



Technische
Universität
Braunschweig



Parametrisierte Algorithmen

Arne Schmidt

Kapitel 2.3 – Vertex Cover und Schranken

Untere Schranken für Vertex Cover

Aufgabe:

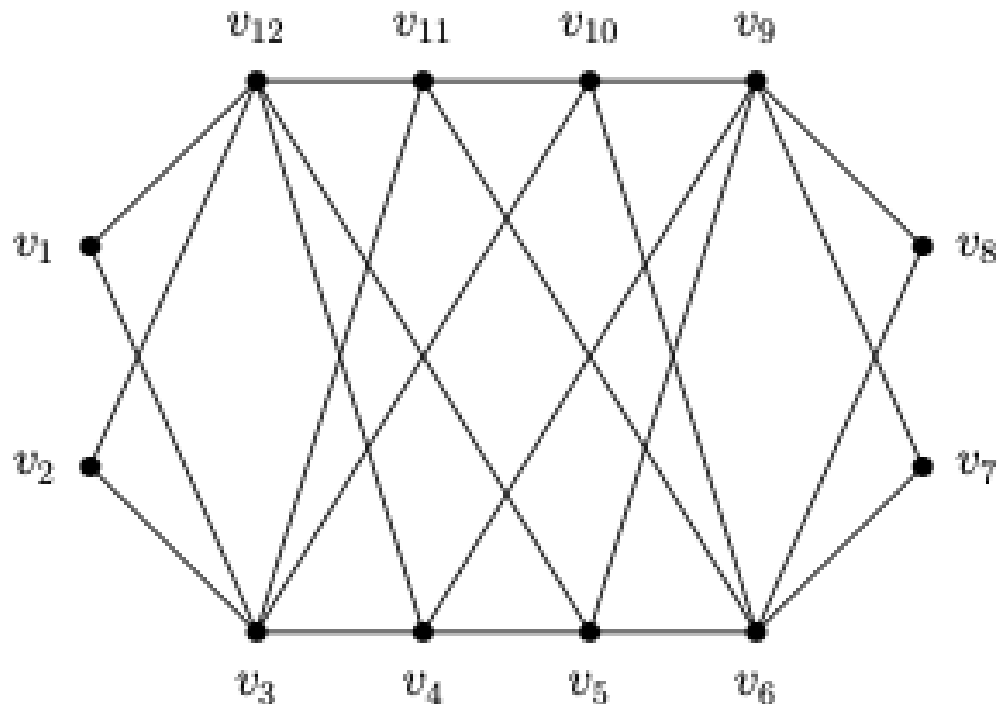
Zerteile den Graphen in möglichst einfache disjunkte Teile, um zu zeigen, dass mind. 6 Knoten für ein Vertex Cover benötigt werden.

Frage:

Welche Strukturen eignen sich?

- Cliquen
- Kreise
- Pfade

Was ist die einfachste Struktur?

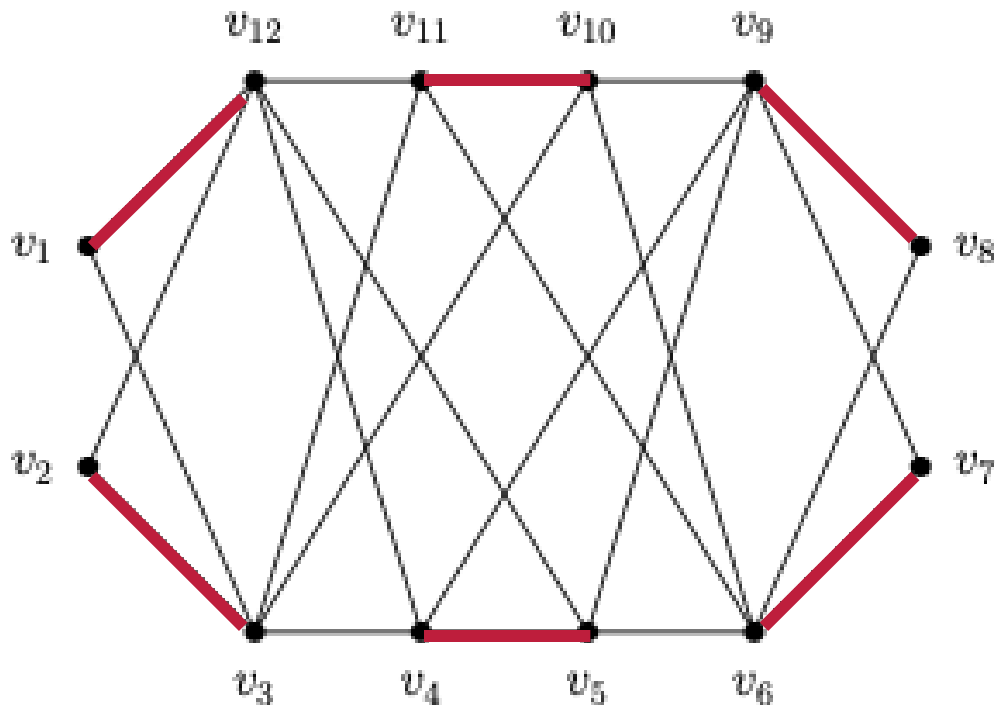


Unabhängige Kanten

Finde eine möglichst große Menge an unabhängigen Kanten $M \subseteq E$.

Für jede Kante $e \in M$ muss ein Knoten im VC liegen.

Damit ist das kleinste VC mind. so groß wie $|M|$!



Kapitel 2.3.1 – Matchings

Matchings

Definition 2.12 (Matchings)

Sei $G = (V, E)$ ein Graph

- (1) Eine Kantenmenge $M \subseteq E$ heißt **Matching**, wenn $e \cap f = \emptyset$ für je zwei Kanten $e, f \in M$ gilt.
- (2) Kanten in M heißen **unabhängig**.
- (3) Ein Matching M heißt **perfektes Matching**, wenn $2|M| = |V|$ gilt.
- (4) Ein Matching heißt **inklusionsmaximal** (engl. maximal), wenn $M \cup \{e\}$ für jede Kante $e \in E \setminus M$ kein Matching ist.
- (5) Ein Matching heißt **(kardinalitäts-)maximal** (engl. maximum), wenn kein Matching M' mit $|M| < |M'|$ existiert.

Matchings

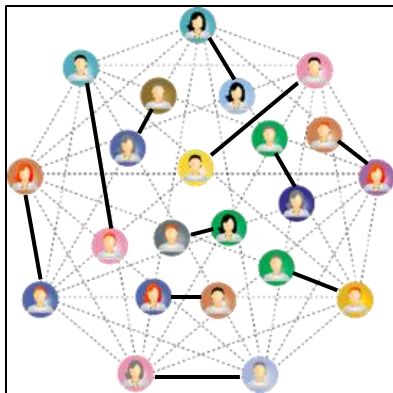
Problem 2.13: Maximales Matching (engl. Maximum Matching)

Gegeben:

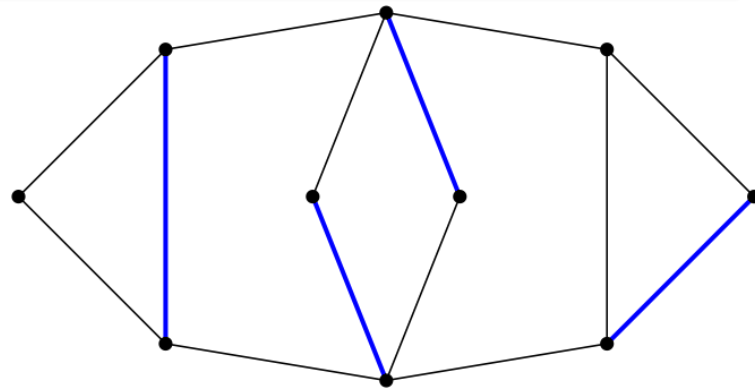
Graph $G = (V, E)$

Gesucht:

Maximales Matching M .

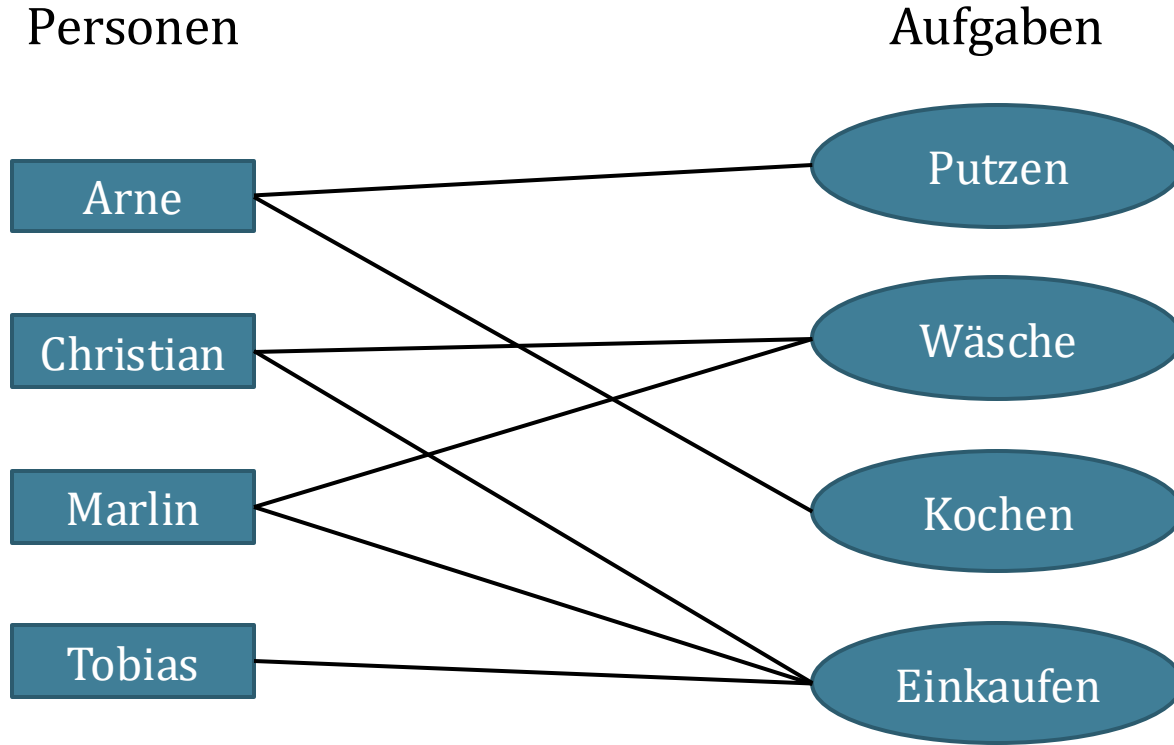


Perfektes Matching



Inklusionsmaximales Matching
Gibt es ein besseres?

Aufgabenverteilung



Kapitel 2.3.2 – Bipartite Graphen

Bipartite Graphen

Definition 2.14 (Bipartite Graphen)

Sei $G = (V, E)$ ein Graph

(1) G heißt **bipartit**, wenn sich V disjunkt in zwei Knotenmengen V_1, V_2 zerlegen lässt, sodass $u \in V_1$ und $v \in V_2$ für jede Kante $\{u, v\} \in E$ gilt.

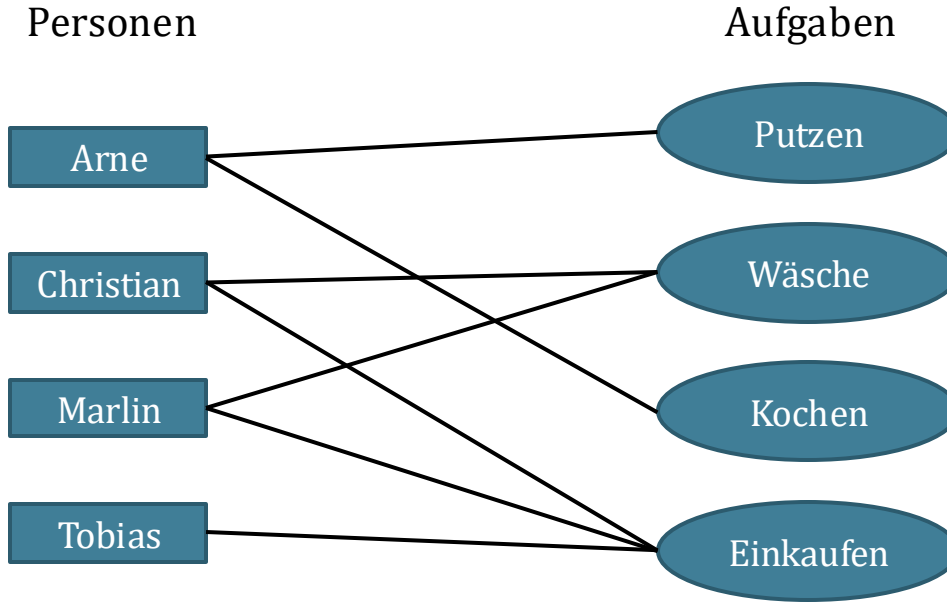
(2) Mit $K_{a,b}$ wird der **vollständig bipartite Graph** bezeichnet, d.h. $|V_1| = a, |V_2| = b$ und zwischen jedem Paar $(u, v) \in V_1 \times V_2$ verläuft eine Kante.

Satz 2.15

Sei G ein Graph. Dann sind folgende Aussagen äquivalent.

1. G ist bipartit.
2. G ist zweifärbbar.
3. G enthält keine Kreise ungerader Länge.

Aufgabenverteilung



Möchten sich einer Aufgabe zuweisen.

Möchten einer Person zugewiesen werden.

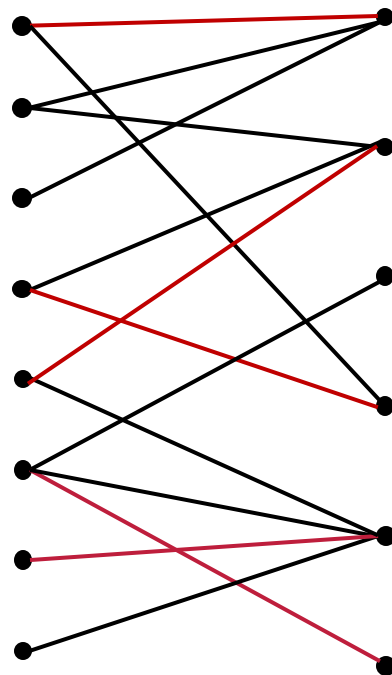
Weiteres Beispiel

Wie groß ist ein kleinstes Vertex Cover?

Für jede Matchingkante benötigen wir ein Knoten im Vertex Cover.

Beobachtung:

Nachbarn von ungematchten Knoten sind gematcht.
Nimm diese in das VC auf!



Weiteres Beispiel

Wie groß ist ein kleinstes Vertex Cover?

Für jede Matchingkante benötigen wir ein Knoten im Vertex Cover.

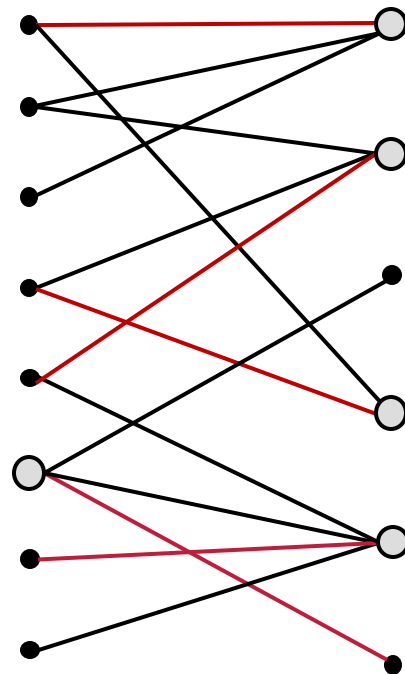
Beobachtung:

Nachbarn von ungematchten Knoten sind gematcht.
Nimm diese in das VC auf!

Fragen:

Warum ist damit immer maximal ein Endknoten jeder
Matchingkante im VC?

Was bleibt für ein Restgraph mit nicht überdeckten Kanten über?



Noch ein weiteres Beispiel

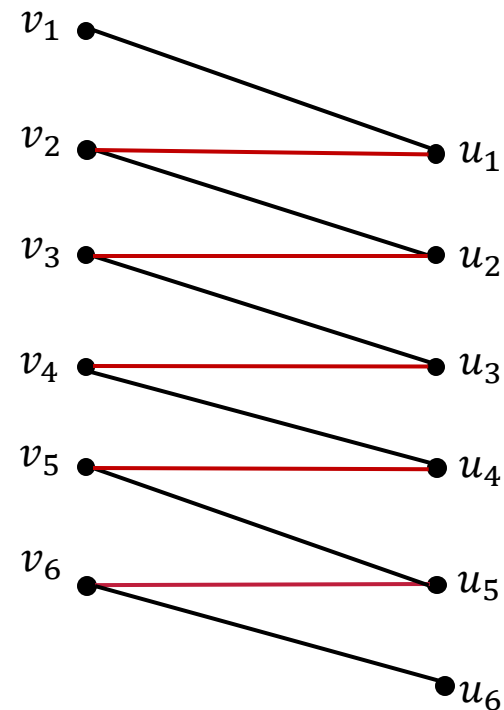
- Wegen v_1 muss u_1 ins Vertex Cover
- Wegen v_2 muss dann u_2 ins Vertex Cover
- Wegen v_3 muss dann u_3 ins Vertex Cover
- Wegen v_4 muss dann u_4 ins Vertex Cover
- Wegen v_5 muss dann u_5 ins Vertex Cover
- Wegen v_6 muss dann u_6 ins Vertex Cover

Aber u_6 ist doch gar nicht gematcht?!

Beobachtung:

Das Matching ist nicht maximal!

Drehe Matching- und Nicht-Matching-Kanten um!



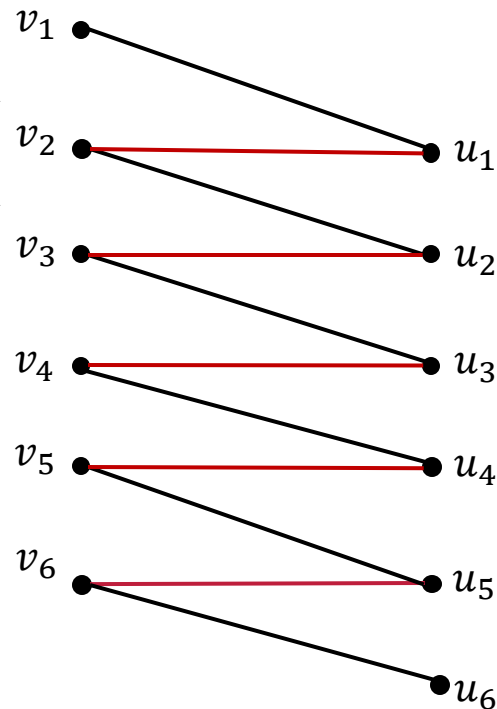
Das gleiche Spiel mit Matchings

- Wenn v_1 mit u_1 matchen möchte, muss v_2 jemand anderes haben
- Wenn v_2 mit u_2 matchen möchte, muss v_3 jemand anderes haben
- ...
- Wenn v_5 mit u_5 matchen möchte, muss v_6 jemand anderes haben
- Oh! v_6 kann mit u_6 matchen!

Wir haben einen Pfad gefunden, der abwechselnd Matching- und Nicht-Matching-Kanten verwendet.

Dieser muss ungerade Länge besitzen!
Flippen dieser Kanten vergrößert das Matching.

Dieser Pfad wird auch **augmentierender Pfad** genannt.



Vertex Cover und Matchings

Lemma 2.16

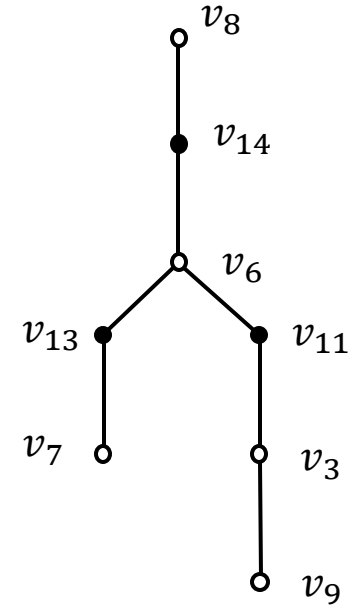
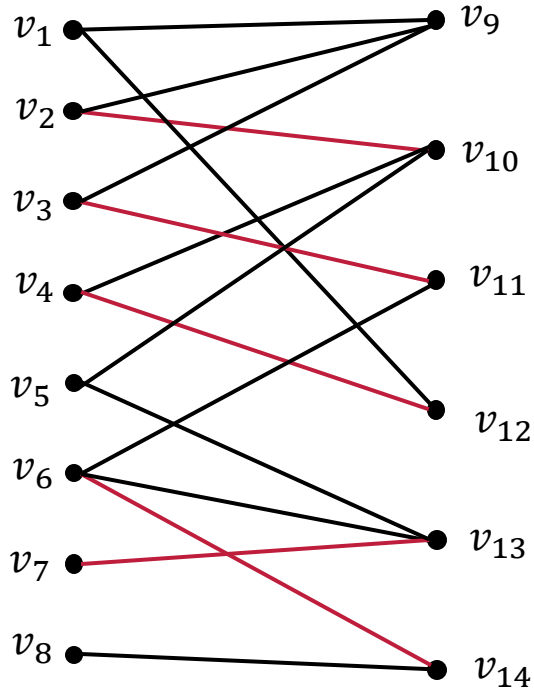
Sei G ein Graph. Für jedes Matching M und jedes Vertex Cover C gilt $|M| \leq |C|$.

Satz 2.17 (König-Egerváry)

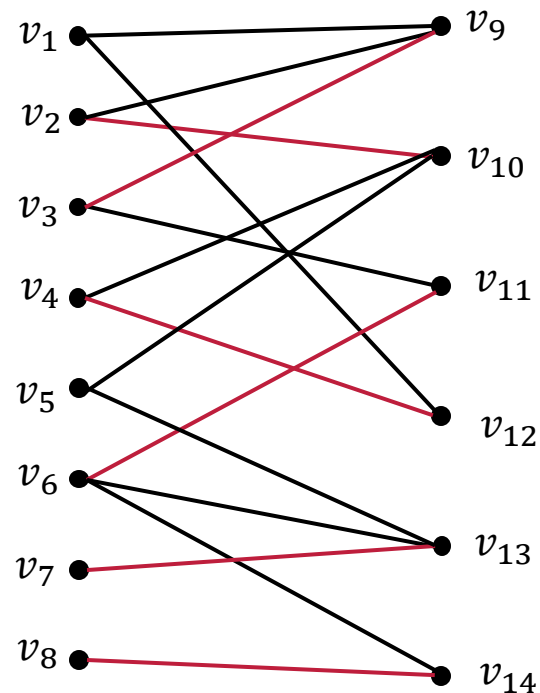
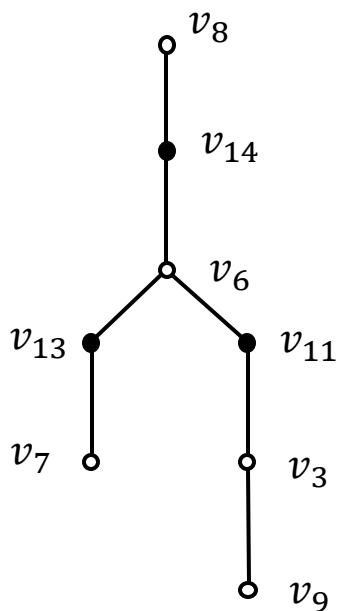
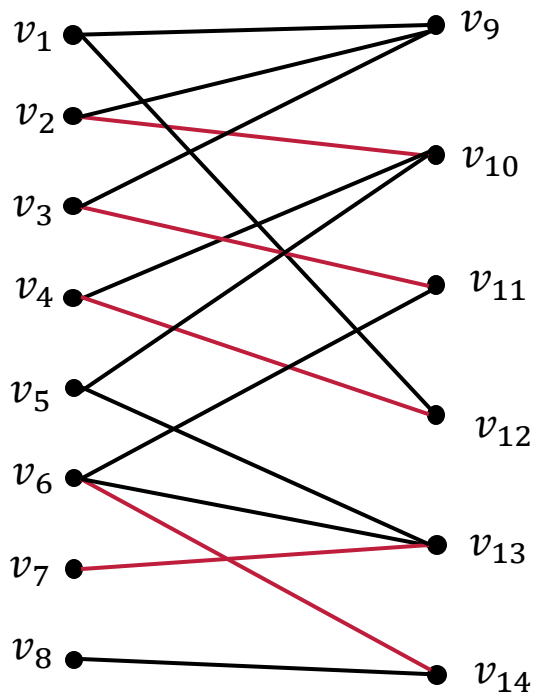
Sei G ein bipartiter Graph. Die Größe eines minimalen Vertex Covers in G entspricht der Größe eines maximalen Matchings in G .

Skizze: Spiele das Spiel wie eben und zeige, dass für jede Matchingkante genau 1 Endknoten im VC liegt oder andernfalls das Matching erweitert werden kann.

Suche nach augmentierenden Pfaden



Erweitern von Matchings



Matching für bipartite Graphen

Algorithmus 2.18

Eingabe:

Bipartiter Graph $G = (V, E)$

Ausgabe:

Maximales Matching M

```
1.  Function BIPARTITEMATCHING( $G$ )
2.      Set  $M := \emptyset$ 
3.      for  $r \in V_1$  mit  $r$  ungematcht do
4.          Setze  $T := (\{r\}, \emptyset)$  und  $W(T) := \{r\}$ 
5.          While (es ex. Kante  $\{v, w\} \in E$  mit  $v \in W(T)$  und  $w \notin V(T)$ )
6.              If  $w$  ist ungematcht then
7.                  Benutze  $w$  um augmentierenden Pfad zu bilden.
8.                  Augmentiere  $M$ 
9.                  If es ex. Kein ungematchter Knoten mehr then
10.                     Return perfektes Matching  $M$ 
11.                 else
12.                     Gehe zu Zeile 3.
13.             else
14.                 Sei  $\{w, z\}$  Matchingkante an  $w$ .
15.                 Füge  $w, z$  zu  $V(T)$  hinzu und  $z$  zu  $W(T)$ 
16.                 Füge  $\{v, w\}, \{w, z\}$  zu  $E(T)$  hinzu.
17.      Return  $M$ 
```

Analyse

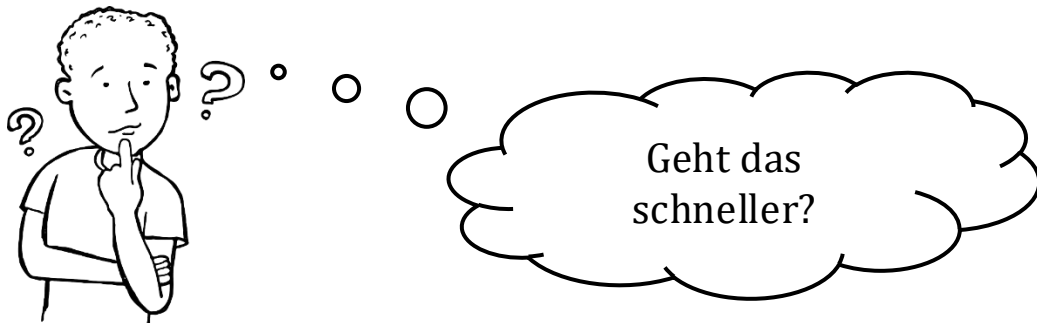
Satz 2.19

Algorithmus 2.18 löst Problem 2.13 (Max Matching) korrekt in Zeit $O(mn)$.

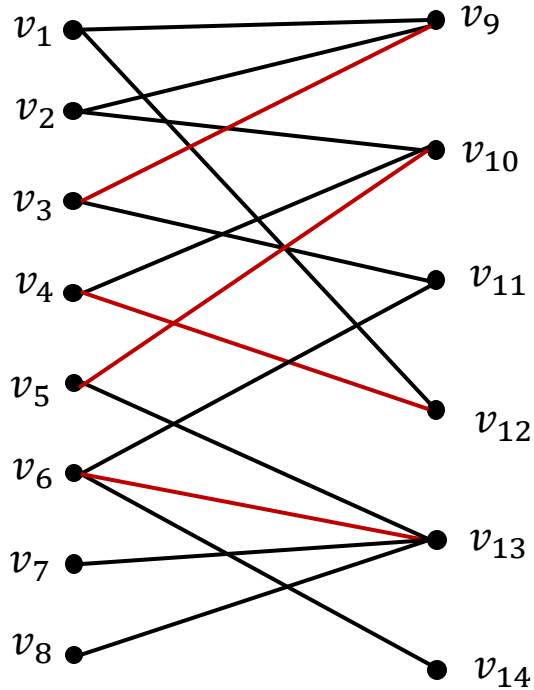
Beweis:

Korrektheit klar mit vorherigen Überlegungen.

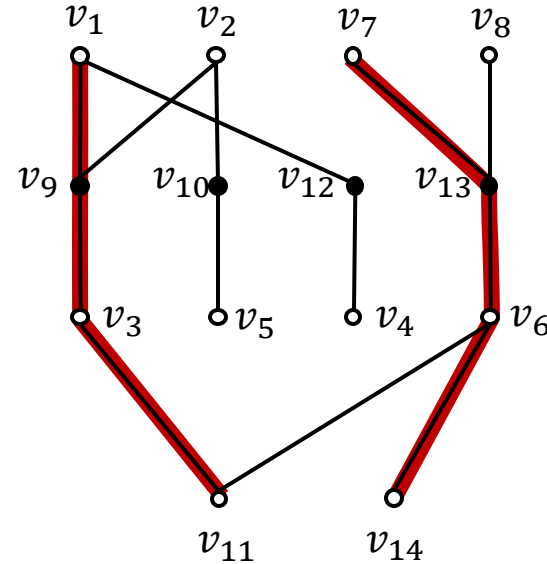
Laufzeit: n Iterationen mit Breitensuche $O(n + m)$



Exkurs: Suche über Level



Starte von allen ungematchten Knoten eine Suche wie gehabt.



Wie BFS

Wie DFS

Vom Level mit ungematchten Knoten, starte Suche rückwärts.

Algorithmus von Hopcroft-Karp

Algorithmus 2.20

Eingabe:

Ungerichteter Graph $G=(V,E)$

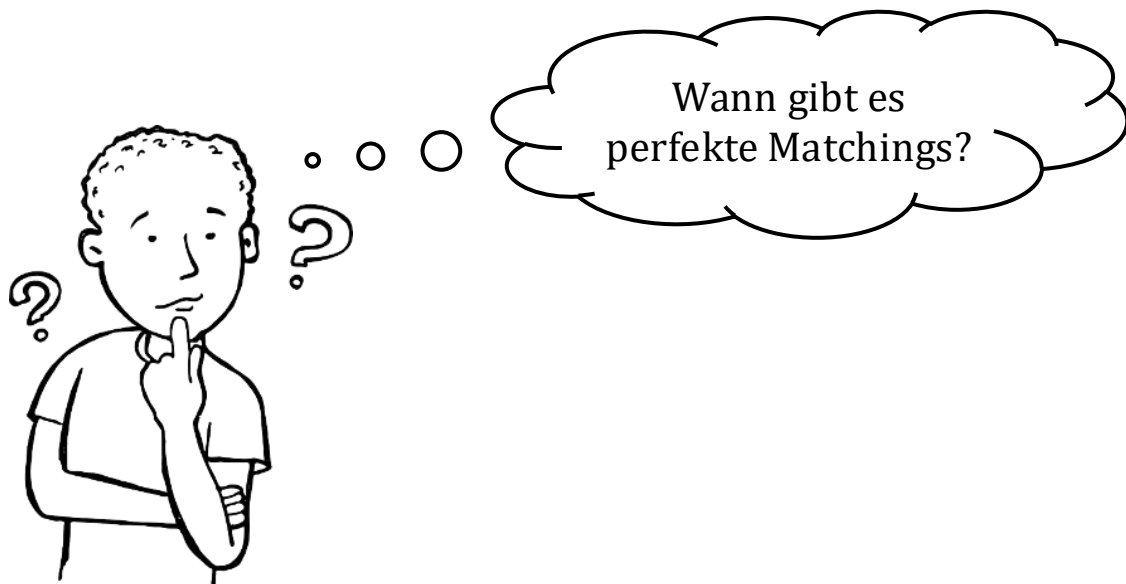
Ausgabe:

Maximales Matching M

1. Function HOPCROFTKARP(G)
2. Setze $M = \emptyset$
3. Setze $R := \{v \in V_1 \mid v \text{ ungematcht}\}$
4. Starte von allen Knoten in R BFS-mäßig, um „Level-Graphen“ zu konstruieren
5. Stoppe bei Level, welcher ungematchte Knoten aus V_2 enthält.
6. Sei $F \subseteq V_2$ die Menge an ungematchten Knoten in diesem Level.
7. Suche über diese Level DFS-mäßig augmentierende Pfade zwischen Knoten aus R und F .
8. Augmentiere M .
9. **If** es gab eine Verbesserung und V_1 enthält ungematchte Knoten **then**
10. Gehe zu Zeile 3.
11. **Else**
12. Return M

Satz 2.21

Algorithmus von Hopcroft-Karp löst Problem 2.13 (Max Matching) korrekt in Zeit $O(m\sqrt{n})$.



Satz von Hall

Definition 2.22 (Nachbarschaft)

Sei $G = (V, E)$ ein Graph. $N(X)$ bezeichnet die Nachbarschaft einer Menge $X \subseteq V$, also die Menge an Knoten, die adjazent zu Knoten aus X sind.

Satz 2.23

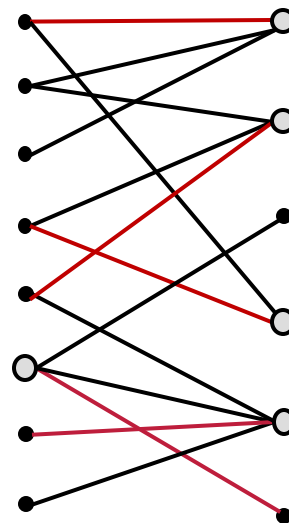
Sei $G = (V, E)$ ein bipartiter Graph mit $V = V_1 \dot{\cup} V_2$. Dann hat G genau dann ein V_1 überdeckendes Matching, wenn $|N(X)| \geq |X|$ für alle $X \subseteq V_1$ gilt.

Satz von Hall

Satz 2.23

Sei $G = (V, E)$ ein bipartiter Graph mit $V = V_1 \dot{\cup} V_2$. Dann hat G genau dann ein V_1 überdeckendes Matching, wenn $|N(X)| \geq |X|$ für alle $X \subseteq V_1$ gilt.

Beweis: Tafel!



Heiratssatz von Frobenius

Satz 2.23

Sei $G = (V, E)$ ein bipartiter Graph mit $V = V_1 \dot{\cup} V_2$. Dann hat G genau dann ein V_1 überdeckendes Matching, wenn $|N(X)| \geq |X|$ für alle $X \subseteq V_1$ gilt.

Korollar 2.24

Sei $G = (V, E)$ ein bipartiter Graph mit $V = V_1 \dot{\cup} V_2$. Dann hat G genau dann ein perfektes Matching, wenn $|V_1| = |V_2|$ und $|N(X)| \geq |X|$ für alle $X \subseteq V_1$ gilt.

Nächste Woche

Der Kernel ist König

