

TSP

Gegeben: • Vollständiger Graph mit n Knoten

• Distanzen zwischen allen Knoten:

Abkürzung für $\{1, 2, \dots, n\} = V$

$\ell \cdot \binom{[n]}{2} \rightarrow \mathbb{R}$

Funktionsname ℓ für length

Jedem Knotenpaar wird eine Länge $\in \mathbb{R}$ zugeordnet

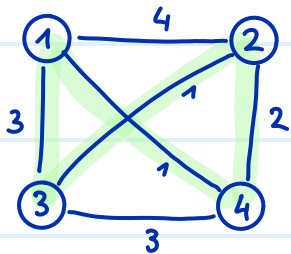
Menge aller Knotenpaare
= Kantenmenge = E

" $[n]$ choose 2"

Gesucht: minimale Länge eines Hamiltonkreises

(die Summe aller Kantengewichte $\sum \ell(e)$ vom Hamiltonkreis soll minimal sein)

Bsp.



$$V = [4] = \{1, 2, 3, 4\}$$

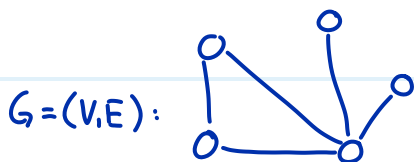
$$E = \binom{[4]}{2} = \left\{ \{1, 2\}, \{1, 3\}, \{1, 4\}, \{2, 3\}, \{2, 4\}, \{3, 4\} \right\}$$

minimale Länge eines Hamiltonkreises? $\rightarrow$

NP vollständig: Wenn man TSP effizient lösen könnte, könnte man auch effizient bestimmen, ob es einen Hamiltonkreis gibt (... was sehr wahrscheinlich nicht geht)

Idee: wir formulieren das Hamiltonkreis-Problem zu einem TSP-Problem um

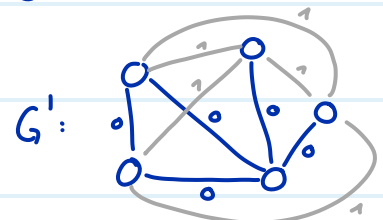
$$\exists \text{ Hamiltonkreis in } G \iff \text{TSP Lösung von } G' = 0$$



Konstruktion von G'

fehlende Kanten $\rightarrow \ell(e) := 1$

Kanten in $E \rightarrow \ell(e) := 0$

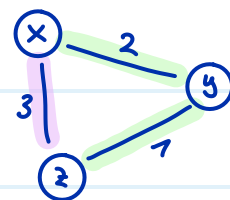


Deshalb kann es auch keinen C -Approximationsalgorithmus geben für $C > 0$

→ Sehr hoffnungslos...

Metrisches TSP: erfüllt zusätzlich die Dreiecksungleichung

$$\forall x, y, z \in V: \ell(x, z) \leq \ell(x, y) + \ell(y, z)$$



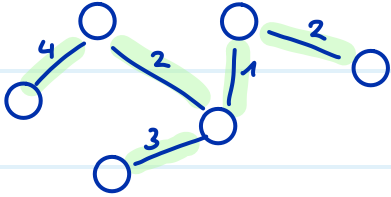
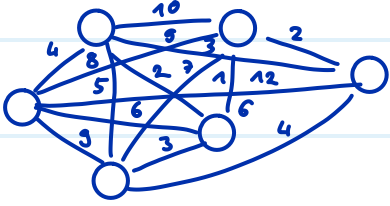
$\Leftrightarrow$ Abkürzen macht die Länge eines Weges kürzer oder sie bleibt gleich

Aufgabe: Sei $P = (a, b, c, d)$ ein Pfad. Zeige dass $\ell(a, d) \leq \ell(P)$ mit der Annahme, dass die Dreiecksungleichung gilt.

$$\begin{aligned} \ell(P) &= \ell(a, b) + \ell(b, c) + \ell(c, d) && \text{Dreiecksungleichung mit } a, b, c \\ &\geq \ell(a, c) + \ell(c, d) && \text{Dreiecksungleichung mit } a, c, d \\ &\geq \ell(a, d) \end{aligned}$$

2-Approximationsalgorithmus für METRISCHES TSP (nicht für normales TSP)

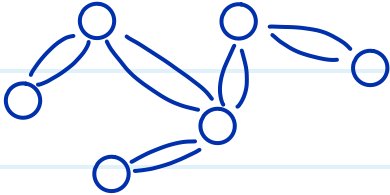
sei OPT die optimale Lösung



1. Finde MST

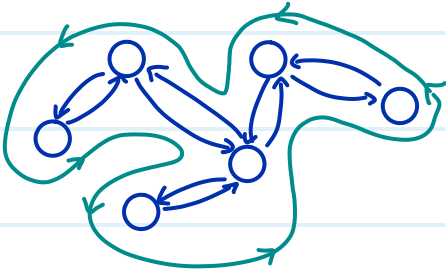
$$\ell(\text{MST}) \leq \text{OPT}$$

↳ teuerste Kante von OPT entfernen ergibt einen Spannbaum



2. Verdopple MST Kanten

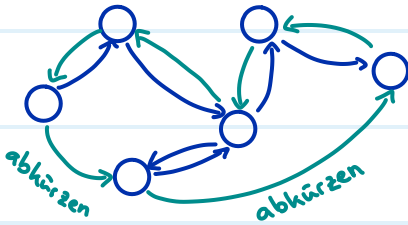
$$2 \cdot \ell(\text{MST}) \leq 2 \cdot \text{OPT}$$



3. $W :=$ Eulertour der verdoppelten MST Kanten
einfach einmal "um den MST herum laufen"

$$\ell(W) = 2 \cdot \ell(\text{MST}) \leq 2 \cdot \text{OPT}$$

↳ def Eulertour



4. $C :=$ Laufe W ab und kürze ab
(bereits verwendete Knoten überspringen)

$$\ell(C) \leq \ell(W) \leq 2 \cdot \text{OPT}$$

↳ abkürzen, Dreiecksungleichung

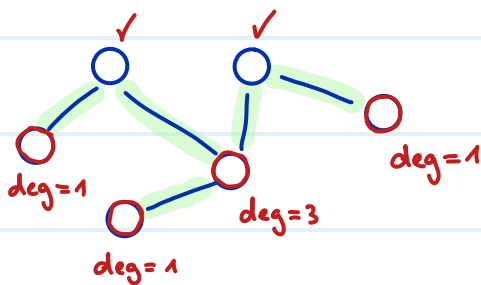
- C ist ein Hamiltonkreis weil wir jeden Knoten genau einmal besuchen,
- Approximation ist im worst case doppelt so lange wie OPT
- Laufzeit $O(n^2)$ mit Adjazenzmatrix

1.5-Approximationsalgorithmus für METRISCHES TSP (nicht für normales TSP)

Frage: Wieso haben wir die MST Kanten verdoppelt?

→ Damit alle Knoten einen geraden Grad haben bezüglich der verdoppelten Kanten, und eine Eulertour existiert

Idee Statt alle Kanten zu verdoppeln, fügen wir nur Kanten zwischen den "Problemknoten" ein, die einen ungeraden Grad haben im MST



Anzahl Knoten mit ungeradem Grad ist gerade
(Handschlaglemma)

Sei M ein gewichtsminimales perfektes Matching der Problemknoten

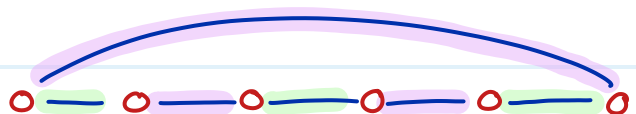
$$\ell(M) \leq \frac{1}{2} \text{OPT}$$

Beweis: Sei OPT_p die Länge der optimalen TSP Route, wenn wir nur die Problemknoten betrachten

$$\text{OPT}_p \leq \text{OPT}$$

↳ abkürzen

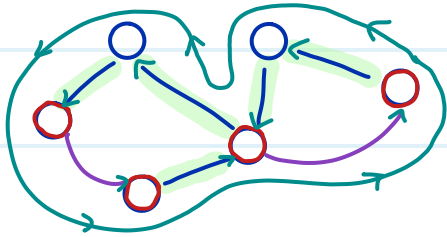
wir teilen die Kanten abwechselnd in zwei Mengen (A und B)



$$\text{entweder } \ell(A) \leq \frac{1}{2} \text{OPT}_p \text{ oder } \ell(B) \leq \frac{1}{2} \text{OPT}_p$$

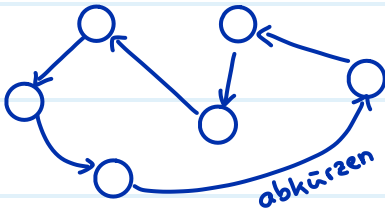
$$\text{weil } \ell(A) + \ell(B) = \text{OPT}_p$$

die kleinere Menge A oder B ist ein Matching mit Gewicht $\leq \frac{1}{2} \text{OPT}$



3. W = Eulertour mit den Kanten von MST u M

$$\ell(W) = \ell(\text{MST}) + \ell(M) \leq \text{OPT} + \frac{1}{2} \text{OPT} = 1.5 \text{OPT}$$



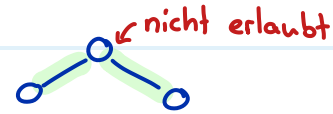
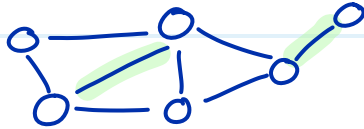
4. abkürzen (wie vorher)

$$\ell(C) \leq \ell(W) \leq 1.5 \text{OPT}$$

$\Rightarrow$ das ist ein 1.5-Approximationsalgorithmus

Matchings (wichtig)

- Ein Matching M ist eine Menge von Kanten $M \subseteq E$, sodass kein Knoten zu mehr als einer Matchingkante inzident ist



- Ein Knoten v wird von M **überdeckt**, wenn v zu einer Matchingkante inzident ist
- M ist **perfekt** $\Leftrightarrow$ alle Knoten v werden von M überdeckt

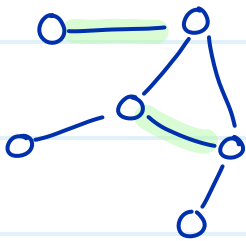
$$\Leftrightarrow |M| = \frac{|V|}{2}$$

- M ist **kardinalitätsmaximal** $\Leftrightarrow |M| \geq |M'|$ für alle Matchings M'

$\Leftrightarrow$ Es gibt kein Matching mit mehr Kanten

- M ist **inklusionsmaximal** $\Leftrightarrow M \cup \{e\}$ ist kein Matching für alle $e \in E \setminus M$

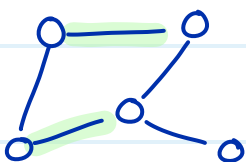
$\Leftrightarrow$ Es gibt kein Matching M' mit $M \subseteq M'$ und $|M| < |M'|$



perfekt? $\rightarrow$ Nein

kardinalitätsmaximal? $\rightarrow$ Nein

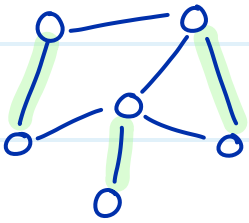
inklusionsmaximal? $\rightarrow$ Ja



perfekt? $\rightarrow$ Nein

kardinalitätsmaximal? $\rightarrow$ Ja

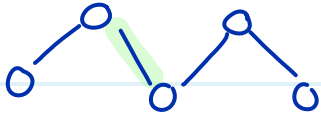
inklusionsmaximal? $\rightarrow$ Ja



perfekt? $\rightarrow$ Ja

kardinalitätsmaximal? $\rightarrow$ Ja

inklusionsmaximal? $\rightarrow$ Ja



perfekt? $\rightarrow$ Nein

kardinalitätsmaximal? $\rightarrow$ Nein

inklusionsmaximal? $\rightarrow$ Nein

Wie "gut" ist ein Matching?

$\mathcal{P}(E)$ = Menge aller Teilmengen von E

Matchings

inklusionsmaximal

kardinalitätsmaximal

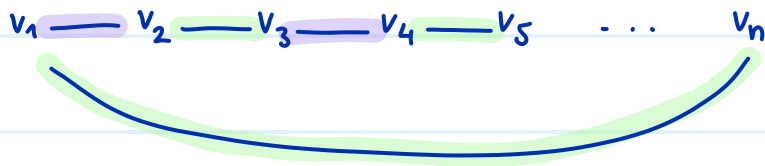
perfekt

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

$$M_1 \cap M_2 = \emptyset$$

- (a) **True or False:** If G has a Hamiltonian cycle, then G contains two disjoint perfect matchings.

Number of edges in Hamiltoncycle is even (because G is bipartite)



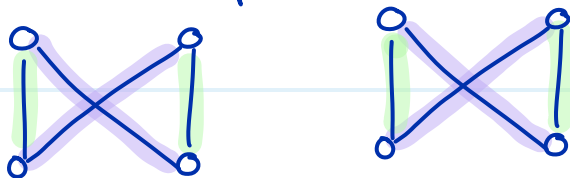
M_1 = even edges

M_2 = odd edges

$\Rightarrow$ True

- (b) Prove or disprove: If G contains two disjoint perfect matchings, then G has a Hamiltonian cycle.

Counterexample:



1. Let $G = (V, E)$ be a graph with a perfect matching.

- Prove or Disprove: G must be a connected graph.

Counterexample:



3. Let G be a graph with an odd number of vertices.

- Prove or Disprove: G cannot have a perfect matching.

Proof by contradiction. Assume M is a perfect matching in G

$$\Rightarrow |M| = \frac{|V|}{2} \quad \text{by definition}$$

$\Rightarrow |V|$ must be even, contradiction

5. Let T be a tree with more than 1 vertex.

- Prove or Disprove: T must have a perfect matching.

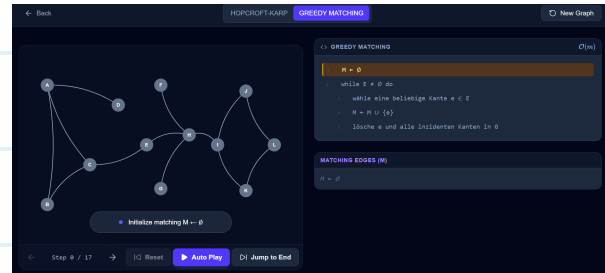
Counterexample:



Greedy Algorithm:

GREEDY-MATCHING (G) in $O(|E|)$

- 1: $M \leftarrow \emptyset$
- 2: **while** $E \neq \emptyset$ **do**
- 3: wähle eine beliebige Kante $e \in E$
- 4: $M \leftarrow M \cup \{e\}$
- 5: lösche e und alle inzidenten Kanten in G

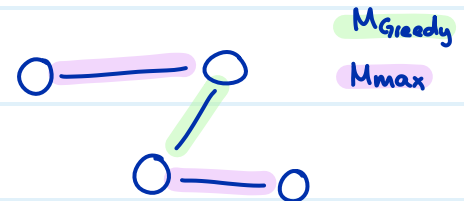


• M_{Greedy} ist inklusionsmaximal

• $|M_{\text{Greedy}}| \geq \frac{1}{2} |\text{kardinalitätsmaximales Matching } M_{\text{max}}|$

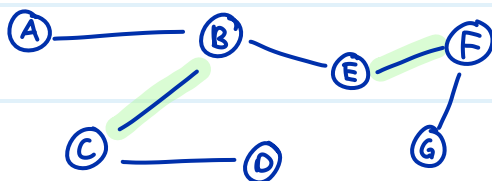
mindestens ein Endpunkt jeder Kante aus M_{max} muss von M_{Greedy}

überdeckt werden (sonst könnten wir sie zu M_{Greedy} hinzufügen $\rightarrow$ Contradiction)



Frage: Gegeben ein Matching, wie könnte man es verbessern?

Idee: Ein **M-augmentierender Pfad** ist ein Pfad der abwechselnd Kanten aus M und nicht aus M verwendet, und mit unüberdeckten Knoten beginnt und endet.



Augmentierende Pfade. $P_1 = (A, B, C, D)$

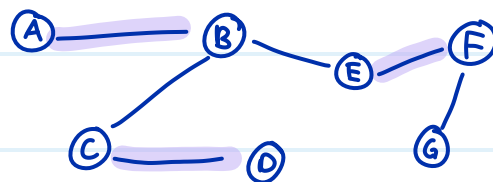
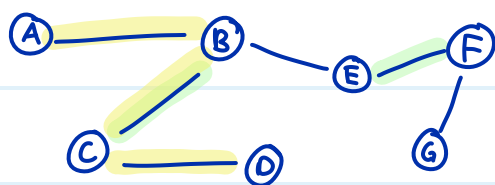
$P_2 = (D, C, B, E, F, G)$

$M = \{\{B, C\}, \{E, F\}\}$

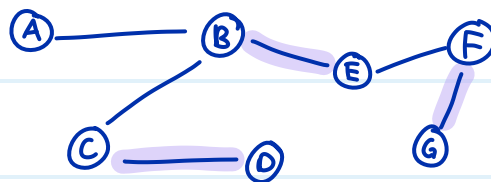
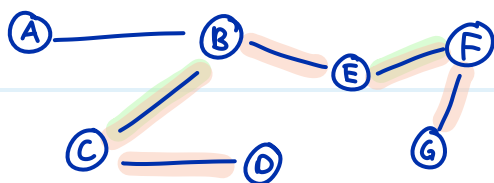
$$M \oplus P_1$$

$\oplus \hat{=} \text{XOR}$

$\hat{=} \text{addition modulo 2}$



$$M \oplus P_2$$



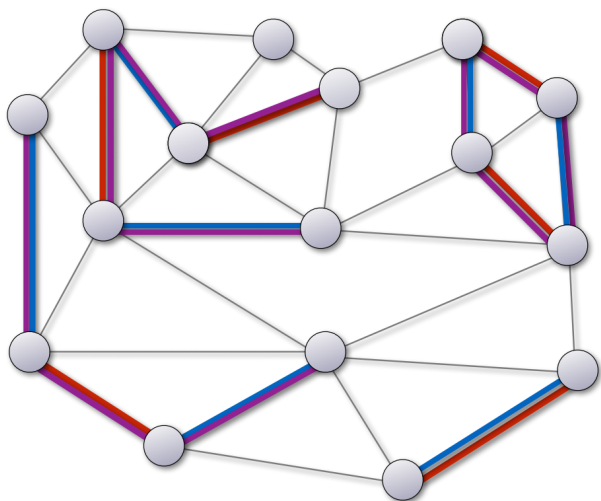
Durch Augmentieren wächst $|M|$ um 1

Achtung: P_2 ist kein augmentierender Pfad für $(M \oplus P_1)$

Satz von Berge Jedes nicht kardinalitätsmaximale Matching hat einen augmentierenden Pfad

Beweis: Sei M ein nicht kardinalitätsmaximales Matching

und $M_{\max}$ ein kardinalitätsmaximales Matching



Wir betrachten $M \oplus M_{\max}$

$\Rightarrow$ Jeder Knoten hat $\text{Grad} \leq 2$
(höchstens +1 von M und +1 von $M_{\max}$)

$\Rightarrow$ Graph ist in Pfade und Kreise zerlegbar

$\Rightarrow$ Alle diese Pfade und Kreise wechseln zwischen Kanten aus M und aus $M_{\max}$.

$\Rightarrow$ Kreise haben gerade Längen und benutzen deshalb gleich viele Kanten aus M wie aus $M_{\max}$

$\Rightarrow$ Weil $|M| < |M_{\max}|$ muss es mindestens einen Pfad P ungerader Länge geben, der mehr Kanten aus $M_{\max}$ als aus M verwendet

$\Rightarrow$ Die Endpunkte von P werden von $M_{\max}$ überdeckt aber nicht von M

$\Rightarrow P$ ist ein augmentierender Pfad



Algorithmus findet ein kardinalitätsmaximales Matching

Input: Graph $G = (V, E)$

Output: maximales Matching M (maximal = kardinalitätsmaximal)

Starte mit $M = \emptyset$.

repeat

- Suche augmentierenden Pfad P .
- **if** kein solcher Pfad existiert **then return** M .
- **else** $M := M \oplus P$.

• Laufzeit $O(|V| \cdot |E|)$, Beweis dazu ist nicht Vorlesungsstoff

• nächste Woche mehr dazu

Satz von Hall.

Ein bipartiter Graph $G = (A \uplus B, E)$ hat ein Matching M der Grösse $|M| = |A|$

$\Leftrightarrow |X| \leq |N(X)|$ für alle Teilmengen $X \subseteq A$

(Beweis dazu ist weniger wichtig als z.B. Hopcroft-Karp oder Dirac)

Exercise 1: The Card Dealer's Dilemma

Take a standard deck of 52 playing cards and deal them face up into 13 piles of 4 cards each.

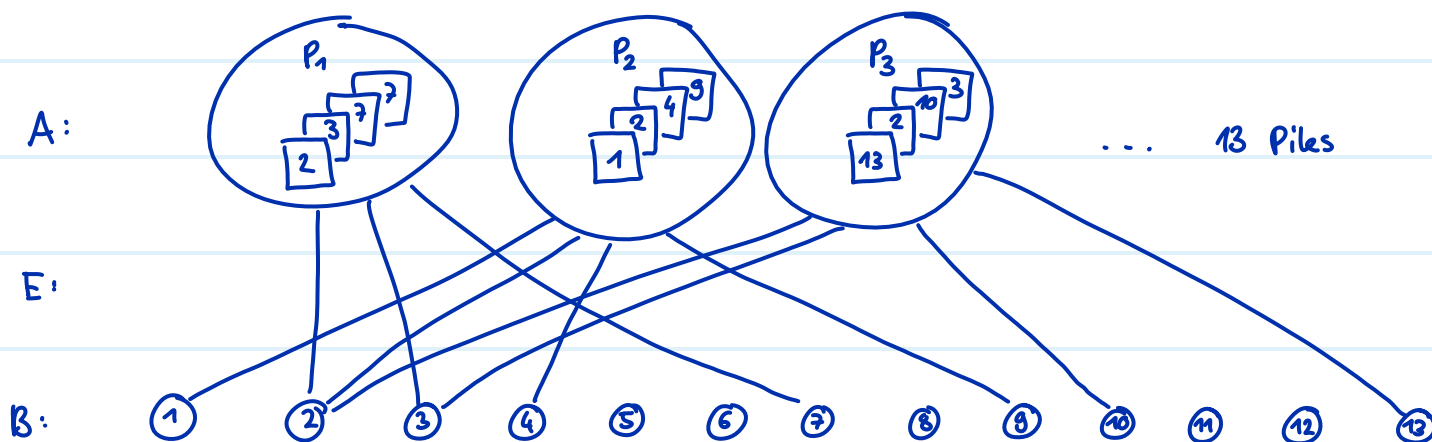
Question: Prove that it is always possible to pick exactly one card from each of the 13 piles such that your 13 chosen cards contain exactly one of every rank (one Ace, one 2, one 3, ..., one King).



Sei $G = (A \uplus B, E)$ wobei A einen Knoten für jeden der 13 Piles enthält,

und $B = \{1, 2, \dots, 13\}$

und $\{a, b\} \in E$ falls die Karte $b \in B$ im Stapel $a \in A$ ist



Sei $X \subseteq A$ beliebig

$\Rightarrow X$ enthält $4|X|$ viele Karten

$\Rightarrow N(X)$ enthält mindestens $|X|$ viele Zahlen, weil jede Zahl höchstens 4 mal vorkommt in X (im Kartenset gibt es keine Zahl, die mehr als 4 mal vorkommt)

$\Rightarrow \forall X \subseteq A: |X| \leq |N(X)|$

Hall
 $\Rightarrow$ es gibt ein Matching M mit $|M| = |A|$

$\Rightarrow$ falls $\{a, b\} \in M$ dann wählen wir die Zahl b aus Stapel a

Zu zeigen: Jeder k -reguläre bipartite Graph $G = (A \uplus B, E)$ hat ein perfektes Matching

Beweis: Sei $X \subseteq A$ beliebig

$\Rightarrow$ Anzahl Kanten die X verlassen und zu $N(X)$ gehen

$$= \sum_{v \in X} \deg(v) \quad (\text{weil } G \text{ bipartit ist})$$

$$= \sum_{v \in X} k$$

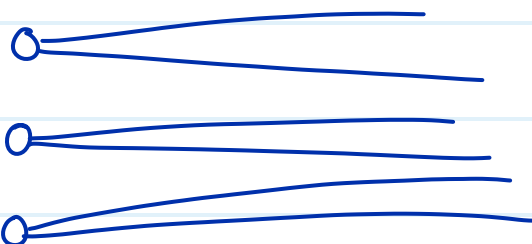
$$= |X| k$$

$\Rightarrow |N(X)| \geq |X|$ weil $|X| k$ viele Kanten nach $N(X)$ gehen, und jeder Knoten aus $N(X)$ höchstens zu k dieser Kanten inzident ist

$X \subseteq A$

$N(X) \subseteq B$

für $k=2$



$\Rightarrow$ Es gibt ein Matching M der Grösse $|M| = |A|$ (Satz von Hall)

$\Rightarrow M$ ist perfekt weil $|M| = |A| = \frac{|A| + |B|}{2} = \frac{|V|}{2}$

$$(|A| = |B| \text{ weil } \sum_{v \in A} \deg(v) = \sum_{v \in B} \deg(v))$$

Organisatorisches: • Peergrading ist Einzelarbeit

- 1 Woche Zeit fürs abgeben der eigenen Lösung
- 5 Tage danach Zeit für das Peergrading