

# Algorithms and Probability

Week 3

G09 - mkilic

5.III.2026

# Overview

---

1. Matchings
2. Augmentierende Pfade
3. Exercise: Bipartite Matching
4. Der Satz von Hall
5. Extra Exercise: Quantitative Hall's Theorem

# Roadmap

---

## 1. Graphentheorie

- Zusammenhang
- Kreise
- Matchings
- Färbungen

## 2. W'keitstheorie

- Bedingte W'keit
- Unabhängigkeit
- (mehrere) Zufallsvariablen
- Diskrete Verteilungen
- Abschätzen von W'keiten
- Randomisierte Algorithmen

## 3. Algorithmen

- Lange-Bunte Pfade
- MaxFlow
- MinCut
- Kleinster umschliessender Kreis
- Konvexe Hülle

# Matchings

---

## Matchings

Eine Kantenmenge  $M \subseteq E$  heisst **Matching** in einem Graphen  $G = (V, E)$ , falls kein Knoten des Grahen zu mehr als einer Kante aus  $M$  inzident ist.

Formal heisst das:  $e \cap f = \emptyset$  für alle  $e, f \in M$  mit  $e \neq f$

- Matching ist eine Menge von Kanten ohne gemeinsame Knoten.
- Ein Knoten  $v$  wird von  $M$  **überdeckt**, falls es eine Kante  $e \in M$  gibt, die  $v$  enthält.
- $M$  heisst **perfektes Matching** wenn jeder Knoten durch genau eine Kante aus  $M$  überdeckt wird. Äquivalent ist  $|M| = |V|/2$

---

<sup>0</sup>Engl. *überdeckt: matched*

# Maximalität

---

## Inklusionsmaximal

Sei  $G = (V, E)$  ein Graph und  $M$  ein matching in  $G$ :

$M$  heisst *inklusionsmaximal*, falls gilt  $M \cup \{e\}$  ist kein Matching für alle Kanten  $e \in E \setminus M$

Äquivalent sind:

- Man kann keine Kante mehr hinzufügen ohne die Matching Eigenschaft zu zerstören.
- Es existiert kein Matching  $M'$  mit  $M \subseteq M'$  und  $|M'| > |M|$
- Mit den Kanten, die ich zurzeit im Matching habe, geht es nicht besser
- Jede Kante muss mindestens eine Kante in  $M$  berühren.

---

<sup>0</sup>Engl. *inklusionsmaximales Matching* : *maximal Matching*

# Maximalität

---

## Kardinalitätsmaximal

Sei  $G = (V, E)$  ein Graph und  $M$  ein matching in  $G$ :

$M$  heisst *kardinalitätsmaximal*, falls gilt  $|M| \geq |M'|$  für alle Matchings  $M'$  in  $G$ .

- Es gibt kein Matching, das mehr Kanten enthält als  $M$ .
- Es geht nicht besser.

---

<sup>0</sup>En *inklusionsmax. Matching* : maximal Matching, *kardinalitätsmax. Matching* : maximum Matching

# Maximalität

---

Ein kardinalitätsmaximales Matching ist **immer** auch inklusionsmaximal.



Kardinalitätsmaximales Matching auch inklusionsmaximal

Ein inklusionsmaximales Matching muss nicht unbedingt kardinalitätsmaximal sein.



Inklusionsmaximales Matching, nicht kardinalitätsmaximal

# Greedy Algorithmus

## Greedy

Mit dem *Greedy Algorithmus* kann man in Zeit  $O(|E|)$  ein inklusionsmaximales Matching  $M_{greedy}$  bestimmen mit:

$$|M_{greedy}| \geq \frac{|M_{max}|}{2}$$

wobei  $M_{max}$  ein kardinalitätsmaximales Matching ist.

```
1  M ← ∅
2  while E ≠ ∅ do
3      wähle eine beliebige Kante e ∈ E
4      M ← M ∪ {e}
5      lösche e und alle inzidenten Kanten in G
6
```

# Greedy Algorithmus

---

Beweis von der Box:

1. Greedy findet inklusionsmaximales Matching per konstruktion. (Lösche inzidente Kanten, do while  $E \neq \emptyset$ )
2. Bei jeder Kante aus  $M_{max}$  muss mindestens einen Endpunkt in  $M_{Greedy}$  überdeckt sein, da die Kante sonst zu  $M_{Greedy}$  hinzugefügt werden könnte.
3. Da  $M_{max}$  ein Matching ist, kann jeder von diesen Endpunkten nur eine Kante in  $M_{max}$  berühren.
4. Also gibt es mindestens so viele Endpunkte in  $M_{Greedy}$  wie es Kanten in  $M_{max}$  gibt:

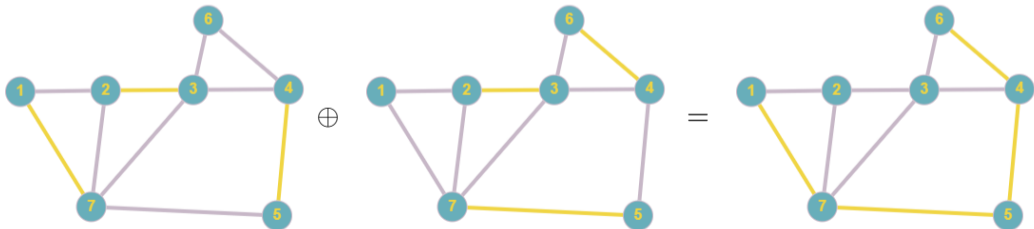
$$|M_{max}| \leq |\text{Endpunkte in } M_{Greedy}| = 2 \cdot |M_{Greedy}|$$

# Exor von zwei Matchings

## Exklusives Oder

Exklusives Oder von zwei Matchings (oder im Allgemeinen von zwei Kantenmengen)  $M_1$  und  $M_2$  ist:

$$M_1 \oplus M_2 = (M_1 \cup M_2) \setminus (M_1 \cap M_2)$$



## Augmentierende Pfade (Augmenting Paths)

---

Im Folgenden seien  $M_1$  und  $M_2$  zwei beliebige Matchings in  $G = (V, E)$ .

1. Jeder Knoten im Graphen  $G_M = (V, M_1 \oplus M_2)$  hat Grad höchstens 2.
2. Wegen 1. können die ZHKe von  $G$  alles Pfade und/oder Kreise sein wobei alle Kreise gerade Länge haben müssen (die Kanten sind abwechselnd aus  $M_1$  und  $M_2 =$  bipartiter Kreis)
3. Jede Komponente, die ein Kreis oder ein Pfad mit gerader Länge ist, enthält jeweils gleich viele Kanten aus  $M_1$  und  $M_2$

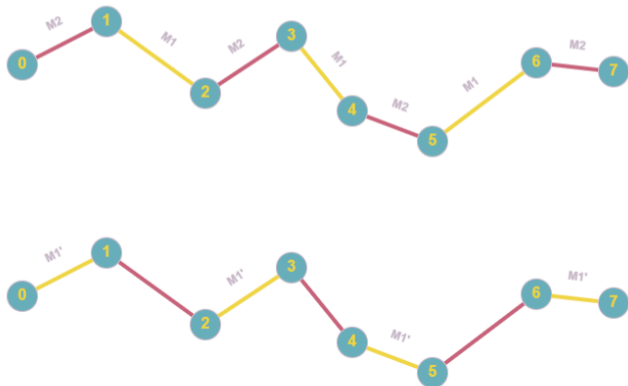
Sei jetzt o.B.d.A.  $|M_1| < |M_2|$

4. Wegen 3., muss es ZHKe geben, die Pfade  $P_i$  sind, die mehr Kanten aus  $M_2$  als aus  $M_1$  enthalten.
5. Da sich Kanten aus  $M_1$  und  $M_2$  abwechseln, kann ein solcher Pfad höchstens eine Kante mehr aus  $M_2$  enthalten (Beide äussere Kanten von  $P$  gehören zu  $M_2$ ).
6. Aus 4. + 5. folgt:  $|M_2| = |M_1| + k \Rightarrow$  es muss mindestens  $k$  solche (wie in 4.) Pfade geben.

# Augmentierende Pfade

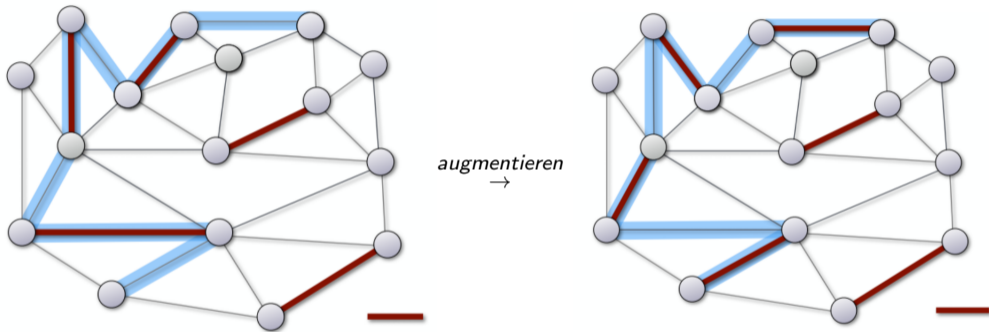
Jetzt können wir aus dem Matching  $M_1$  ein neues Matching  $M'_1$  erhalten, das eine Kante mehr enthält. Wir haben einen Pfad  $P$  wie oben beschrieben.

Wir **augmentieren**: Setze  $M'_1 := M_1 \oplus P$ .



# Augmentierende Pfade

---



$M_1' > M_1$  Also haben wir jetzt ein grösseres Matching!

---

<sup>0</sup>Hier  $M_1$  : rote Kanten,  $P$  : blaue Kanten

# Augmentierende Pfade

---

## M-augmentierender Pfad

Formal ist ein M-augmentierender Pfad definiert durch: Die beiden Endknoten von P werden durch M nicht überdeckt und P besteht abwechselnd aus Kanten, die nicht zu M gehören, und Kanten aus M.

## Satz von Berge

Ist M ein Matching in einem Graphen  $G = (V, E)$ , das nicht kardinalitätsmaximal ist, so existiert ein augmentierender Pfad zu M.

- Eine einzelne Kante, die keine Endpunkte in M hat, ist ein M-augmentierender Pfad der Länge 1.

# Aug. Pfade: Algorithmus

---

Mit diesen Ideen kann man einen ersten Algorithmus konstruieren, der **kardinalitätsmaximale Matching** bestimmt.

```
1   Start with  $M = \emptyset$ 
2   REPEAT:
3       Search for augmenting path  $P$ 
4       if no such path exists then return  $M$ 
5       else  $M = M \oplus P$ 
6
```

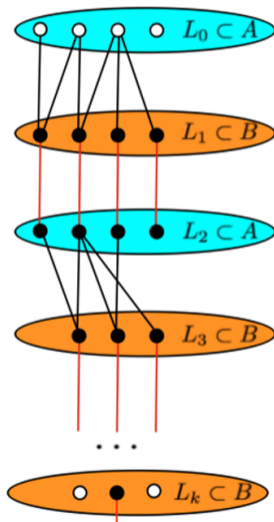
*Spätestens nach  $\frac{|V|}{2} - 1$  solcher Schritten ist das Matching dann maximal, denn ein Matching kann nicht mehr als  $\frac{|V|}{2}$  viele Kanten enthalten.*

Wie suchen wir augmentierende Pfade effizient? BFS! (in bipartiten Graphen)

# Aug. Pfade: Algorithmus

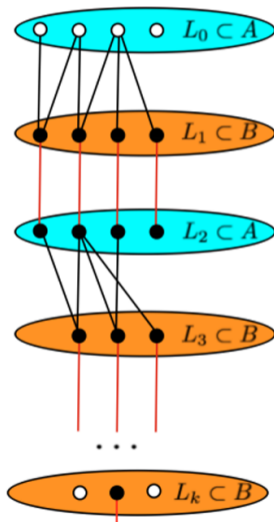
Sei  $G = (A \uplus B, E)$  ein bipartiter Graph:

1. Ein augmentierender Pfad muss immer ungerade Länge haben, denn er hat eine Kante aus  $E \setminus M$  mehr als Kanten in  $M$ .
2. Deshalb für alle augmentierenden Pfade gilt: Eine Endpunkt in A eine Endpunkt in B
3. Wegen (2.) reicht es mit unüberdeckten Knoten in A anzufangen. Alle augmentierende Pfade haben einen Endpunkt in der Menge von unüberdeckten Knoten in A. Keine wird übersehen.



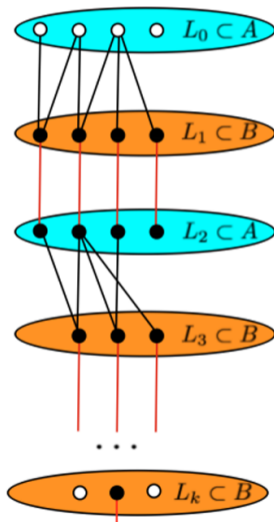
## Aug. Pfade: Algorithmus

- Da wir mit unüberdeckten Knoten anfangen benutzen wir Kanten aus  $E \setminus M$  um von  $L_0$  zu  $L_1$  zu übergehen.
- Weil wir beim Schritt (4.)  $E \setminus M$  Kanten benutzt haben, landen wir auf ein Knoten aus  $B$ , von denen wir mit einer Matching Kante weitergehen können.
- Dann benutzen wir für jeden Knoten in  $L_1$  die zu einem Knoten in der Menge von unüberdeckten Knoten in  $A$  benachbart sind, die für jeden Knoten eindeutig bestimmten Matching Kanten aus  $M$  die uns von überdeckten Knoten in  $B$  zu überdeckten Knoten in  $A$  führen.



# Aug. Pfade: Algorithmus

- Knoten aus  $B$  erreichen wir daher grundsätzlich über Kanten aus  $E \setminus M$ , und Knoten aus  $A$  über die (eindeutig bestimmte) inzidente Kante aus  $M$ .
- Insbesondere liegen alle unüberdeckten Knoten, die der Algorithmus besucht, im ersten und letzten Layer.
- Wir stoppen falls  $L_i$  unüberdeckte Knoten enthält. Weil das würde heissen, dass wir einen unüberdeckten Knoten aus  $B$  entdeckt haben, da wir alle unüberdeckten Knoten aus  $A$  in  $L_0$  hinzugefügt haben.



# Pseudocode

---

```
AUGMENTING_PATH (G = (A  $\uplus$  B, E), M)
1: L0 := {unüberdeckte Knoten in A}
2: Markiere alle Knoten aus L0 als besucht.
3: if L0 = ∅ then
4:   return M ist maximal
5: for all i = 1 to n do
6:   if i ungerade then
7:     Li := {unbesuchte Nachbarn von Li-1 via Kanten in E \ M}
8:   else
9:     Li := {unbesuchte Nachbarn von Li-1 via Kanten in M}
10:  Markiere alle Knoten aus Li als besucht.
11:  if Li enthält unüberdeckten Knoten v then
12:    Finde Pfad P von L0 nach v durch backtracking
13:    return P // terminiert Algorithmus
14: return M ist schon maximal
```

---

# Laufzeit

---

## Laufzeitanalyse

Mit der modifizierten Breitensuche kann man einen augmentierenden Pfad in Zeit  $O(|E|)$  finden. (falls er existiert)

Wir müssen höchstens  $|V|/2$  oft augmentieren, denn es gilt immer  $|M_{max}| < |V|/2$ .  
Insgesamt erhalten wir einen Algorithmus mit Laufzeit  $O(|V||E|)$ .

Hopcroft und Karp hatten eine bessere Idee (nächste Woche).

## Exercise: Bipartite Matching

---

Let  $G = (A \dot{\cup} B, E)$  be a bipartite graph.

- (a) Prove or disprove: If  $G$  has a Hamiltonian cycle, then  $G$  contains two disjoint perfect matchings.
- (b) Prove or disprove: If  $G$  contains two disjoint perfect matchings, then  $G$  has a Hamiltonian cycle.
- (c) Let  $A' \subseteq A$  and  $B' \subseteq B$ . Assume that there is a matching  $M_A$  covering  $A'$  and a matching  $M_B$  covering  $B'$  (these two matchings do not have to be disjoint!). Show that there is a matching  $M$  that covers both  $A'$  and  $B'$  (i.e. it covers  $A' \cup B'$ ). (*Hint:* Which properties has the graph  $(V, M_A \cup M_B)$ ? Try to build a matching using only edges in  $M_A \cup M_B$ .)

# Satz von Hall a.k.a. Der Heiratssatz

---

## Satz von Hall

Für einen bipartiten Graphen  $G = (A \uplus B, E)$  gibt es **genau dann** ein Matching  $M$  der Kardinalität  $|M| = |A|$ , **wenn** gilt

$$|N(X)| \geq |X| \text{ für alle } X \subseteq A$$

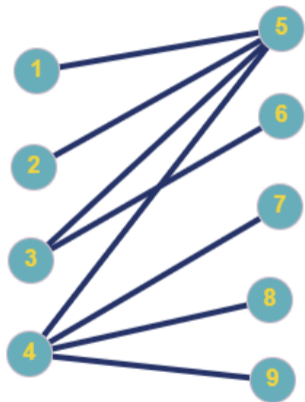
Wir definieren dabei die Nachbarschaft einer Knotenmenge  $X \subseteq V$  als:

$$N(X) = \bigcup_{v \in X} N(v)$$

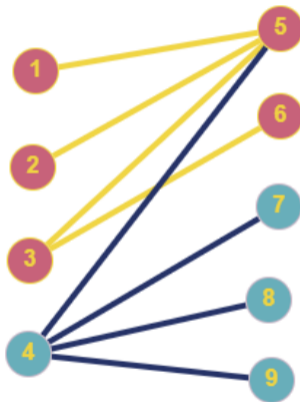
---

<sup>0</sup>Achtung! Dieser Satz gilt in beide Richtungen:  $\exists M. |M| = |A| \iff |N(X)| \geq |X| \text{ für alle } X \subseteq A$

# Satz von Hall - Beispiel



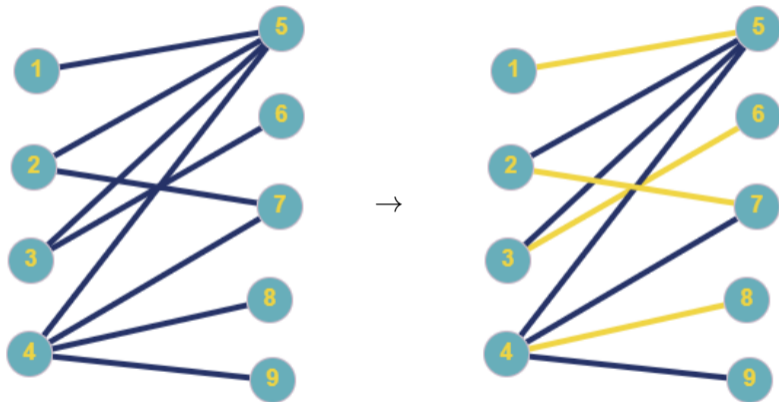
$$X = \{1, 2, 3\} \\ \subseteq A$$



Kein Matching der Grösse von  $|A| = |\{1, 2, 3, 4\}|$ .

## Satz von Hall - Beispiel

---



$\{\{1, 5\}, \{2, 7\}, \{3, 6\}, \{4, 8\}\}$  ist ein Matching der Grösse von  $|A| = |\{1, 2, 3, 4\}|$ .

# Satz von Hall - Beweis

---

Satz von Hall:  $\exists M. |M| = |A| \iff |N(X)| \geq |X|$  für alle  $X \subseteq A$

( $\Rightarrow$ ): Wir zeigen, falls es ein Matching  $M$  mit  $|M| = |A|$  existiert, dann gilt  $|N(X)| \geq |X|$  für alle  $X \subseteq A$ .

Sei jetzt  $M$  ein Matching Kardinalität  $|M| = |A|$ . In dem durch  $M$  gegebenen Teilgraphen  $H = (A \uplus B, M)$  hat jede Teilmenge  $X \subseteq A$  nach Definition eines Matchings genau  $|X|$  Nachbarn. Es gilt also  $|N(X)| \geq |X|$  für alle  $X \subseteq A$ .

# Satz von Hall - Beweis

---

( $\Leftarrow$ ): Wir zeigen jetzt, falls es gilt  $|N(X)| \geq |X|$  für alle  $X \subseteq A$  ( $\star$ ) dann existiert ein Matching  $M$  mit  $|M| = |A|$ .

Das machen wir per Induktion über die Kardinalität von  $A$ :  $|A| = a$

*Base Case:*  $a = 1$  Hier impliziert ( $\star$ ), dass der einzige Knoten in  $A$  zu mindestens einer Kante inzident ist. Diese Kante (od. jede solche Kante falls es mehrere gibt) ist ein Matching.

*Induktionsannahme:* Für  $|A| < a$  gilt  $|N(X)| \geq |X|$  für alle  $X \subseteq A \Rightarrow \exists M. |M| = |A|$

# Satz von Hall - Beweis

---

Für  $a > 1$  machen wir eine Fallunterscheidung:

1.  $\forall : \emptyset \neq X \subsetneq A : |X| < |N(X)|$  ( $\star$ ) gilt als strikte Ungleichung
2.  $\exists : \emptyset \neq X_0 \subsetneq A : |X_0| = |N(X_0)|$  ( $\star$ ) gilt für mind. eine Teilmenge mit Gleichheit

# Satz von Hall - Beweis

---

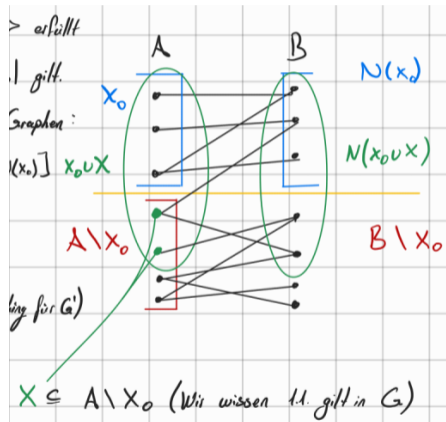
**1.Fall:** Falls  $(\star)$  mit strikter Ungleichung gilt dann wählen wir eine beliebige Kante  $e = \{x, y\}$  des Graphen, fügen  $e$  zum matching hinzu.

1. Wir löschen  $x, y$  und alle inzidente Kanten.
2. Sei  $G' = G[V \setminus \{x, y\}]$
3. Da  $(\star)$  in  $G$  mit  $<$  erfüllt war, ist  $(\star)$  jetzt auch für  $G'$  erfüllt, weil wir nur einen Knoten von  $B$  gelöscht haben.  $|X| < |N(X)| \Rightarrow |X| \leq |N(X)| - 1$
4. Wir benutzen die Induktionsannahme um die Kante  $e$  durch ein Matching so ergänzen dass  $M$  alle Knoten in  $A$  überdeckt.

# Satz von Hall - Beweis

**2.Fall:** Falls  $(*)$  nicht für alle Teilmengen mit  $>$  erfüllt ist, so existiert eine nichtleere Teilmenge  $X_0$  von  $A$  wobei  $X_0 \subsetneq A$  für die gilt:  $|N(X_0)| = |X_0|$

1. Dann können wir die Induktionsannahme auf beiden induzierten Teilgraphen anwenden. Diese Graphen sind knotendisjunkt:  $G' = [X_0 \uplus N(X_0)]$   
 $G'' = [A \setminus X_0 \uplus B \setminus N(X_0)]$
2. Die Bedingung  $(*)$  ist offensichtlich für  $G'$  erfüllt da es gilt  $|N(X_0)| = |X_0|$ .



# Satz von Hall - Beweis

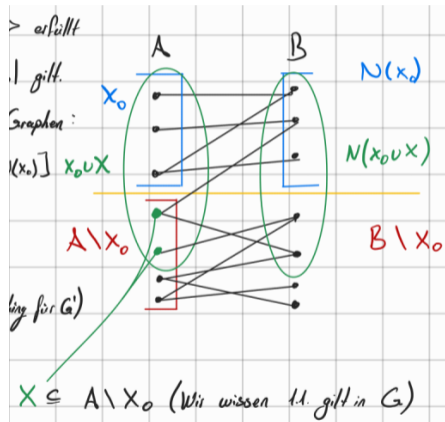
Die Bedingung (\*) folgt für  $G''$  aus Folgendem:

3. Sei  $X \subseteq A \setminus X_0$ . Da (\*) für  $G$  gilt wissen wir

$$|X| + |X_0| \stackrel{\text{(disjunkt)}}{=} |X \cup X_0| \stackrel{\text{(*) für } G}{\leq} |N(X \cup X_0)| \\ = |N(X_0)| + |N(X) \setminus N(X_0)|$$

4. Wegen  $|N(X_0)| = |X_0|$  folgt daraus

$$|X| \leq |N(X) \setminus N(X_0)| = |N(X) \cap (B \setminus N(X_0))|.$$



## Satz von Hall - Beweis

---

- Das heisst die Nachbarschaft von  $X$  in dem Graphen  $G''$  besteht aus mindestens  $|X|$  Knoten. Da dies für alle Teilmengen gilt, gilt  $(\star)$  auch für  $G''$ .
- Aus der Induktionsannahme folgt daher: Es gibt ein Matching  $M'$  in  $G'$ , das alle Knoten in  $X_0$  überdeckt, und ein Matching  $M''$  in  $G''$ , das alle Knoten in  $A \setminus X_0$  überdeckt.  $M = M' \cup M''$  ist dann ein Matching in  $G$  mit  $|M| = |A|$ .



## Extra Exercise: Quantitative Hall's Theorem

---

For a bipartite graph  $G = (A \cup B, E)$ , with  $|A| = |B| = n$ , we define a *deficiency* of  $G$  ( $\text{deff}(G)$ ) to be a maximum over  $X \subset A$  of  $|X| - |N(X)|$ . The Hall's theorem states that  $G$  has a perfect matching if and only if the deficiency of  $G$  is 0. We will show a quantitative version of Hall's theorem, using the Hall's theorem itself.

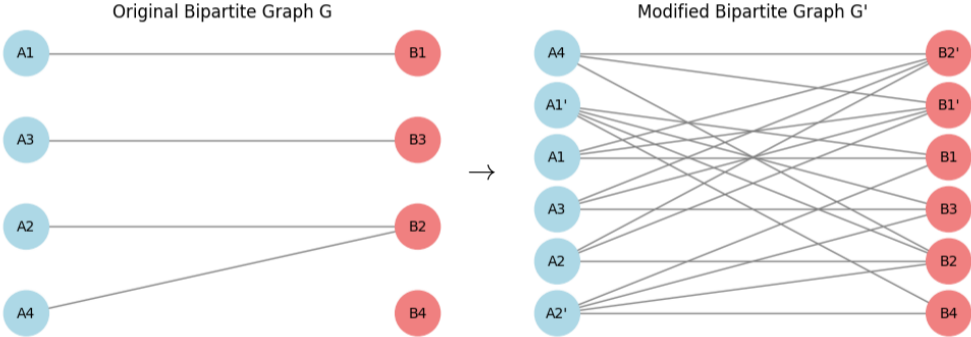
## Exercise: Quantitative Hall's Theorem

---

Specifically, show that the size of the largest (in terms of the number of edges) matching  $M$  in the bipartite graph  $G$  is equal to  $n - \text{deff}(G)$ .

1. Show that if  $\text{deff}(G) = k$  then every matching  $M$  has size  $|M| \leq n - k$ . (*Hint: this can be done with a direct argument.*)
2. Show that when  $\text{deff}(G) \leq k$ , there is a matching  $M$  in  $G$  of size at least  $n - k$ . (*Hint: modify the graph  $G$  by adding  $k$  vertices on each side, and apply Hall's theorem to the modified graph  $G'$ .)*

# Visualization of the solution



The End

