

Quiz

1. Jeder k -reguläre bipartite Graph $G = (A \cup B, E)$ für $k \geq 1$ hat ein Matching der Größe $|A|$.

- ☒ Wahr
- ☐ Falsch

Satz aus dem Skript
(folgt aus Satz von Hall)

2. Das (kardinalitäts-)maximale Matching in einem bipartiten Graphen kann in der Zeit $O(\sqrt{|V|} \cdot |E|)$ gefunden werden.

- ☒ Wahr
- ☐ Falsch

Hopcroft-Karp Algorithmus

3. Jeder Graph ohne Dreieck hat eine chromatische Zahl von höchstens 100.

- ☐ Wahr
- ☒ Falsch

$\chi(G)$ kann trotzdem beliebig gross sein
(Satz aus dem Skript)

4. Wenn G ein bipartiter Graph ist und M in G ein Matching aufweist, das nicht kardinalitätsmaximal ist, dann gibt es in G einen augmentierenden Pfad bezüglich M .

- ☒ Wahr
- ☐ Falsch

Satz von Berge

5. Wenn G ein zusammenhängender Graph mit einem maximalen Grad von 100 ist, dann hat G eine korrekte Einfärbung in 100 Farben, es sei denn, G ist ein vollständiger Graph.

- ☐ Wahr
- ☐ Falsch

Falls: G ist ein Pfad ungerader Länge

$\Rightarrow \chi(G) = 3 \Rightarrow G$ ist 100 färbbar

Sonst:

$\Rightarrow$ Satz von Brooks anwenden $\Rightarrow G$ ist 100 färbbar

6. Für einen vollständigen Graphen G mit gerader Anzahl von Knoten und positiven Gewichten $\ell(x, y)$, der die Dreiecksungleichung erfüllt, sei M die minimale Kosten-perfekte ^{Matching} Paarung und C die minimale Kosten-Reise des Handlungsreisenden. Dann ist $\ell(M) \leq \ell(C)/2$.

- ☒ Wahr wird im Beweis für die 1.5-Approximation verwendet
- ☐ Falsch

7. Ist G ein Graph mit maximalem Grad $\Delta(G)$, so findet ein Greedy-Algorithmus stets eine zulässige Färbung mit höchstens $\Delta(G) + 1$ Farben.

- ☒ Wahr
- ☐ Falsch

8. Jeder Kreis ist 2-färbbar.

- ☒ Wahr
- ☐ Falsch

Gegenbeispiel:



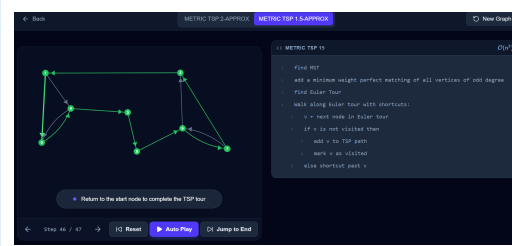
9. Es gibt einen Algorithmus mit polynomieller Laufzeit, der eine 1,5-Approximation für das metrische Problem des Handlungsreisenden findet.

- ☒ Wahr
- ☐ Falsch

10. Es gibt einen polynomiellen Algorithmus, der für jeden planaren Graphen eine geeignete Einfärbung mit 6 Farben findet.

- ☒ Wahr (sogar auch mit nur 4 möglich)
- ☐ Falsch

15 Approximation für metrisches TSP



• Auf Notizen von vorheriger Woche

Hopcroft-Karp Algorithmus

- Findet ein kardinalitätsmaximales Matching für einen bipartiten Graph
(viele Matching-Probleme sind bipartit)

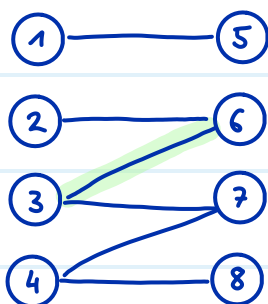
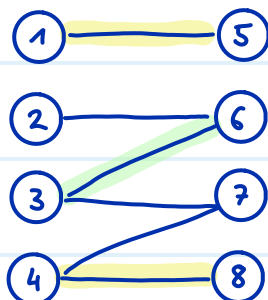
⊕

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

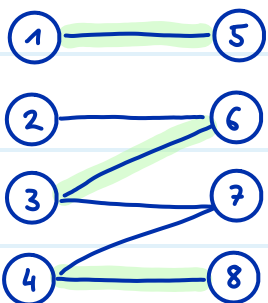
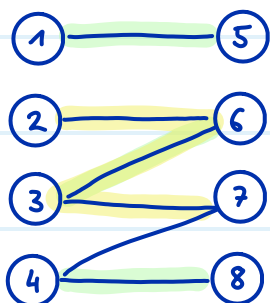
- 1: $M := \{e\}$ für irgendeine Kante $e \in E$.
 - 2: **while** es gibt noch augmentierende Pfade **do**
 - 3: $k :=$ Länge eines kürzesten augmentierenden Pfades
 - 4: Finde eine inklusionsmaximale Menge S von paarweise disjunkten augmentierenden Pfaden der Länge k .
 - 5: **for all** P aus S **do**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

- "inklusionsmaximal" = wir können keinen weiteren Pfad zu S hinzufügen
- "paarweise disjunkt" = es gibt keine zwei Pfade, die beide denselben Knoten v enthalten
- "augmentierend" = Endpunkte des Pfades sind nicht überdeckt, und die Kanten wechseln ab zwischen Matchingkanten und nicht Matchingkanten
- "Länge" = Anzahl Kanten ($\neq$ Anzahl Knoten)

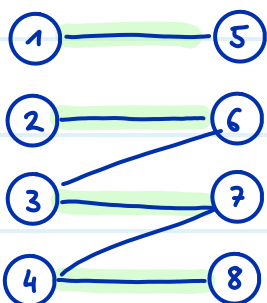
Bsp:

1: $M := \{e\}$ für irgendeine Kante $e \in E$.2: **while** es gibt noch augmentierende Pfade ✓ **do**3: $k :=$ Länge eines kürzesten augmentierenden Pfades $= 1$ 4: Finde eine inklusionsmaximale Menge S von paarweise disjunkten augmentierenden Pfaden der Länge k .

$$S = \{(1,5), (4,8)\}$$

es gäbe auch andere Möglichkeiten, z.B. $S = \{(1,5), (4,7)\}$ 5: **for all** P aus S **do** $P_1 = (1,5), P_2 = (4,8)$ 6: $M := M \oplus P$. // augmentiere entlang der Pfade aus S 2: **while** es gibt noch augmentierende Pfade ✓ **do**3: $k :=$ Länge eines kürzesten augmentierenden Pfades $= 3$ 4: Finde eine inklusionsmaximale Menge S von paarweise disjunkten augmentierenden Pfaden der Länge k .

$$S = \{(2,6,3,7)\}$$

5: **for all** P aus S **do** $P = (2,6,3,7)$ 6: $M := M \oplus P$. // augmentiere entlang der Pfade aus S 2: **while** es gibt noch augmentierende Pfade ✗ falsch **do**7: **return** M Satz von Berge garantiert, dass M kardinalitätsmaximal ist

Feststellungen:

1) Augmentierende Pfade haben immer ungerade Längen

→ weil die erste und letzte Kante nicht aus M sind und die Kanten dazwischen aus M und nicht M abwechseln

2) Die Knoten von einem augmentierenden Pfad wechseln zwischen A und B ab

→ weil G bipartit ist, d.h. jede Kante hat genau einen Endpunkt in A und einen in B

3) Genau ein Endknoten von einem augmentierenden Pfad ist in A und der andere in B

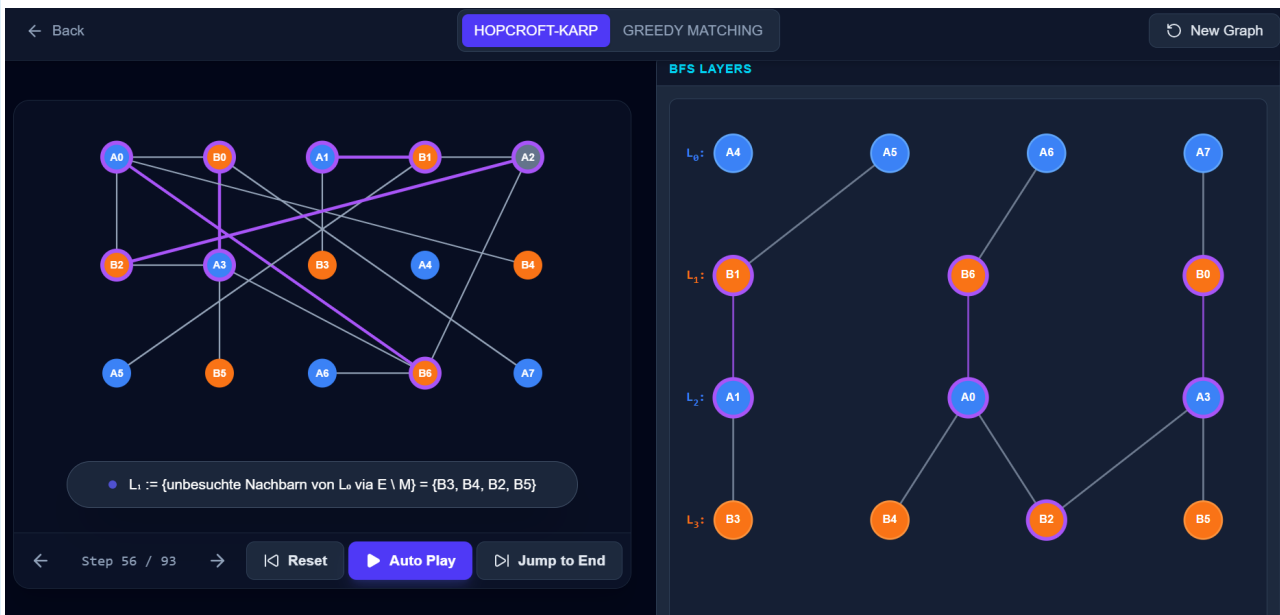
→ weil die Länge ungerade ist (1) und Knoten aus A und B abwechseln (2)

4.) $L_0 = \{v \in A \mid v \text{ ist nicht überdeckt}\}$ enthält alle Startknoten aller möglichen augmentierenden Pfade → folgt aus (3)

$L_1 = \{v \in B \mid v \text{ ist über eine Kante aus nicht } M \text{ mit einem Knoten aus } L_0 \text{ verbunden}\}$

L_2

:



Laufzeit: $O(\sqrt{|V|} |E|)$

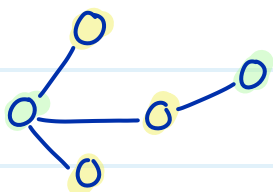
- Beweis dazu sollte man verstehen können, wird aber hier aus Zeitgründen ausgelassen
- Wenn ihr diesen Algorithmus erklären könnt, seid ihr gut vorbereitet für Matchings!
- Wichtigster Algorithmus bis jetzt

Färbungen

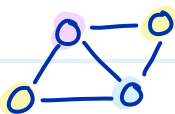
$[k] = \{1, 2, \dots, k\}$ sind die Farben

- Eine Färbung eines Graphen mit k Farben ist eine Abbildung $c: V \rightarrow [k]$, sodass keine zwei Nachbarknoten dieselbe Farbe haben, d.h. $\forall \{u, v\} \in E: c(u) \neq c(v)$
- Die chromatische Zahl $\chi(G)$ ist die minimale Anzahl an Farben, die für eine Färbung von G benötigt werden

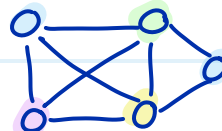
Bsp



$$\chi(G) = 2$$

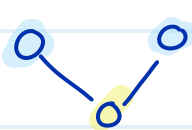


$$\chi(G) = 3$$



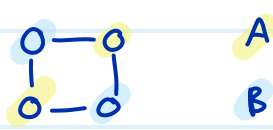
$$\chi(G) = 4$$

G ist bipartit falls man V in zwei Mengen A und B aufteilen kann, sodass alle Kanten genau einen Endpunkt in A und einen in B haben



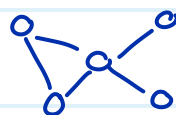
A
 B

bipartit? $\rightarrow$ Ja



A
 B

bipartit? $\rightarrow$ Ja

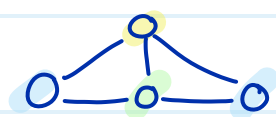


bipartit? $\rightarrow$ Nein

G ist bipartit $\Leftrightarrow G$ ist 2 färbbar

auch independent sets genannt

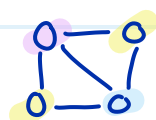
G ist k -partit falls man V in k Mengen $V_1, V_2, \dots, V_k$ aufteilen kann, sodass für alle Kanten die beiden Endpunkte nicht in der selben Menge V_i liegen.



V_1
 V_2
 V_3

2-partit? $\rightarrow$ Nein

3-partit? $\rightarrow$ Ja



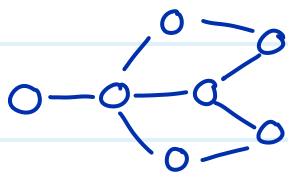
V_1
 V_2
 V_3

2-partit? $\rightarrow$ Nein

3-partit? $\rightarrow$ Ja

G ist k -partit $\Leftrightarrow G$ ist k färbbar

G ist bipartit $\Leftrightarrow$ es gibt keinen Kreis ungerader Länge (Beweis ausgelassen)

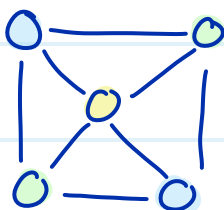


$\rightarrow$ ist bipartit $\rightarrow$ ist 2 färbbar

Eine Farbkasse ist eine Menge von Knoten derselben Farbe (= eine Menge V_i der Partition)

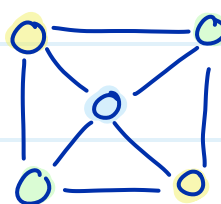
Wir dürfen Farbklassen tauschen.

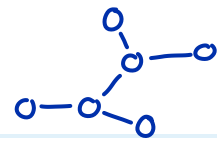
V_1 V_2 V_3



V_1 und V_2 tauschen

V_1 V_2 V_3



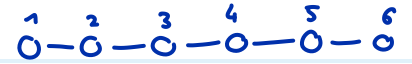


Was ist $\chi(T)$ für einen Baum T ?

→ 2 (weil ein Baum keine Kreise hat, und deshalb bipartit ist)

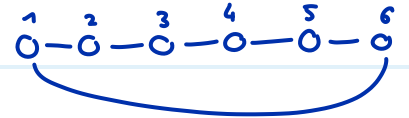
Was ist $\chi(P)$ für einen Pfad P ?

→ 2 (V_1 = gerade Knoten, V_2 = ungerade Knoten)



Was ist $\chi(C_n)$ für einen Kreis C_n gerader Länge?

→ 2 (V_1 = gerade Knoten, V_2 = ungerade Knoten)



Was ist $\chi(C_n)$ für einen Kreis C_n ungerader Länge?

→ 3 (einen Knoten entfernen, dann bleibt noch ein Pfad, $1+2=3$)

Was ist $\chi(K_n)$ für den vollständigen Graphen mit n Knoten?

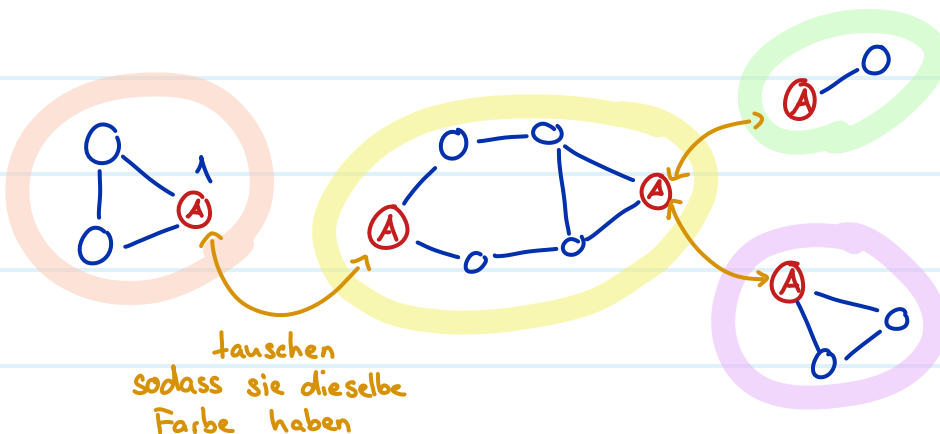
→ n (alle Knoten sind verbunden, also können keine zwei Knoten dieselbe Farbe haben)

Was ist $\chi(G)$ falls jeder einzelne Block von G k -färbbar ist?

→ k (durch Farbklassen tauschen)



Blöcke überschneiden sich nur in Artikulationsknoten

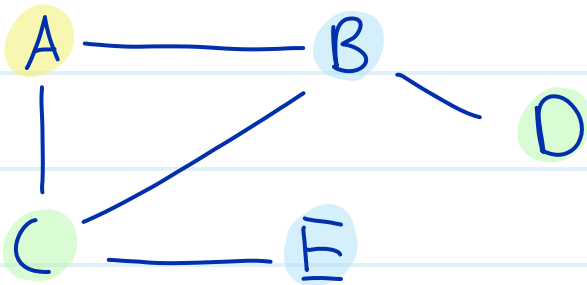


kann keine Konflikte
geben weil der Block-
Graph immer ein
Baum ist

Ein Chemiker muss fünf verschiedene Chemikalien (A, B, C, D, E) in Sicherheitsbehältern lagern. Manche Chemikalien reagieren heftig miteinander und dürfen daher **nicht im selben Behälter** aufbewahrt werden:

- A verträgt sich nicht mit B und C.
- B verträgt sich nicht mit A, C und D.
- C verträgt sich nicht mit A, B und E.
- D verträgt sich nicht mit B.
- E verträgt sich nicht mit C.

Was ist die kleinste Anzahl an Sicherheitsbehältern, die der Chemiker kaufen muss? $\chi(G) = 3$



Das ETH Prüfungsteam muss die Prüfungstermine festlegen. Ein Schüler darf jeweils nur eine Prüfung pro Tag schreiben.

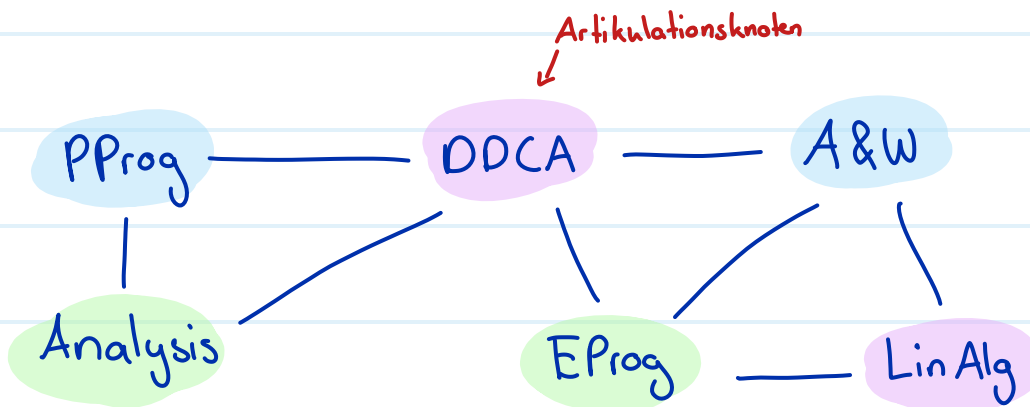
- Anna hat die Fächer PProg, DDCA, Analysis
- Peter hat die Fächer DDCA, EProg, A&W
- Max hat die Fächer EProg, LinAlg
- Theo hat die Fächer A&W, LinAlg
- Luna hat die Fächer DDCA, PProg

Tag 1: DDCA, LinAlg

Tag 2: A&W, PProg

Tag 3: EProg, Analysis

Wie viele Prüfungstage muss es mindestens geben? $\chi(G) = 3$





Sei G 2-färbbar. Wie können wir eine Färbung berechnen?

→ BFS (V_1 = Knoten mit gerader Distanz, V_2 = Knoten mit ungerader Distanz)

"Gibt es eine Färbung mit $\leq k$ Farben?" ist NP-vollständig für $k \geq 3$

Greedy-Färbung

GREEDY-FÄRBUNG (G)

- 1: wähle eine beliebige Reihenfolge der Knoten: $V = \{v_1, \dots, v_n\}$
- 2: $c[v_1] \leftarrow 1$
- 3: for $i = 2$ to n do
- 4: $c[v_i] \leftarrow \min\{k \in \mathbb{N} \mid k \neq c(u) \text{ für alle } u \in N(v_i) \cap \{v_1, \dots, v_{i-1}\}\}$

= kleinste Farbe, die kein Nachbar von v_i hat

- braucht höchstens $\Delta(G) + 1$ Farben (weil wenn wir v_i färben, gibt es höchstens $\Delta(G)$ gefärbte Nachbarn)
maximaler Grad
- Für jeden Graphen gilt deshalb: $\chi(G) \leq \Delta(G) + 1$
- Laufzeit $O(n+m)$ Achtung Skript: $O(m)$ falls zusammenhängend

• Wie gut der Greedy Algorithmus ist hängt sehr von der Reihenfolge ab

Verbesserung: Heuristik zur Bestimmung der Knotenreihenfolge
= Abschätzung

Idee: Schwierige Knoten zuerst färben, einfachere erst am Schluss
viele gefärbte Nachbarn wenig gefärbte Nachbarn

The screenshot shows a graph coloring simulation interface. On the left, a graph with 15 nodes (A-O) is displayed. Below the graph, a color palette shows 6 colors (1-6) and a button to 'Beginne Schleife: i = 15 bis 1'. On the right, the 'SMALLEST LAST COLORING' algorithm is shown with its pseudocode:

```

1  for i = n down to 1 do
2    vi ← vertex with minimum degree in current G
3    Remove vi and incident edges from G
4    c[vi] ← 1
5    for i = 2 to n do
6      c[vi] ← min{k ∈ N | k ≠ c(u) ∀ u ∈ N(vi) ∩ {v1, ..., vi-1}}
  
```

Below the pseudocode, the 'SMALLEST-LAST ORDER' section shows 'Determining Order...' and a legend for 'Current' (yellow), 'Colored' (green), and 'Pending' (grey). At the bottom, there are navigation buttons: 'Step 0 / 46', 'Reset', 'Auto Play', and 'Jump to End'.

Sei $k \in \mathbb{N}$. Falls jeder induzierte Subgraph einen Knoten mit Grad höchstens k hat, dann gibt es eine Färbung mit $k+1$ Farben.

Laufzeit $O(|V| + |E|)$, Beweis unwichtig

Achtung Skript: $O(m)$ falls zusammenhängend