Algorithmen und Datenstrukturen

Große Übung vom 01.11.2012

Christiane Schmidt

Diese Folien

- Braucht man nicht abzuschreiben
- Stehen im Netz unter www.ibr.cs.tu-bs.de/courses/ws1213/aud/index.html

Mailingliste

- Natürlich auf: www.ibr.cs.tu-bs.de/courses/ws1213/aud/index.html
- Name + Emailadresse + Passwort eintragen und absenden
- Mailingliste schickt E-Mail an die angegebene Adresse
- E-Mail der Mailingliste per „Antworten“ zurückschicken. Fertig!

Hausaufgaben

- 5 Hausaufgabenblätter + (vorher) 2 unbewertete Übungsblätter
- 14-tägig auf www.ibr.cs.tu-bs.de/courses/ws1213/aud/index.html
- Bearbeitungszeit: 14 Tage
- Abgabe: Mittwochs bis 11:25 vor dem IZ

1
2
3

Hausaufgaben

- 5 Hausaufgabenblätter + (vorher) 2 unbewertete Übungsblätter
- 14-tägig auf www.ibr.cs.tu-bs.de/courses/ws1213/aud/index.html
- Bearbeitungszeit: 14 Tage
- Abgabe: Mittwochs bis 11:25 vor dem IZ
  - Rückgabe: in den kleinen Übungen
  - 50% der Hausaufgabenpunkte sind Voraussetzung für Studienleistung

Große Übung

- Aufarbeitung des Vorlesungsstoffes
- An manchen Stellen auch Vertiefung des Stoffes
- Ihr könnt Themen mitbestimmen! Dazu einfach eine E-Mail an mich.
- Fragen stellen ausdrücklich erlaubt!

Übersicht über das Semester

KW	VL – Nr. (Di+Mi)	Gr. UE (Do)	Kl. UE (Mi+Do+Fr)	HA Ausgabe (Mi abends)	HA Abgabe (Mi bis 11:25 Uhr)	HA Rückgabe (in kl. UE)
42	–	–	–			
43	1, 2					
44	3, 4	1		UE_0 *, UE_0B*		
45	5, 6	2	1	HA_1		UE_0 *
46	7, 8					
47	9, 10	3	2	HA_2	HA_1	UE_0B *
48	11, 12	4				
49	13, 14		3	HA_3	HA_2	HA_1
50	15,16	5				
51	17,18	6	4	HA_4	HA_3	HA_2
52	Weihnachtsferien					
1						
2	19, 20		5			HA_3
3	21,22	7		HA_5	HA_4	
4	23,24	8	6			HA_4
5	25,26	9	7		HA_5	HA_5

*Bearbeitung in den ersten kleinen Übungen; keine Bewertung

Klausur

- Termin: 19.02.2013

Das Rundreise-Problem

Fortsetzung der Rundreise

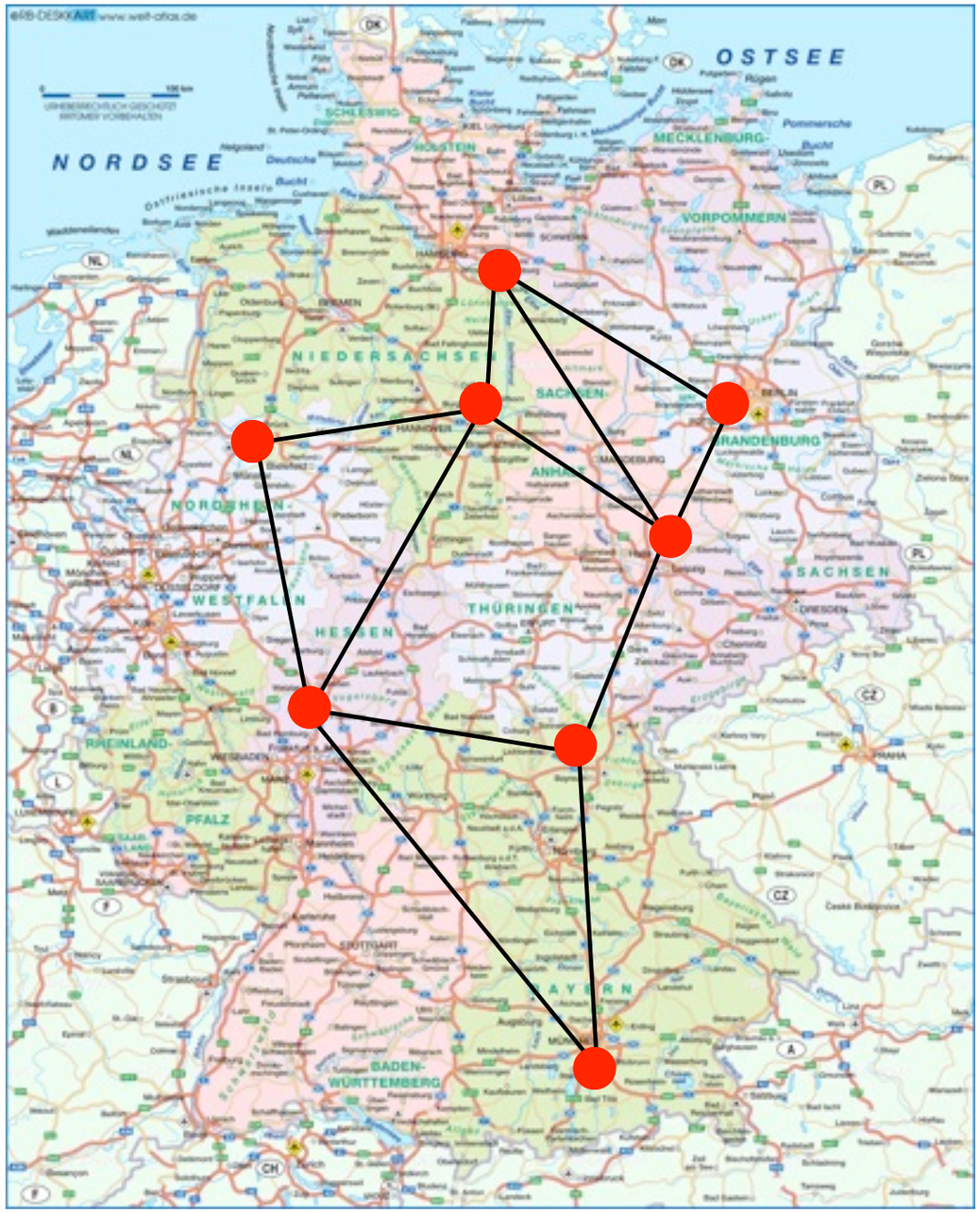
Problem: Finde eine Rundreise, (d.h. eine Route die jeden Ort besucht und zum Startort zurückkehrt) die möglichst kurz ist.

Fortsetzung der Rundreise

Fortsetzung der Rundreise

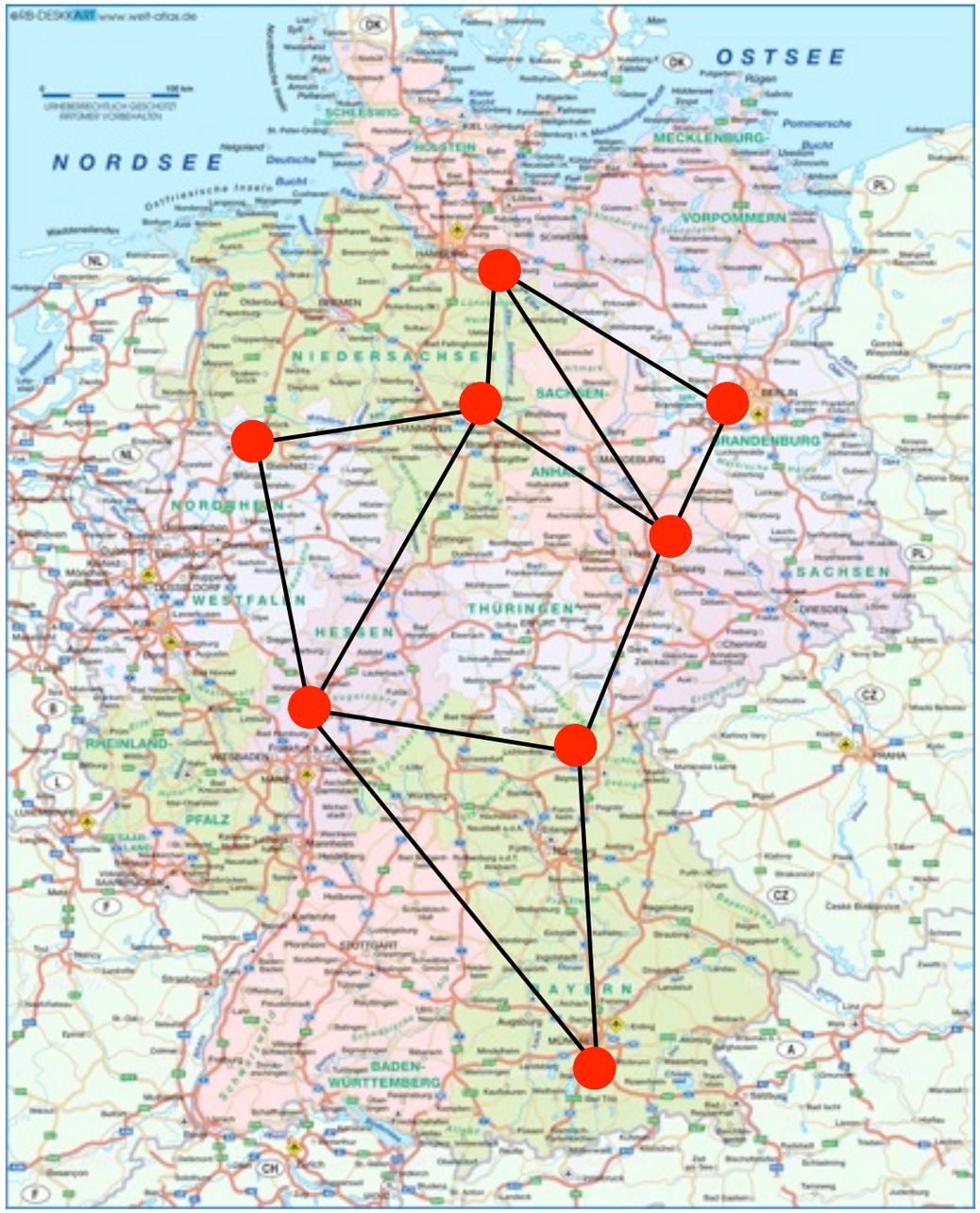
- Wie funktioniert map24.de?
- Man kann zwischen den Städten beliebig hin und her reisen.

Phase 1: Modellierung



- Zur Modellierung kann man Graphen benutzen
- Ein Graph G besteht aus Knoten und Kanten

Fortsetzung der Rundreise
Phase 1: Modellierung

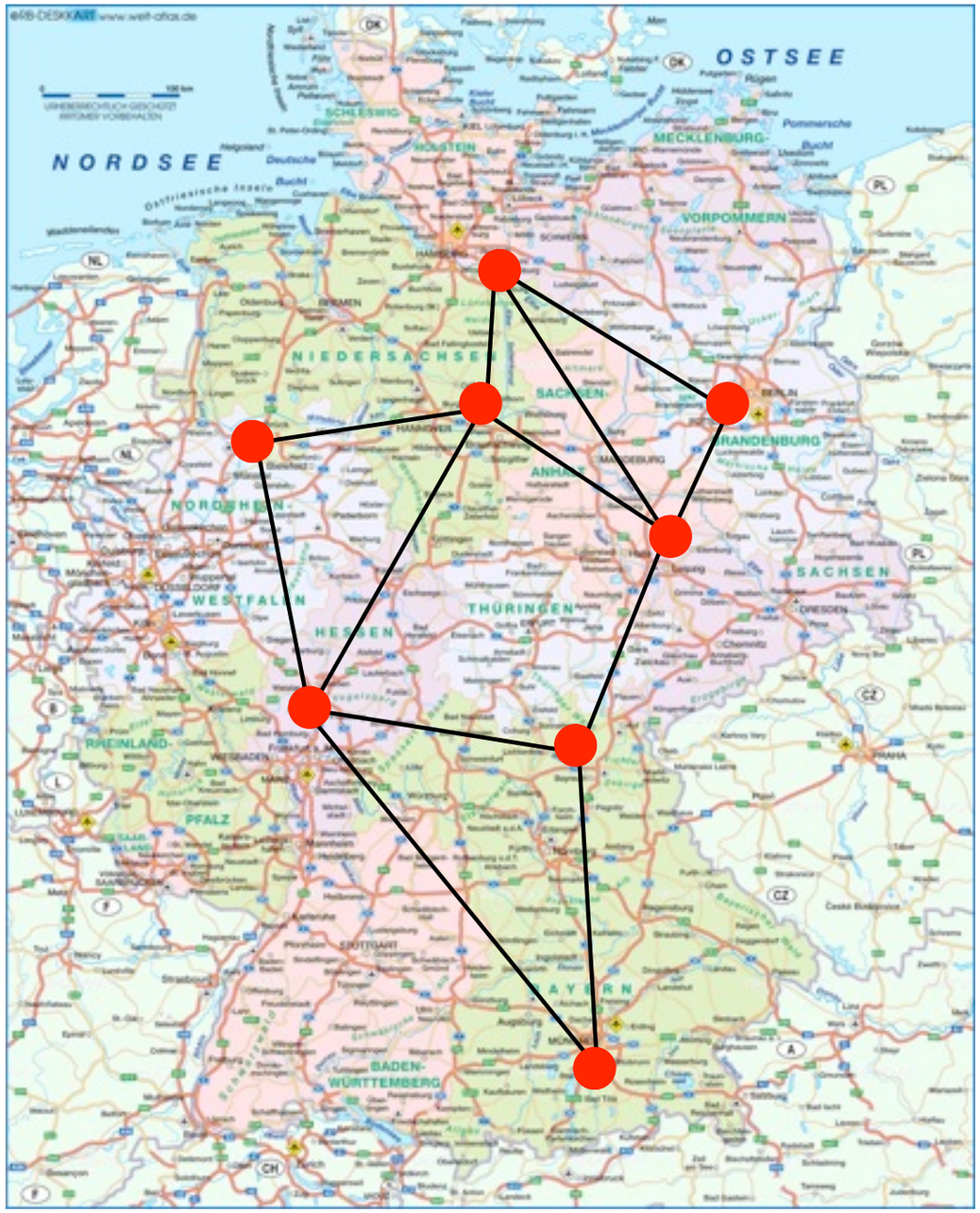


- Knoten:
  - Hamburg
  - Berlin
  - Braunschweig
  - Leipzig
  - ……

Typische Schreibweise

- Kanten:
  - Hamburg, Berlin
  - BS, Berlin
  - BS, Leipzig
  - ……

Phase 2: Datenstruktur



- Daten in „geeigneter Form“ an den Computer übergeben
- Was sind die relevanten Daten in dem Problem?
  - Städte
  - Verbindungen
  - Entfernungen

Phase 2: Datenstruktur

Bald in dieser Vorlesung

- Eine anwendbare Datenstruktur sind z.B. Listen
- Speichere alle „benachbarten Orte“ in [illegible]er Liste
- [illegible]rg, Leipzig, [illegible]ster, Frankfurt,…)
- [illegible]e Entfernungen in einer anderen:
- (220, 200, 250,…)

Phase 3: Algorithmen

Bald in dieser Vorlesung

- Es ist „schwierig“ die beste Lösung zu finden
- Aber man kann schnell „ziemlich gute“ finden
- Al[illegible] Startort [illegible]e zum nächstgelegenen Ort außer, dieser ist bereits besucht

Turnierplanung

Szenario

- n Spieler einer Mannschaft sollen nach Spielstärke aufgestellt werden
- Dazu: Turnier „jeder gegen jeden“
- Pro Abend spielt jeder Spieler ein Spiel
- Frage: wie viele Abende werden benötigt?

Anzahl

- Sei n=6 (wir haben 6 Spieler)
- Jeder Spieler muss genau einmal gegen jeder der 5 anderen Spieler antreten
- Damit: 6*5/2 = 15 Spiele zu absolvieren (denn: Spiel i gegen j würde für beide Spieler gezählt werden -->*1/2)
- Da jeder Spieler ein Spiel am Abend: 3 Spiele pro Abend

> 15/3=5 Abende werden benötigt

Erster Ansatz: einfach Spielen

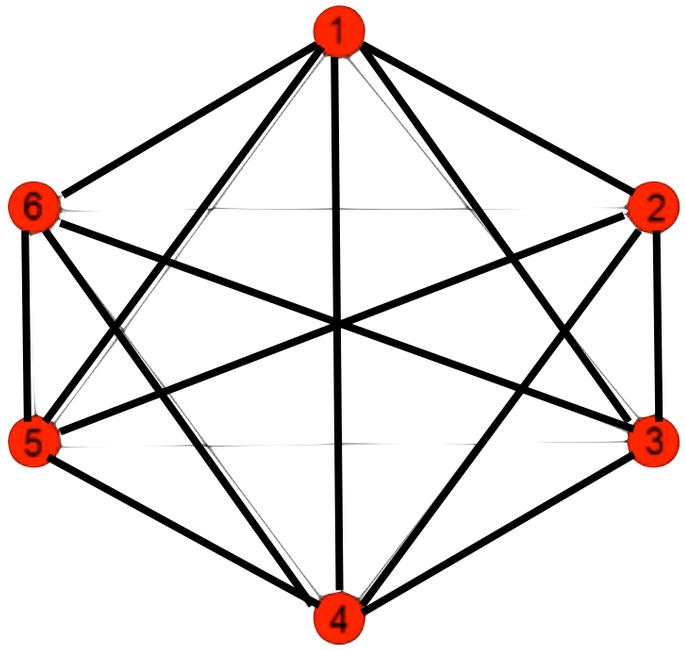
1. Abend	2. Abend	3. Abend
1–2	1–3	1–4
3–5	2–6	2–5
4–6	4–5	3–6

Aber:
restlichen 6 Spielpaarungen nicht an 2 Abenden!!
(1-5 kann weder mit 1-6 noch 5-6 parallel ausgetragen werden)

Hier muss auch je ein Spieler aussetzen!!

4. Abend		6. Abend
1–5		5–6
2–3	2–4	3–4

- Wir modellieren das Problem mit Hilfe eines Graphen:
  - Knoten: jeder Spieler
  - Kanten: Spiele
- Für n=6:

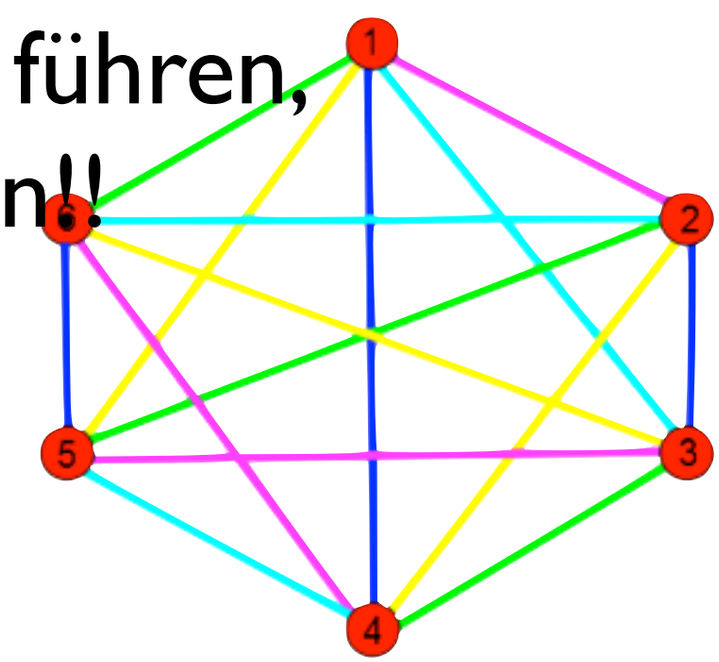


Graphen in denen jeder Knoten mit jedem verbunden ist heißen *vollständig*

Ansatz: Färben

- Wir färben die Kanten mit den Farben 1,…, n-1
- Jede Farbe entspricht einer Runde
- Färbung zulässig – wenn sie zulässigem Spielplan entspricht

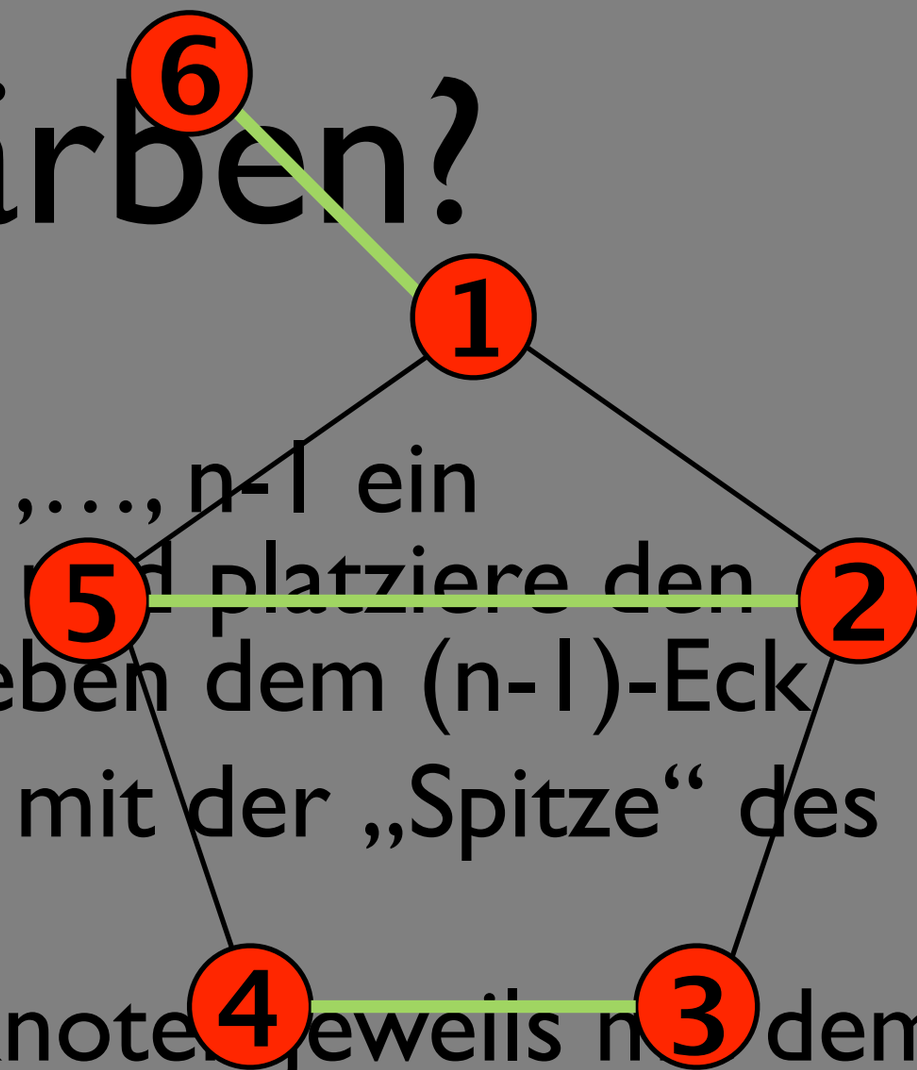
> Alle Kanten, die in gleich Knoten führen, müssen unterschiedlich gefärbt sein!!



Wie Färben?

1. Bilde aus den Knoten 1,…, n-1 ein regelmäßiges (n-1)-Eck und platziere den Knoten n links oben neben dem (n-1)-Eck
2. Verbinden den Knoten mit der „Spitze“ des (n-1)-Ecks
3. Verbinde die übrigen Knoten jeweils mit dem gegenüberliegenden Knoten auf der gleichen Höhe im (n-1)-Eck
4. Die eingefügten n/2 Kanten werden mit der ersten Farbe gefärbt

Wie Färben?

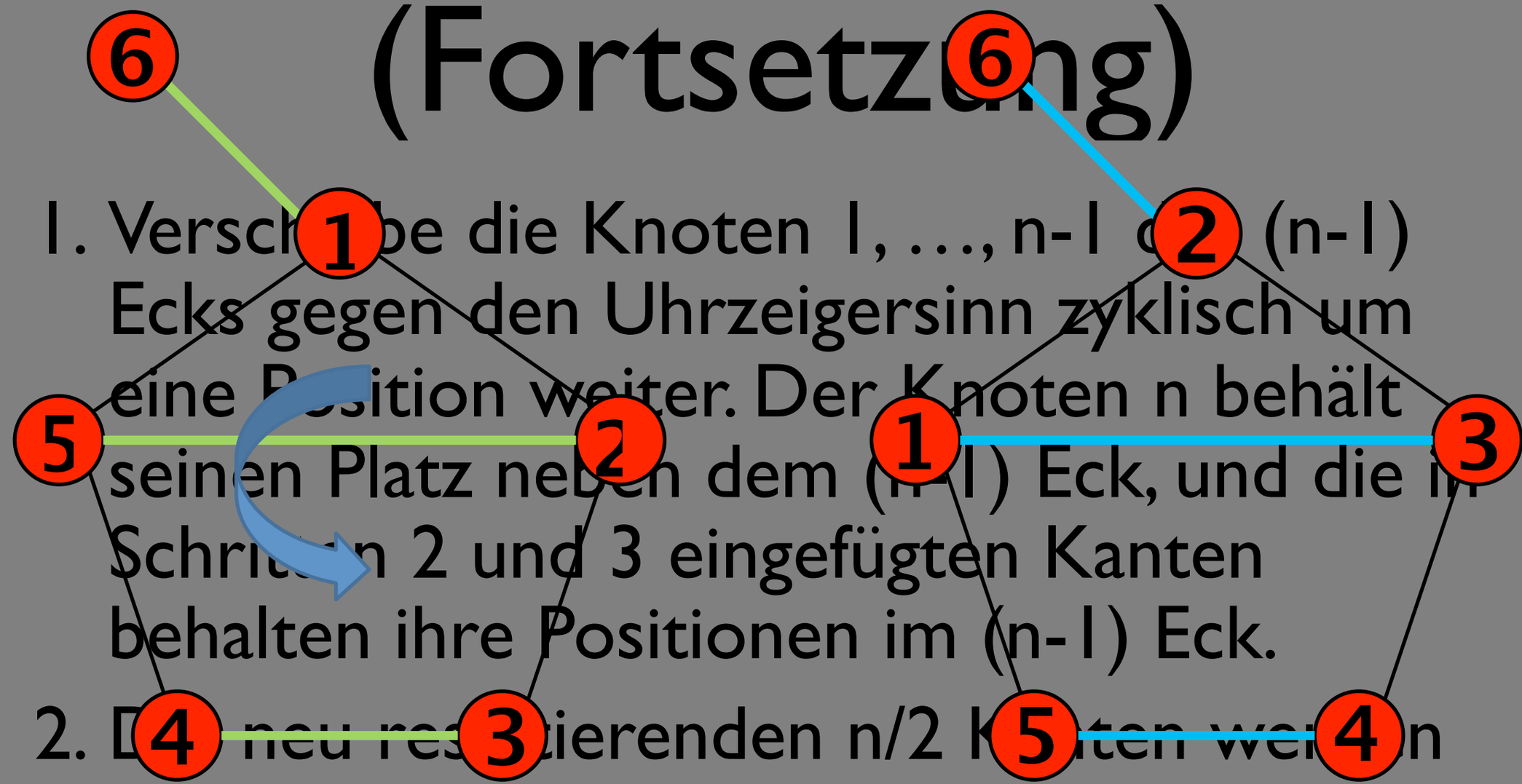


1. Bilde aus den Knoten 1,…, n-1 ein regelmäßiges (n-1)-Eck und platziere den Knoten n links oben neben dem (n-1)-Eck
2. Verbinden den Knoten mit der „Spitze“ des (n-1)-Ecks
3. Verbinde die übrigen Knoten jeweils mit dem gegenüberliegenden Knoten auf der gleichen Höhe im (n-1)-Eck
4. Die eingefügten n/2 Kanten werden mit der ersten Farbe gefärbt

Wie Färben? (Fortsetzung)

1. Verschiebe die Knoten 1, …, n-1 des (n-1) Ecks gegen den Uhrzeigersinn zyklisch um eine Position weiter. Der Knoten n behält seinen Platz neben dem (n-1) Eck, und die in Schritten 2 und 3 eingefügten Kanten behalten ihre Positionen im (n-1) Eck.
2. Die neu resultierenden n/2 Kanten werden mit der zweiten Farbe gefärbt
3. Die Schritte 5 und 6 werden für die Farben 3, …, n-1 wiederholt

Wie Färben? (Fortsetzung)



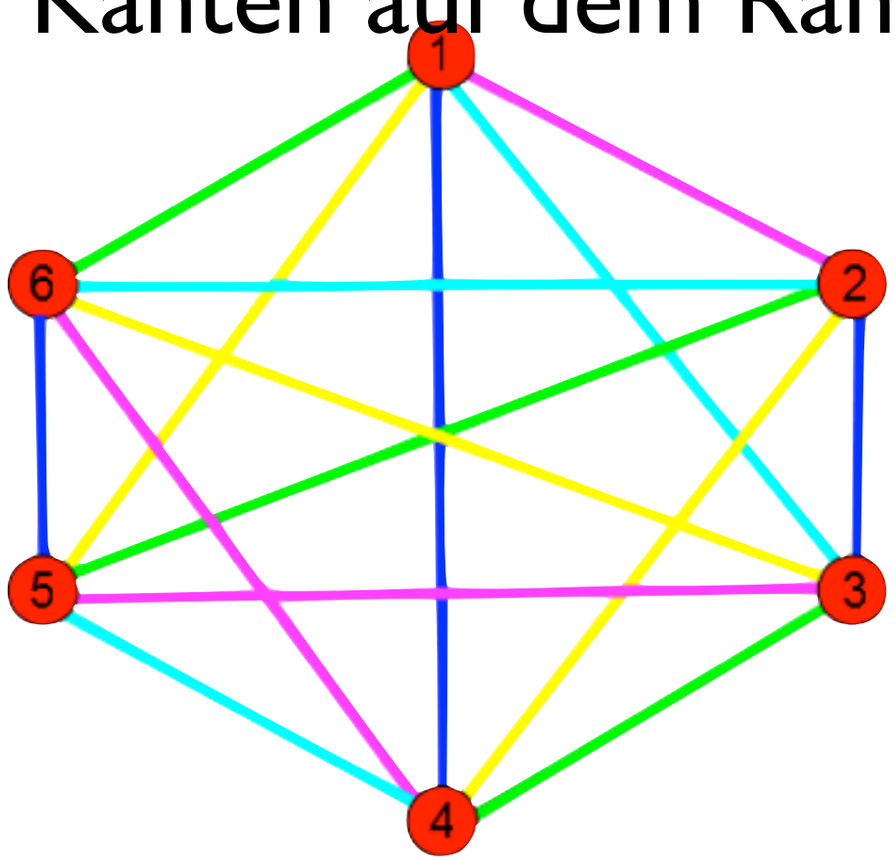
1. Verschiebe die Knoten 1, …, n-1 auf (n-1) Ecks gegen den Uhrzeigersinn zyklisch um eine Position weiter. Der Knoten n behält seinen Platz neben dem (n-1) Eck, und die in Schritten 2 und 3 eingefügten Kanten behalten ihre Positionen im (n-1) Eck.
2. Die neu resultierenden n/2 Kanten werden mit der zweiten Farbe gefärbt
3. Die Schritte 5 und 6 werden für die Farben 3, …, n-1 wiederholt

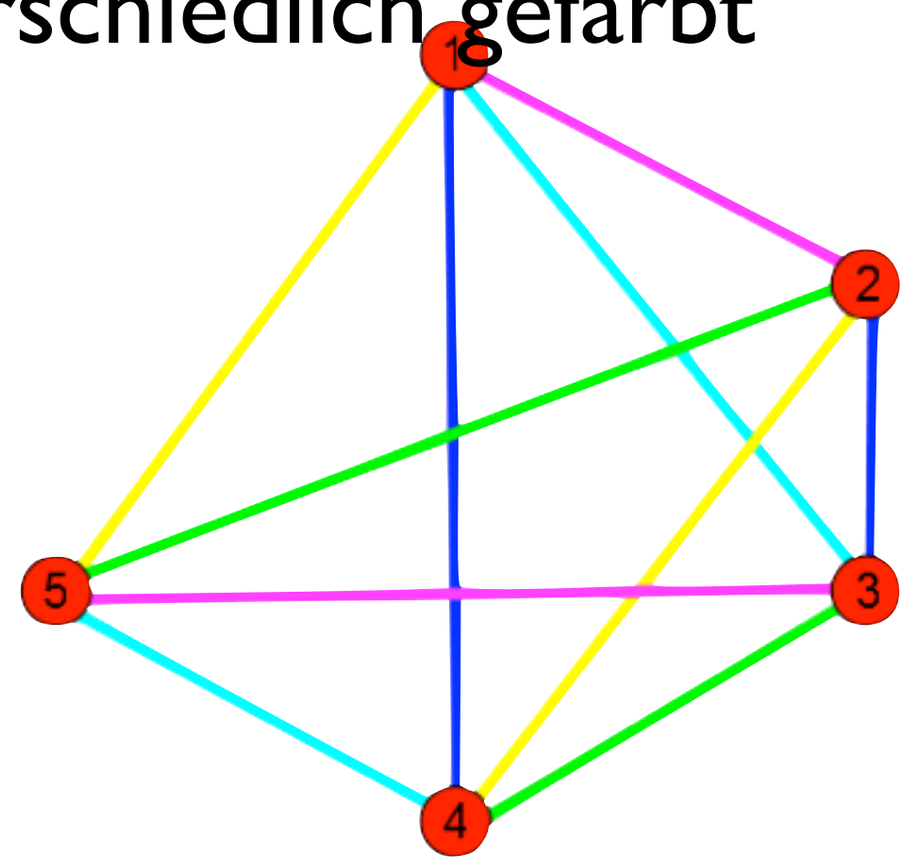
usw.

Wieso funktioniert dieser Algorithmus?

- Beobachtung:

Entfernen wir Knoten 6 und alle 5 seiner Kanten, erhalten wir 5-Eck, in dem alle 5 Kanten auf dem Rand unterschiedlich gefärbt

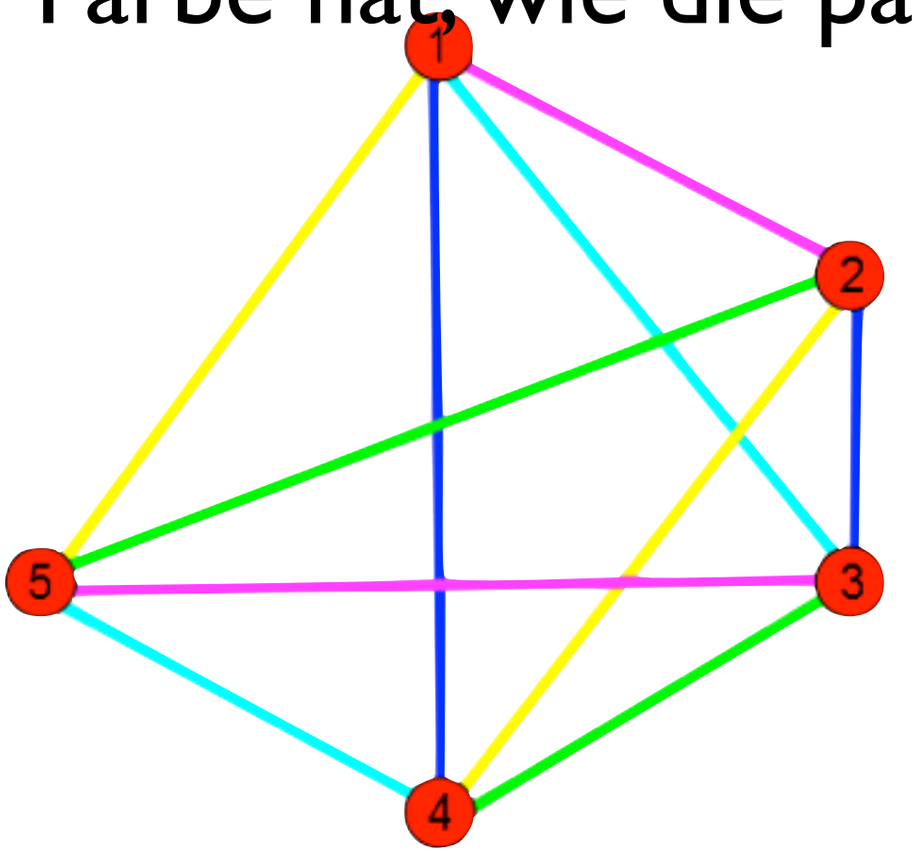


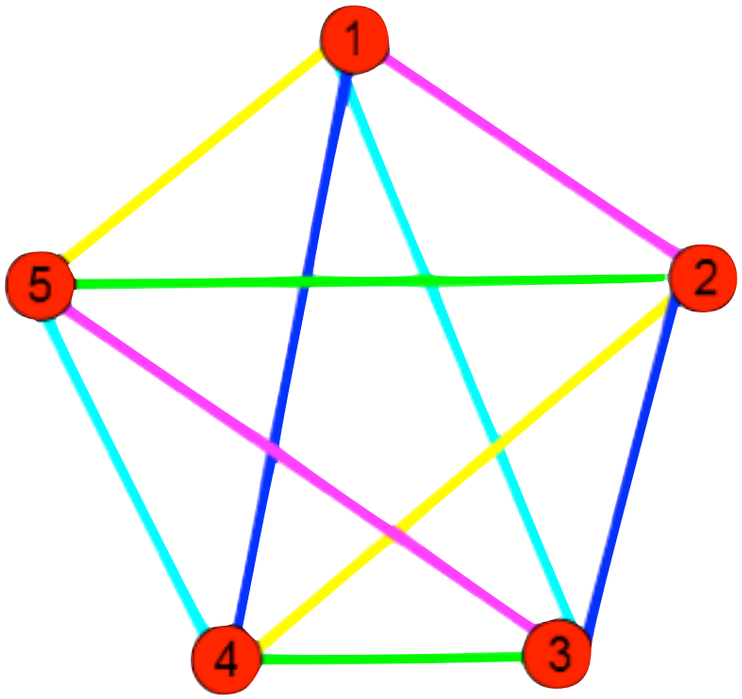


Wieso funktioniert dieser Algorithmus?

- Beobachtung 2:

Betrachten wir dies als regelmäßiges 5-Eck, fällt auf, dass jede Kante im Inneren die gleiche Farbe hat, wie die parallele Kante auf Rand.





Wieso funktioniert dieser Algorithmus?

- Wenn immer nur parallele Kanten die gleiche Farbe haben, die ja nicht in den gleichen Knoten führen, ist unsere Zulässigkeitsbedingung erfüllt
- Und: an jedem der 5 Knoten werden 4 Farben verwendet – dabei jeweils eine andere, die für die Verbindung zu Knoten 6 verwendet werden kann

Einfachere Formulierung

- Oben: wir müssten (n-1)-Ecke speichern
- Geht das einfacher?
- Dazu: Modulo-Rechnung (Division mit Rest)

Division mit Rest

- Zwei natürliche Zahlen *a* und *b* sollen mit Rest dividiert werden, d.h., man sucht eine Darstellung: $a = b*n + r$
- Die Modulofunktion ordnet zwei Zahlen den Teilerrest zu – also: $a \bmod b = r$
- Beispiele:
  - 19 mod 7 = 5,
  - denn: 19=2*7+5
  - 37 mod 17 = 3,
  - denn: 37= 2*17 + 3

Einfachere Formulierung

1. Für alle Farben
2. Färbe die Kante [i,n] mit der Farbe i
3. Für k=1, …, n/2-1

   färbe alle Kanten

   [(i+k) mod (n-1), (i-k) mod (n-1)]

   mit der Farbe i.

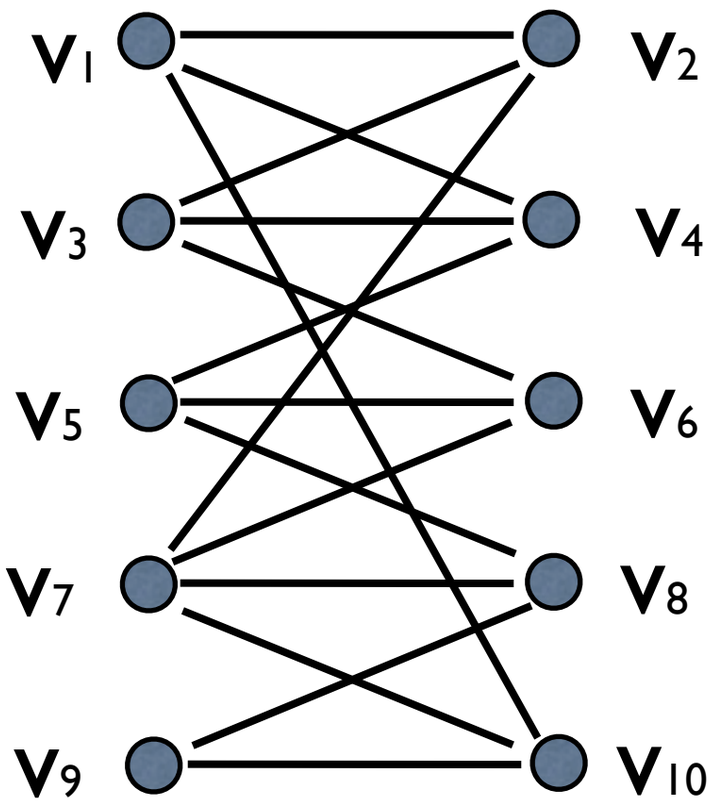
Da unsere Knoten von 1,...,n-1 durchnummeriert sind (und nicht von 0,1,...,n-2), wird der Rest 0 als n-1 interpretiert.

Was haben wir gesehen?

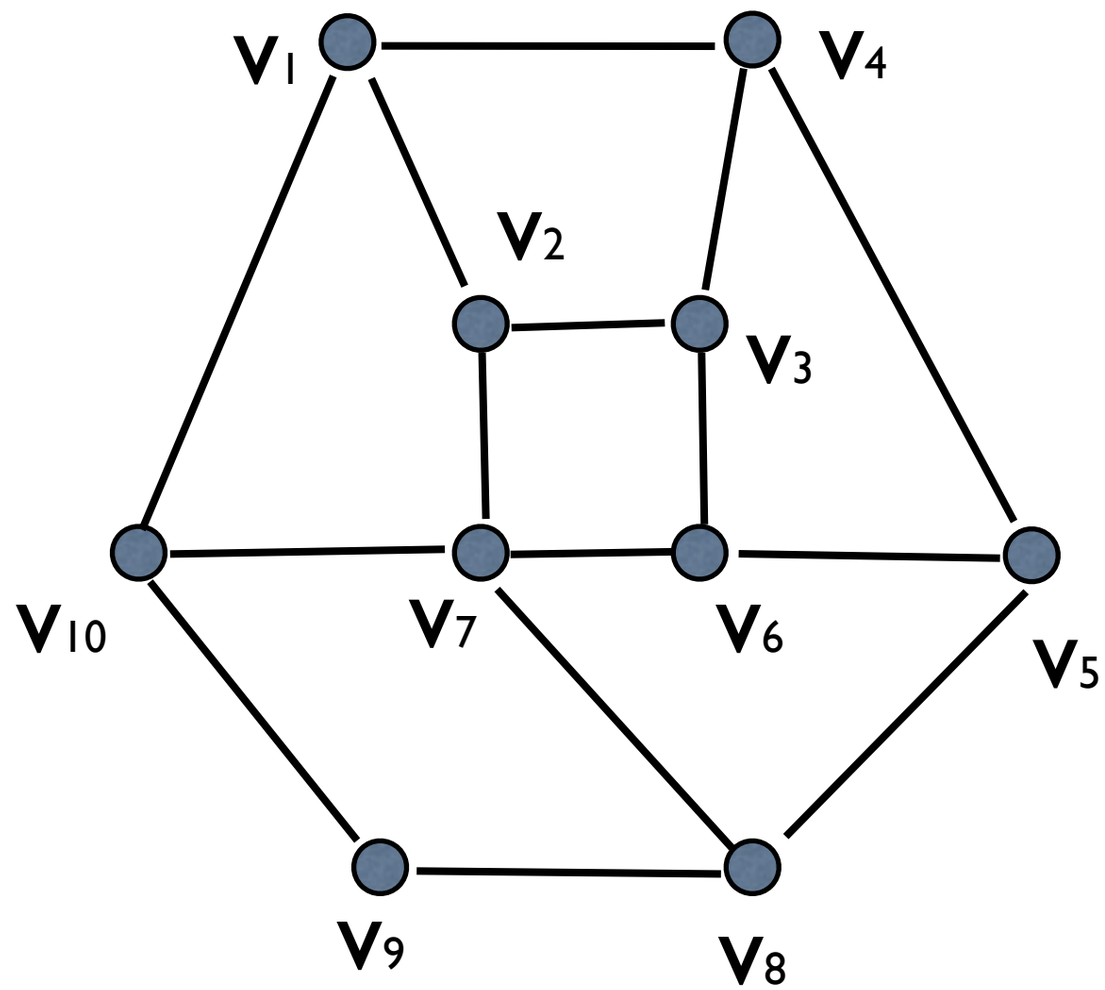
- Es lohnt sich über Algorithmen nachzudenken (statt (n-1)-Ecke mit zyklischer Verschiebung, Modulo-Rechnung)
- Pseudo-Code:

1. Für alle Farben
2. Färbe die Kante [i,n] mit der Farbe i
3. Für k=1, …, n/2
   färbe alle Kanten
   [(i+k) mod (n-1), (i-k) mod (n-1)] mit der Farbe i.

Graphen

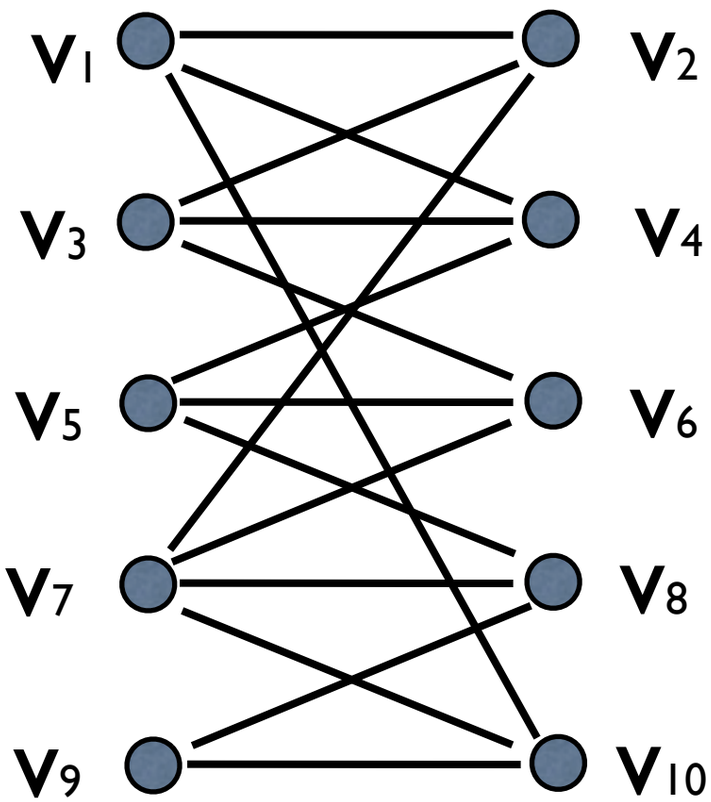


G_1

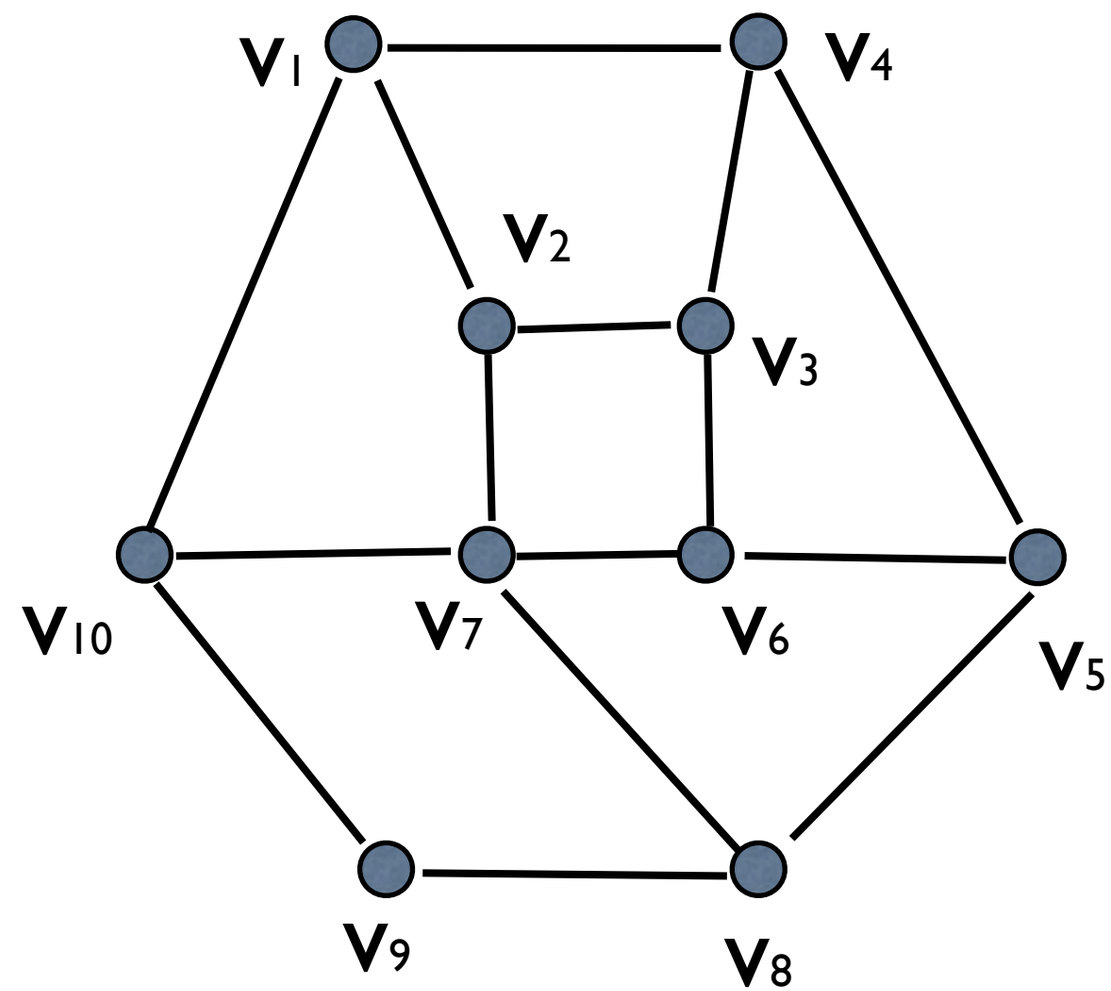


G_2

Graphen



G_1

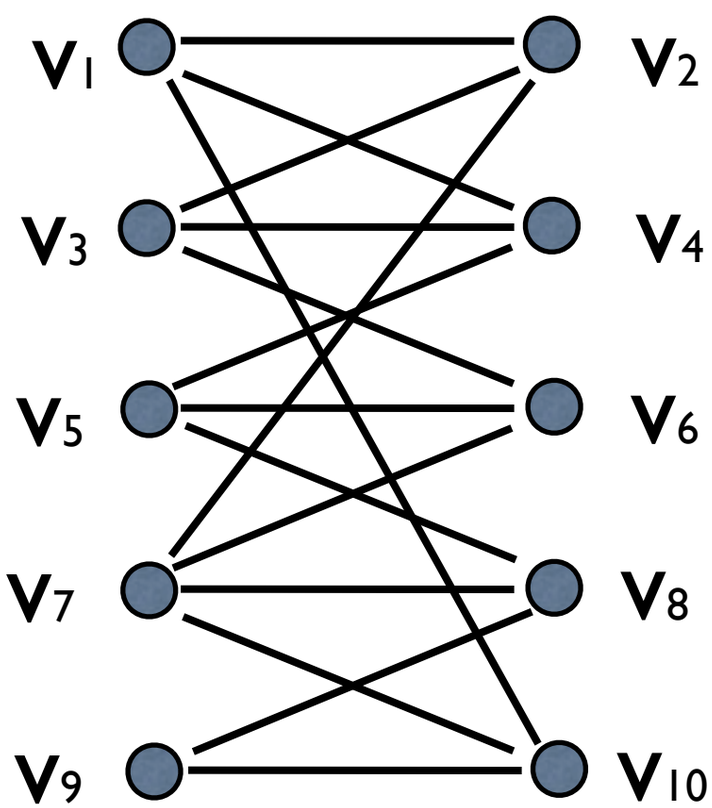


Sind G1 und G2 gleich ?

G_2

Einbettung ist unterschiedlich! => Graph kann unterschiedlich aussehen!

Graphen



G_1

Knotenmenge:

$V=\{v_1,v_2,v_3,v_4,v_5,v_6,v_7,v_8,v_9,v_{10}\}$

$n=|V|=10$

Kantenmenge:

$E=\{e_{1,2}; e_{1,4}; e_{1,10}; ...\}$

$m=|E|=15$

keine parallelen Kanten, keine Schleifen => einfach!

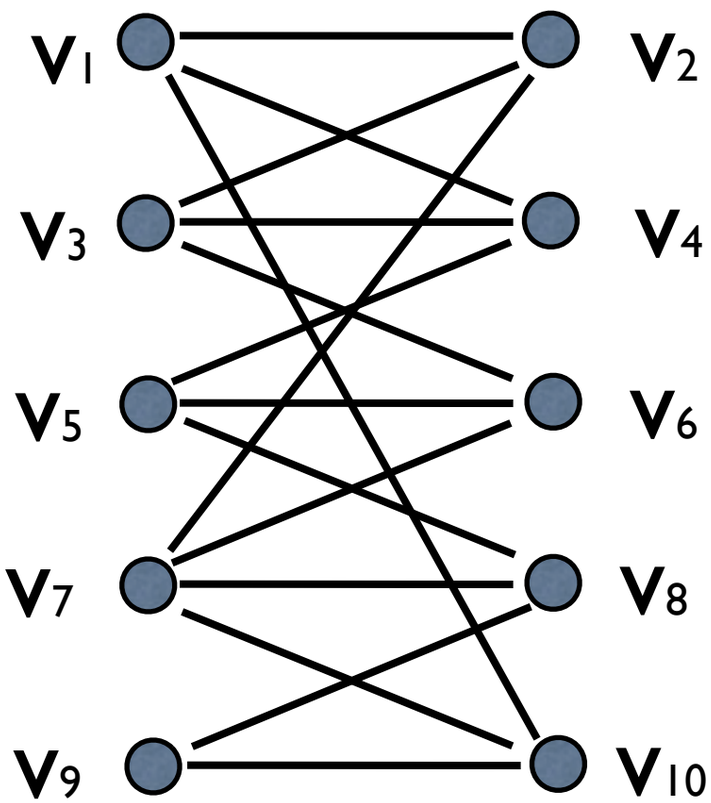
adjazent zu v_7:

v_2, v_6, v_8, v_{10}

inzident zu v_7:

$e_{2,7}$; $e_{6,7}$; $e_{7,8}$; $e_{7,10}$

Graphen



G_1

Grad von v_7: $\delta(v_7)=4$

Teilgraph:
$H=(V, \{e_{1,2}; e_{2,3}; e_{3,6}; e_{2,7}; e_{3,4}; e_{5,6}; e_{7,8}; e_{8,9}; e_{9,10}\})$
ist aufspannend

Kantenfolge: v_2; $e_{2,3}$; v_3; $e_{3,6}$; v_6; $e_{6,7}$; v_7; $e_{7,2}$; v_2; $e_{2,3}$; v_3; $e_{3,4}$; v_4

ist kein Weg! ($e_{2,3}$ zweimal)

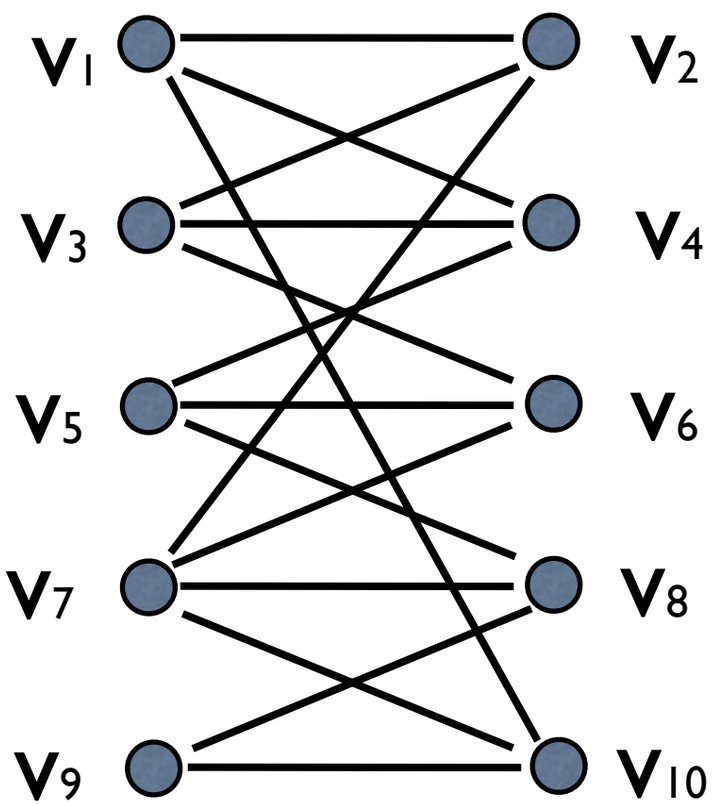
Weg: $e_{2,3}$; $e_{3,6}$; $e_{6,7}$; $e_{7,2}$; $e_{2,1}$

ist kein Pfad! (v_2 doppelt!)

Pfad: $e_{1,2}$; $e_{2,3}$; $e_{3,6}$; $e_{6,5}$; $e_{5,8}$; $e_{8,9}$; $e_{9,10}$

Kreis: $e_{1,2}$; $e_{2,3}$; $e_{3,6}$; $e_{6,5}$; $e_{5,8}$; $e_{8,9}$; $e_{9,10}$; $e_{10,1}$

Graphen



G_1

Eulerweg?
gibt es nicht! (max 2 Knoten unger. Grad)

Eulertour?
gibt es nicht!

Hamiltonpfad:
$e_{1,2}$; $e_{2,3}$; $e_{3,4}$; $e_{4,5}$; $e_{5,6}$; $e_{5,7}$; $e_{7,8}$; $e_{8,9}$; $e_{9,10}$

Hamiltonkreis:
$e_{1,2}$; $e_{2,3}$; $e_{3,4}$; $e_{4,5}$; $e_{5,6}$; $e_{5,7}$; $e_{7,8}$; $e_{8,9}$; $e_{9,10}$; $e_{10,1}$

Graph ist zusammenhängend

… nächste Übung in 1 Woche

Schönes Wochenende….