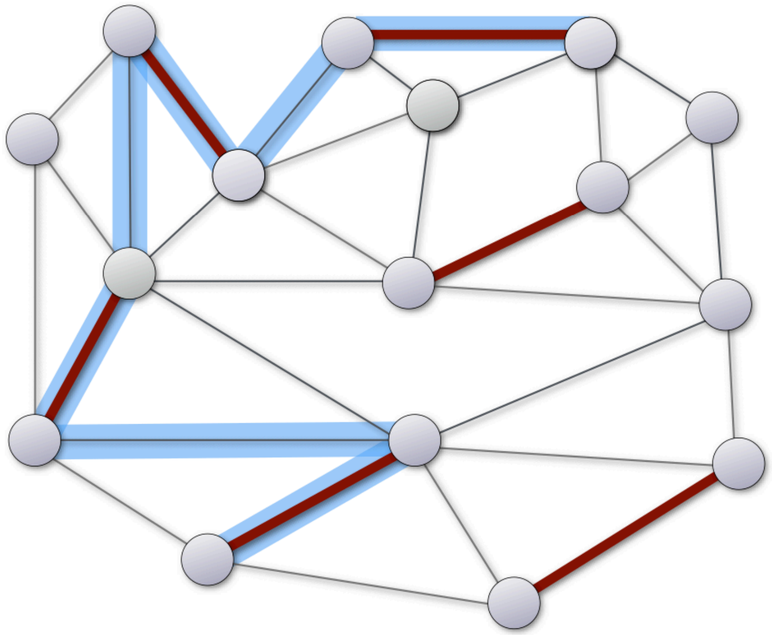
Satz von Berge:

Jedes Matching, das nicht kardinalitätsmaximal ist,

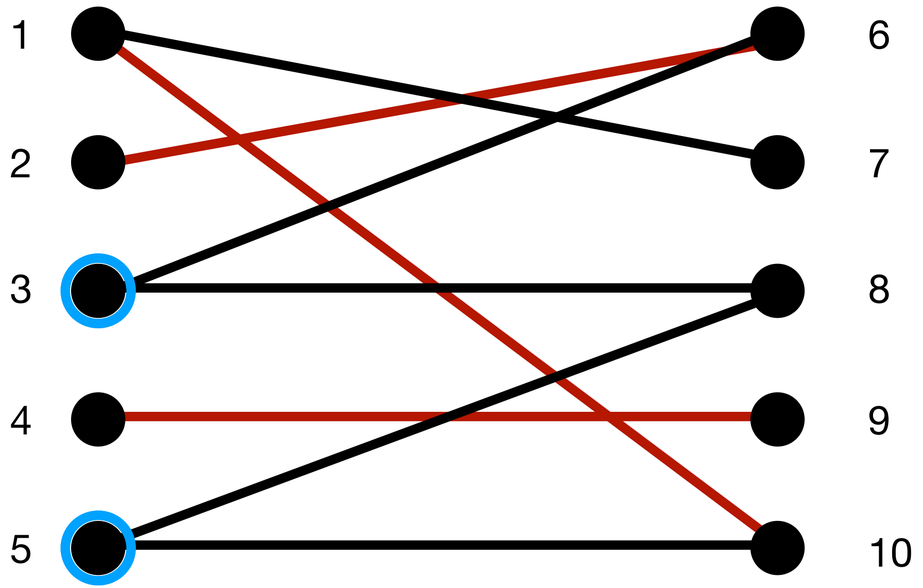
besitzt einen augmentierenden Pfad

Für zwei beliebige Matchings M und M' wobei $|M| < |M'|$:

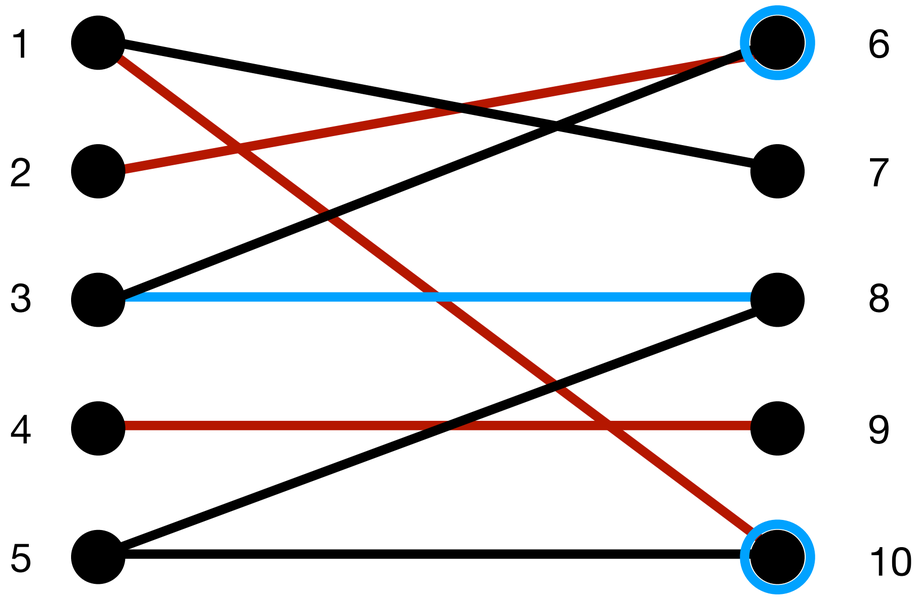
$M \oplus M'$ hat **mindestens $|M'| - |M|$ knoten-disjunkt M-augmentierende Pfade**



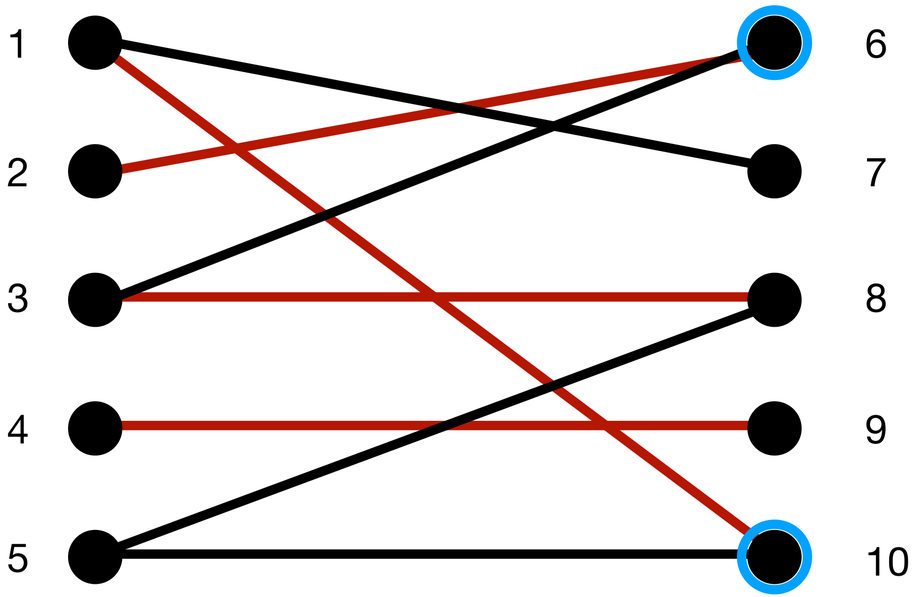
Matchings - Augmentierende Pfade



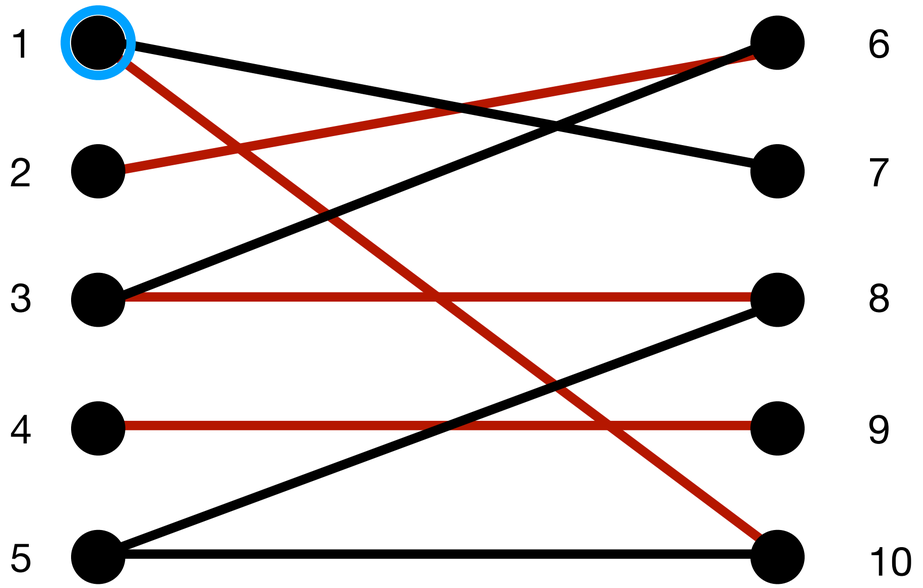
Matchings - Augmentierende Pfade



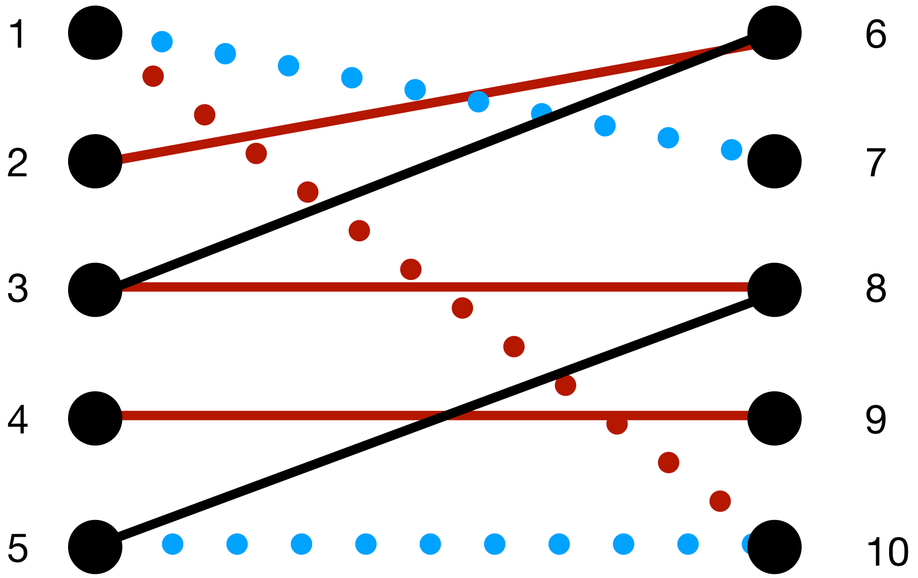
Matchings - Augmentierende Pfade



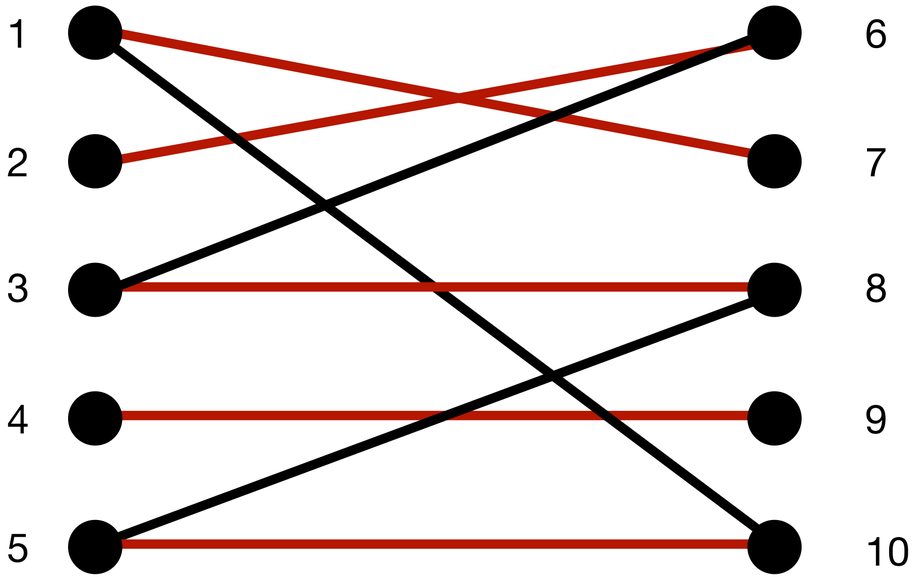
Matchings - Augmentierende Pfade



Matchings - Augmentierende Pfade



Matchings - Augmentierende Pfade



Matchings - Augmentierende Pfade

Algorithmus fürs Finden eines augmentierenden Pfades (bipartit)

Eingabe: ein bipartiter Graph $G = (A \uplus B, E)$, ein Matching M
Ausgabe: kürzester augmentierender Pfad, falls existiert

L_0 = {unüberdeckte Knoten aus A}
Markiere die Knoten in L_0 als *besucht*
for i = 1 … n
  if i ungerade **then**
    $L_i = \{$unbesuchte Nachbarn von L_{i-1} via Kanten in $E \backslash M\}$
  else
    $L_i = \{$unbesuchte Nachbarn von L_{i-1} via Kanten in $M\}$
  Markiere die Knoten in L_i als *besucht*
  if ein Knote v in L_i ist nicht überdeckt $\Longrightarrow$ **return** Pfad zu v

Laufzeit
$O(|E|)$ (BFS)

Algorithmus fürs maximale Matching

Eingabe: $G = (V, E)$
Ausgabe: KM Matching M

Starte mit $M = \emptyset$
  repeat
    Suche augmentierenden Pfad P
    if kein solcher Pfad existiert **then return** M
    else $M = M \oplus P$

Laufzeit
$O(|V| \cdot |E|)$

Hopcroft-Karp
$O(\sqrt{|V|} \cdot |E|)$

Matchings - Christofides Algorithmus

2-Approximation

Eingabe: K_n, metrische Längenfunktion l

Output: Ein Hamiltonkreis, C, sodass $l(C) \leq 2 \cdot \text{opt}(K_n, l)$

1. Finde den **MST** T von G.
2. **Verdopple** alle Kanten in T $\rightarrow$ Denn alle Knoten müssen einen geraden Grad haben für eine Eulertour
3. Bestimmt **Eulertour** W
4. **Kürze** W **ab**, sodass jeder Knoten nur einmal besucht wird $\Longrightarrow$ Hamiltonkreis C

Matchings - Christofides Algorithmus

1.5-Approximation (Christofides)

Eingabe: K_n, metrische Längenfunktion l

Output: Ein Hamiltonkreis, C, sodass $l(C) \leq 1.5 \cdot \text{opt}(K_n, l)$

1. Finde den **MST** T von G.
2. Finde **minimales perfektes Matching** M von $G[U]$ wobei $U := \{v \in T \mid \deg(v) \text{ ungerade}\}$
3. **Füge M zu T hinzu** (Nun haben alle Knoten einen geraden Grad)
4. Bestimmt **Eulertour** W
5. **Kürze W ab**, sodass jeder Knoten nur einmal besucht wird $\Longrightarrow$ Hamiltonkreis C

Analysis

1. Für eine Kante e im Hamiltonkreis H, $H - e$ ist ein Spannbaum. Von daher: $l(T) \leq \text{opt}(K_n, l)$
2. Da H ein Kreis ist, $l(M) \leq \frac{1}{2}\text{opt}(K_n, l)$
3. Eulertour $l(W) = l(T) + l(M) \leq 1.5\text{opt}(K_n, l)$
4. Abkürzen: $l(C) \leq l(W) \leq 1.5\text{opt}(K_n, l)$

Färbungen

Färbung eines Graphen (V, E) **mit** k **Farben**: eine Abbildung $c : V \to [k]$ s.d. $c(u) \neq c(v)$ für alle Kanten $\{u, v\} \in E$

bzw. $V = V_1 \dot{\cup} \ldots \dot{\cup} V_k$, wobei V_i keine Kanten enthält, V_i := Farbklasse

Chromatische Zahl $\chi(G)$: minimale Anzahl Farben, die für eine Färbung von G benötigt wird.

$$\chi(G) \leq k \iff G \text{ ist } k\text{-partit}$$

Gegeben ein Graph $G = (V, E)$**, gilt** $\chi(G) \leq k$?

$\underline{k = 2}$: In $O(|V| + |E|)$ Zeit mit BFS (keine ungeraden Kreise)

$\underline{k > 2}$: **NP-vollständig**

Farbklassen tauschen:

Falls wir **jeden Block** mit k Farben färben können,

können wir **den ganzen Graphen** mit k Farben färben

Kahoot!

Aufgaben