SHORT RECAP: Matchings

Matching := Kantenmenge $M \subseteq E$, kein Knoten des Graphen zu mehr als einer Kante aus M inzident ist

Knoten v wird von M überdeckt, falls es eine Kante $e \in M$ gibt, die v enthält.

perfektes Matching: Wenn jeder Knoten durch genau eine Kante des Matchings überdeckt wird ($|M| = \frac{|V|}{2}$)

Ein Matching M ist...
- ... inklusionsmaximal, falls $M \cup \{e\}$ kein Matching $\forall e \in E \setminus M$
- ... kardinalitätsmaximal, falls $|M| \geq |M'|$ für alle Matchings M' in G

Matchings in Graphen finden

GREEDY-MATCHING (G)

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

Laufzeit: $O(|E|)$

Satz 1.47. Der Algorithmus GREEDY-MATCHING bestimmt in Zeit $O(|E|)$ ein inklusionsmaximales Matching M_{Greedy} für das gilt:

$$|M_{Greedy}| \geq \frac{1}{2}|M_{max}|,$$

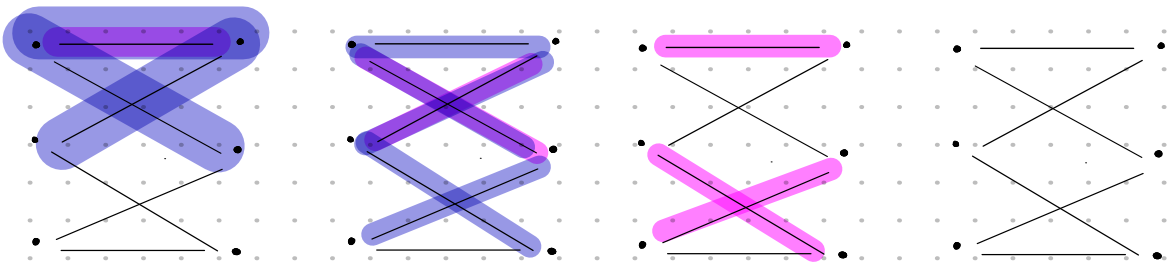
wobei M_{max} ein kardinalitätsmaximales Matching sei.

Konzept der augmentierenden Pfade

Sei M ein Matching in G.
Der M-augmentierende Pfad P...

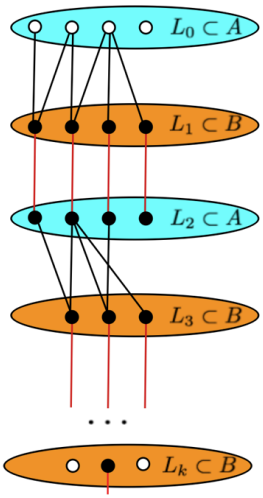
- ... besteht abwechselnd aus Kanten $E \setminus M$ und M
- ... beginnt und endet in einem von M unüberdeckten Knoten

Wir können M nun zu M' vergrößern: $M' = M \oplus P$



AUGMENTING_PATH $(G = (A \uplus B, E), M)$

1: $L_0 := \{\text{unüberdeckte Knoten in } A\}$
2: Markiere alle Knoten aus L_0 als besucht.
3: **if** $L_0 = \emptyset$ **then**
4: **return** M ist maximal
5: **for all** $i = 1$ **to** n **do**
6: **if** i ungerade **then**
7: $L_i := \{\text{unbesuchte Nachbarn von } L_{i-1} \text{ via Kanten in } E \setminus M\}$
8: **else**
9: $L_i := \{\text{unbesuchte Nachbarn von } L_{i-1} \text{ via Kanten in } M\}$
10: Markiere alle Knoten aus L_i als besucht.
11: **if** L_i enthält unüberdeckten Knoten v **then**
12: Finde Pfad P von L_0 nach v durch backtracking
13: **return** P // *terminiert Algorithmus*
14: **return** M ist schon maximal



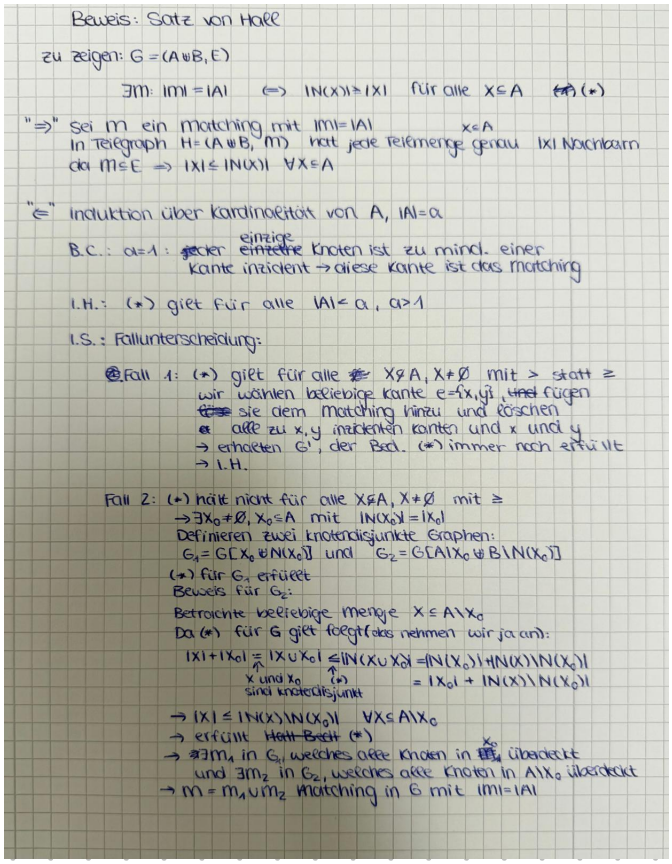
Satz 1.48 (Satz von Berge). Ist M ein Matching in einem Graphen $G = (V, E)$, das nicht kardinalitätsmaximal ist, so existiert ein augmentierender Pfad zu M.

Satz von Hall

Satz 1.52 (Satz von Hall, Heiratssatz). Für einen bipartiten Graphen $G = (A \uplus B, E)$ gibt es genau dann ein Matching M der Kardinalität $|M| = |A|$, wenn gilt

$$|N(X)| \geq |X| \quad \text{für alle } X \subseteq A. \tag{1.1}$$

Da viele gefragt haben: Erklärung des Beweises an der Tafel



Beweis: Satz von Hall

zu zeigen: $G = (A \uplus B, E)$

$\exists M: |M| = |A| \iff |N(X)| \geq |X|$ für alle $X \subseteq A$ (*)

"$\Rightarrow$" Sei M ein Matching mit $|M| = |A|$. In Teilgraph $H = (A \uplus B, M)$ hat jede Teilmenge $X \subseteq A$ genau $|X|$ Nachbarn da $M \subseteq E \Rightarrow |X| \leq |N(X)| \quad \forall X \subseteq A$

"$\Leftarrow$" Induktion über Kardinalität von A, $|A| = \alpha$

B.C.: $\alpha = 1$: jeder einzige einzelne Knoten ist zu mind. einer Kante inzident → diese Kante ist das Matching

I.H.: (*) gilt für alle $|A| < \alpha$, $\alpha > 1$

I.S.: Fallunterscheidung:

Fall 1: (*) gilt für alle $X \subsetneq A$, $X \neq \emptyset$ mit > statt ≥
wir wählen beliebige Kante $e = \{x, y\}$, und fügen sie dem Matching hinzu und löschen alle zu x, y inzidenten Kanten und x und y
→ erhalten G', der Bed. (*) immer noch erfüllt
→ I.H.

Fall 2: (*) hält nicht für alle $X \subsetneq A$, $X \neq \emptyset$ mit ≥
$\rightarrow \exists X_0 \neq \emptyset, X_0 \subseteq A$ mit $|N(X_0)| = |X_0|$
Definieren zwei knotendisjunkte Graphen:
$G_1 = G[X_0 \cup N(X_0)]$ und $G_2 = G[A \setminus X_0 \cup B \setminus N(X_0)]$
(*) für G_1 erfüllt
Beweis für G_2:
Betrachte beliebige Menge $X \subseteq A \setminus X_0$
Da (*) für G gilt folgt(dass nehmen wir ja an):

$|X| + |X_0| = |X \cup X_0| \leq |N(X \cup X_0)| = |N(X_0) \cup N(X) \setminus N(X_0)|$
(X und X_0 sind knotendisjunkt; (*))
$= |X_0| + |N(X) \setminus N(X_0)|$

$\rightarrow |X| \leq |N(X) \setminus N(X_0)| \quad \forall X \subseteq A \setminus X_0$
→ erfüllt Hall-Bed. (*)
→ $\exists M_1$ in G_1, welches alle Knoten in X_0 überdeckt und $\exists M_2$ in G_2, welches alle Knoten in $A \setminus X_0$ überdeckt
→ $M = M_1 \cup M_2$ Matching in G mit $|M| = |A|$

Aufgabe:

Die exklusive Käse Kochschule in einem abgelegenen Schweizer Bergdorf hat $2k$ Schüler. Ironischerweise haben heute alle Schüler aus versehen ihre Brotdosen mit dem Snack fürs Mittagessen zu Hause vergessen. Die entscheiden sich stattdessen ein Festmahl zu kochen: ein legendäres k-Gang Menü. Sie haben schon 15 Minuten ihrer einstündigen Mittagspause darauf verwendet, zu planen, welche Gerichte in dem extravaganten Menü enthalten sein sollen und beschliessen nun schnell mit dem Kochen zu beginnen.

Jedes der gewählten Gerichte ist so kompliziert, dass es die volle Aufmerksamkeit von 2 Köchen braucht. Andererseits, da die Köche sich ja noch in der Ausbildung befinden, kann jeder Schüler nur eine Teilmenge der Gerichte zubereiten. Damit das Menü auch gelingt müssen beide Köche wissen, wie das jeweilige Gericht zubereitet wird.

Jeder der $2k$ Köche hat eine Liste der Gerichte, die er/sie zubereiten kann zur verfügung gestellt. Jetzt ist die Frage, wie die Köche sich in Paare aufteilen sollen, sodass sie alle k Gerichte kochen können. Oder ist dies etwa unmöglich?

(a) Modellieren Sie das Problem mithilfe eines Graphen G, sodass ein Aufteilung der Köche wie oben beschrieben existiert genau dann wenn G ein perfektes Matching hat.

(b) Nehmen Sie an, dass es möglich ist alle Gerichte zu kochen. Zeigen Sie, dass es für jede Menge an Gerichten X wenigstens $2|X|$ Köche gibt die wissen, wie man mindestens eines dieses Gericht zubereitet.

(c) Umgekehrt, nehmen Sie an, dass es für jede Menge an Gerichten X wenigstens $2|X|$ Köche gibt die wissen, wie man mindestens eines dieses Gericht zubereitet. Zeigen Sie, dass es möglich ist, alle Gerichte zuzubereiten.

Aufgabe 2 – *Lateinisches Rechteck*

Ein lateinisches $r \times n$-Rechteck ($r \leq n$) ist eine Anordnung der Zahlen $1, \ldots, n$ in r Zeilen und n Spalten, so dass in jeder Zeile und jeder Spalte jede Zahl höchstens einmal vorkommt. Ein lateinisches n-Quadrat ist ein lateinisches $n \times n$-Rechteck.

$$\begin{bmatrix} 4 & 1 & 2 & 3 \\ 2 & 3 & 1 & 4 \\ 3 & 2 & 4 & 1 \end{bmatrix}$$

Beispiel eines lateinischen 3×4-Rechtecks.

Angenommen wir haben ein lateinisches $r \times n$-Rechteck mit $r < n$ gegeben. Wir wollen sehen, ob wir es zu einem lateinischen n-Quadrat erweitern können. Das heisst, wir wollen $n - r$ weitere Zeilen mit Zahlen aus $1, \ldots, n$ zu dem Rechteck hinzufügen, ohne eine Spalte oder eine Zeile in der eine Zahl mehrfach vorkommt zu erhalten.

(a) Angenommen wir haben ein lateinisches $r \times n$-Rechteck wie oben beschrieben. Wir wollen zeigen, wann man dieses zu einem $(r+1) \times n$-Rechteck erweitern kann. Beschreiben Sie, wie man dieses Problem mit einem bipartiten Graphen $G = (A \uplus B, E)$ modellieren kann und zeigen Sie, dass die Erweiterung genau dann möglich ist, wenn G ein perfektes Matching hat.

(b) Zeigen Sie, dass der in (a) konstuierte Graph regulär ist. Das heisst, zeigen Sie, dass dass es eine ganze Zahl k gibt, sodass alle Knoten (sowohl die Knoten in A als auch die Knoten in B) Grad genau k haben.

(c) Benutzen Sie ihr Ergebnis aus (b) um zu beschreiben, welche $r \times n$-Rechtecke man zu einem lateinischen n-Quadrat erweitern kann.

(d) Geben Sie einen Algorithmus an, der als Eingabe ein Lateinisches $r \times n$-Rechteck nimmt und es zu einem lateinischen n-Quadrat erweitert (falls ein solches existiert) und ansonsten "Nicht möglich" ausgibt.

Hinweis: In (a) kann es hilfreich sein, die Knoten in A mit 'Spalte 1', 'Spalte 2',... zu labeln und die Knoten aus B mit 'Nummer 1', 'Nummer 2',.... Was sind dann die Kanten? Was ist die entsprechende Bedeutung eines perfekten Matchings in Bezug auf das lateinische Quadrat?

Hinweis: Für (d) können Sie die Ergebnisse der vorherigen Teilaufgaben verwenden. Beachten Sie aber, dass zu einem Algorithmus auch immer eine Laufzeitanalyse und einen Korrektheitsbeweis gehören – auch wenn dies nur ein kurzer Hinweis auf eine der vorherigen Teilaufgaben ist!

Algorithmus von Hopcroft & Karp

MAXIMAL_MATCHING $(G = (A \oplus B, E))$ (Hopcroft und Karp)

1: $M := \{e\}$ für irgendeine Kante $e \in E$.
2: **while** es gibt noch augmentierende Pfade **do**
3: $k :=$ Länge eines kürzesten augmentierenden Pfades
4: Finde eine inklusionsmaximale Menge S von paarweise disjunkten augmentierenden Pfaden der Länge k.
5: **for all** P aus S **do**
6: $M := M \oplus P$. // *augmentiere entlang der Pfade aus* S
7: **return** M

Satz 1.49. Der Algorithmus von Hopcroft und Karp durchläuft die while-Schleife nur $O(\sqrt{|V|})$ Mal. Er berechnet daher ein maximales Matching in einem bipartiten Graphen in Zeit $O(\sqrt{|V|} \cdot (|V| + |E|))$.

Gabow's Algorithmus

Satz 1.54. Ist $G = (V, E)$ ein 2^k-regulärer bipartiter Graph, so kann man in Zeit $O(|E|)$ ein perfektes Matching bestimmen.

→ Eulertour W finden
→ Löschen von jeder zweiter Kante in W
→ 2^{k-1}-regulären Graphen
→ rekursiv bis 1-regulären Graphen
→ perfektes Matching

$$\sum_{i=1}^{k} C \cdot \frac{E}{2^{i-1}} \leq \sum_{i=0}^{\infty} C \cdot \frac{E}{2^i} = C \cdot E \cdot \sum_{i=0}^{\infty} \left(\frac{1}{2}\right)^i = C \cdot E \cdot \frac{1}{1-\frac{1}{2}} = 2C \cdot E = O(|E|)$$

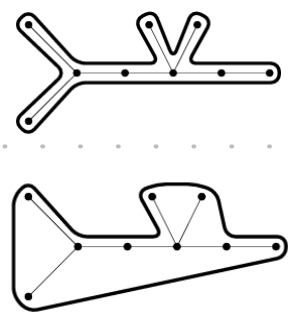
Satz von Frobenius

Satz 1.53. Sei $G = (A \uplus B, E)$ ein k-regulärer bipartiter Graph. Dann gibt es $M_1, \ldots, M_k$ so dass $E = M_1 \uplus \ldots \uplus M_k$ und alle M_i, $1 \leq i \leq k$, perfekte Matchings in G sind.

Beweisidee:
Betrachte beliebige Menge $X \subseteq A$ und definiere $G'[X \uplus N(X)]$.
Da G k regulär gilt für alle $v \in X$ $\deg(v) = k$ und alle $w \in N(X)$ $\deg(w) \leq k$ in G'.
Aufgrund der Bipartitheit von G' gilt: $\sum_{v \in X} \deg(v) = \sum_{w \in N(X)} \deg(w)$
→ $|X| \leq |N(X)|$
→ nach Satz von Hall existiert Matching M_1 in G' und somit auch in G
Entfernen wir nun M_1 aus G_1 so erhalten wir einen $(k-1)$-regulären, bipartiten Graphen.
Nun können wir den ersten Schritt so oft anwenden, bis wir einen 1-regulären Graphen enthalten.
Auch dieser hat ein perfektes Matching M_k.

TRAVELLING SALESMAN PROBLEM (TSP)

Satz: Für das metrische TSP gibt es einen 2-Approximationsalgorithmus mit Laufzeit $O(n^2)$.



1. Bestimme MST T
2. Verdopple alle Kanten von T
3. Bestimme Eulertour W
4. Durchlaufe W, mit Abkürzungen, so dass jeder Knoten nur einmal besucht wird $\Rightarrow$ Hamiltonkreis C

$\ell(T) \leq opt(K_n, \ell)$

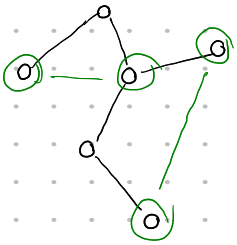
$2\ell(T) \leq 2 \cdot opt(K_n, \ell)$

$\ell(W) = 2\ell(T)$

$\ell(C) \leq \ell(W) = 2 \cdot \ell(T) \leq 2 opt(K_n, \ell)$

↑ Dreiecksungleichung

Satz 1.51. Für das METRISCHE TRAVELLING SALESMAN PROBLEM gibt es einen 3/2-Approximationsalgorithmus mit Laufzeit $O(n^3)$.



1. Bestimme MST T

$\ell(T) \leq opt(K_n, \ell)$

2. X := Knoten mit ungeradem Grad in T
   Bestimme min. Matching M für X

$\ell(M) \leq \frac{1}{2} opt(K_n, \ell)$

3. Bestimme Eulertour W

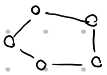
$\ell(W) = \ell(T) + \ell(M) \leq \frac{3}{2} opt(K_n, \ell)$

4. Kürze Eulertour W ab und finde Hamiltonkreis C

$\ell(C) \leq \ell(W) = \ell(T) + \ell(M) \leq \frac{3}{2} opt(K_n, \ell)$

MINIQUIZVORBEREITUNG

1) Angenommen, G ist ein Graph, der eine Eulertour enthält, und die Anzahl Knoten von G ist gerade. Dann enthält G ein perfektes Matching.
→ Falsch 

2) Der Satz von Dirac impliziert, dass Graphen, welche einen Knoten mit weniger als $\frac{n}{2}$ Nachbarn besitzen, keinen Hamiltonkreis haben.
→ Falsch, 

3) Sei M ein Matching in einem Graphen G. Dann gilt für jeden M-augmentierenden Pfad, dass er Länge mind. $\frac{|M|}{2}$ hat.
→ Falsch

4) Für das metrische TSP-Problem gibt es einen 2-Approximationsalgorithmus, aber keinen 4-Approximationsalgorithmus.
→ Falsch

5) Für einen bipartiten Graphen $G=(A \dot\cup B, E)$ gilt genau dann $|N(X)| \geq |X|$ für alle $X \subseteq A$, wenn es ein Matching M der Kardinalität $|M|=|A|$ gibt.
→ Wahr, Satz von Hall

6) Jeder Baum ist bipartit.
→ Wahr

7) Sei G ein vollständiger Graph mit Gewichtsfunktion auf den Kanten, sodass das Gewicht auf jeder Kante mind. 1 ist. Angenommen, dass eine optimale TSP-Route in G Kosten 10 hat. Geben Sie eine bestmögliche obere Schranke auf die Kosten eines minimalen Spannbaums in G.
→ 9

8) In der Vorlesung haben wir einen Algorithmus für eine 2-Approximation des TSP gesehen, der Laufzeit $O(n^2)$ hat.
→ Falsch, nur für metrisches TSP

9) Wir nehmen an, dass G ein inklusionsmax. Matching M der Grösse 8 enthält. Welche der folgenden Optionen sind mögliche Grössen eines kardinalitätsmax. Matchings in G?
4 **8** **13** 20

10) Jeder k-reguläre Graph mit $k \geq 1$ enthält ein perfektes Matching.
→ Falsch  , gilt für bipartite Graphen

11) Sei G ein Graph mit einem nicht-perfekten Matching, zu dem es keinen augmentierenden Pfad gibt. Dann gibt es kein perfektes Matching in G.
→ Wahr, M schon kardinalitätsmaximal

12) Ist ein Matching inklusionsmax., gibt es keine augmentierenden Pfade mehr.
→ Falsch