Modellbildung

Modelle in der Informatik
(bzw. in der Wissenschaft allgemein)

Teil III: Modellbildung

1. Einführung
2. Optimale Zugverbindungen
3. Planare Graphen / Färbungsprobleme

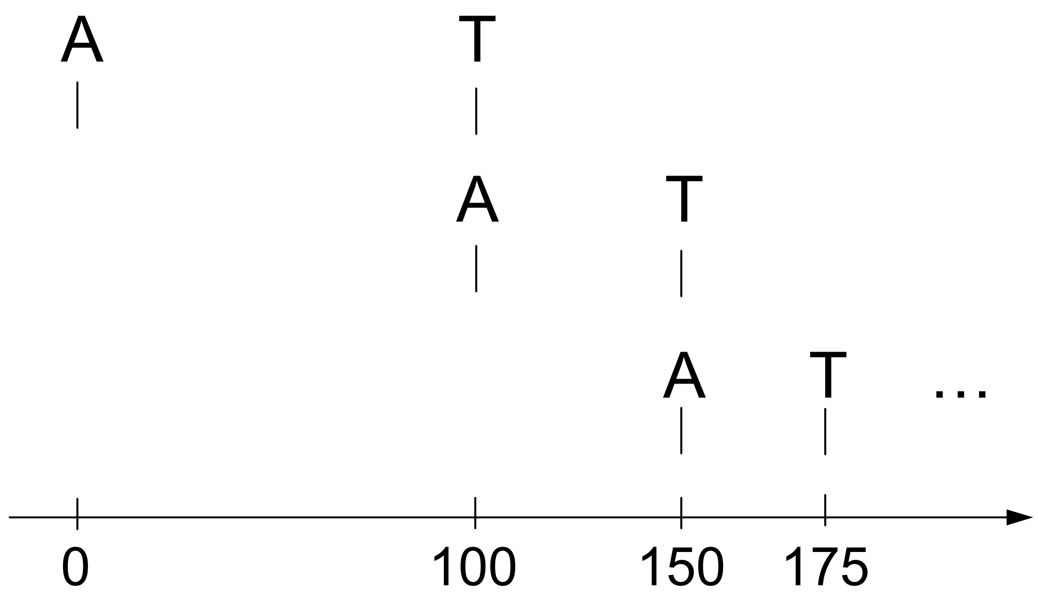
Achille and the turtle

- Achille runs twice as fast as a turtle.
  In a track race, the turtle is granted an initial lead of 100m.
  Both start to run in the same direction at exactly the same time.

- When Achille reaches the starting point of the turtle, the turtle still has a lead, namely by 50m.
  When Achille reaches that point, the turtle is still in the lead by 25m etc.
  Thus, Achille can never overtake the turtle!

(Was stimmt hier nicht?)

Achille and the turtle (1)



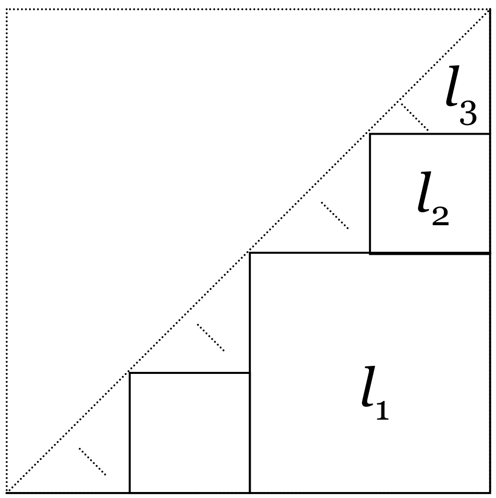
$\sum$: 100 + 50 + 25 + 12.5 + … = 200.

$\boxed{\text{The sum of infinte many things may be finite.}}$

Walking the grid

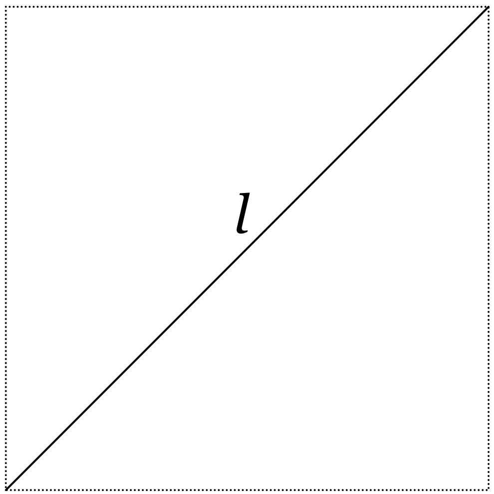
Walk in unit square near or on diagonal.

Manhattan style approximation of diagonal



Length (l_n) = 2, all n

Diagonal



Length (l) = $\sqrt{2} \approx 1.4142$

$$2 \stackrel{?}{=} \sqrt{2}$$

Walking the grid (1)

“Limit of lengths” $\neq$ “Length of limit”

If transitions are not smooth, the order of things matters.

(Rein lebenspraktisch kann man nicht beliebig häufig die Gehrichtung wechseln, so dass sich im Grenzfall $\sqrt{2}$ ergibt.)

The Petersburg paradox

(how to win 1 unit with certainty in roulette)

1. Play for the basic chance (black/red).
2. Start by betting 1 unit.
3. If you win, quit and leave with winning 1 unit.
4. As long as you loose, double the input.
5. When finally you win (probability 1), you win 1 unit in total.

The Petersburg paradox (1)

Example: always go for black

black →	1
red, black →	2 – 1 = 1
red, red, black →	4 – 2 – 1 = 1
red, red, red, black →	8 – 4 – 2 – 1 = 1
red, red, red, red, black →	16 – 8 – 4 – 2 – 1 = 1
⋮	

The world is finite.

(Irgendwann wird die Bank gesprengt.

Andauernde Verdoppelung ist lebenspraktisch unmöglich.)

2. Optimale Zugverbindungen

- Einführung
- Übersetzung der Fahrplansuche in ein kürzestes Wegeproblem
- Suche eines schnellsten Weges im Verbindungsgraphen
- "Optimale" Zugverbindungen

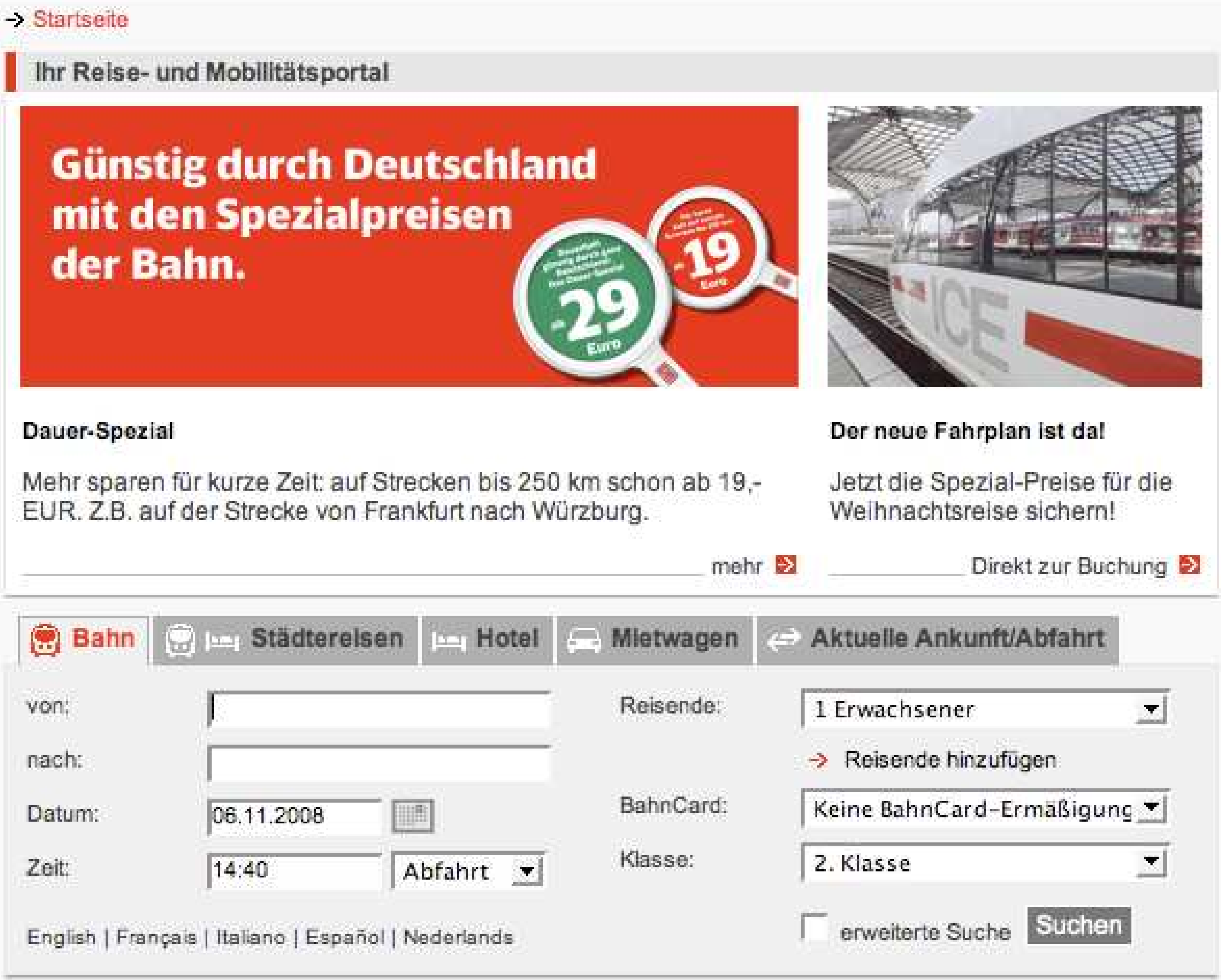
Optimale Zugverbindungen

(graphentheoretische Modellierung)

- Wahl einer Zugverbindung nach Kosten, Fahrzeit (minimale Gesamtfahrzeit, früheste Abfahrt nach Sollzeit, späteste Ankunft vor Sollzeit), Anzahl der Umstiege.
- Zielkriterien im Konflikt.
- Konzentration auf modifiziertes kürzeste Wegeproblem und Lösung durch Dijkstra Algorithmus

  (→ Internetgestütztes Auskunftssystem der Deutschen Bahn und von Privatbahnen)

Optimale Zugverbindungen (1)



Übersetzung der Fahrplansuche in ein kürzestes Wegeproblem

Problem

- Ein Bahnkunde sucht eine günstige Verbindung von **Stadt A** frühestens beginnend mit dem Zeitpunkt **xx Uhr** zur **Stadt B** mit beabsichtigter Ankunft spätestens bis zum Zeitpunkt **yy Uhr**,

  (dies in der Regel am selben Tag und innerhalb Deutschlands mit Zügen der Deutschen Bahn oder „einbezogener“ Züge).

  und ggf. unter **Nebenbedingungen** bzgl. Fahrzeit, Umstiege, Kosten etc.

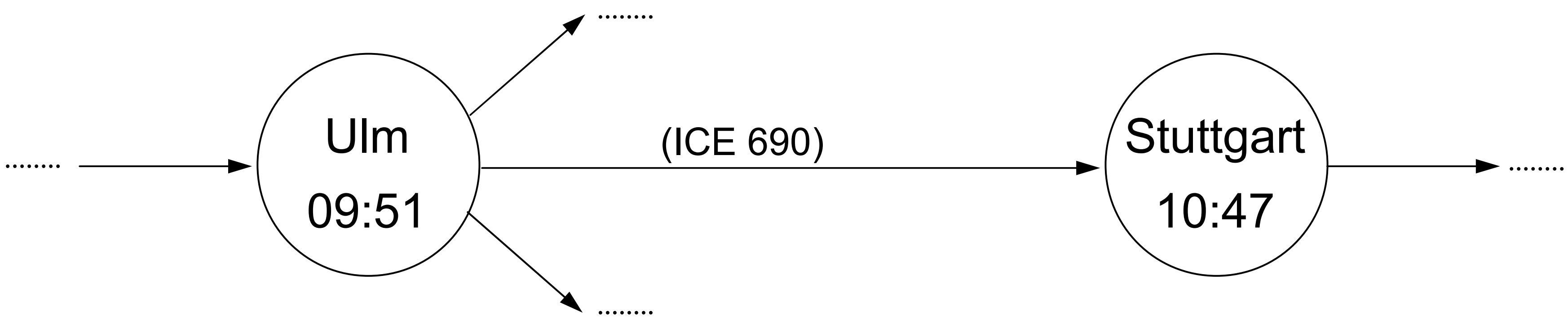
Übersetzung der Fahrplansuche in ein kürzestes Wegeproblem (1)

Definition des Verbindungsgraphen

- Man betrachtet hierzu den aus den Fahrplandaten abgeleiteten (gerichteten) Graphen mit nachfolgend angegeben Knoten und Kanten.
  - Knoten des Graphen sind vom Typ (Bahnhof, Ankunftszeitpunkt) oder vom Typ (Bahnhof, Abfahrtszeitpunkt).
  - Knoten 1 ist genau dann mit einem Knoten 2 durch die gerichtete Kante von 1 nach 2 verbunden, wenn einer der beiden folgenden Fälle vorliegt:

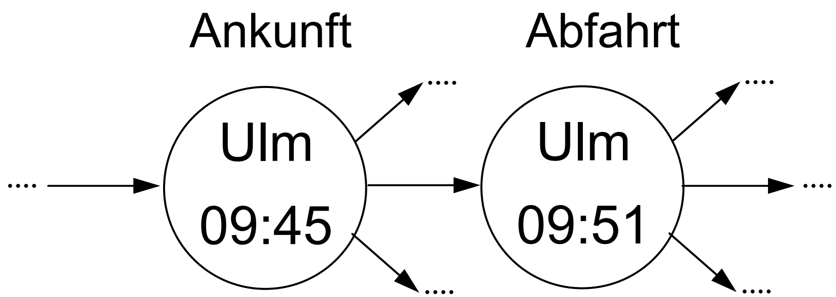
Übersetzung der Fahrplansuche in ein kürzestes Wegeproblem (2)

Fall 1: Wenn sich die beiden Knoten, z.B. Ulm, 09:51 und Stuttgart, 10:47 auf verschiedene Bahnhöfe beziehen und es eine Zugverbindung (hier ICE 690) gibt, die um 09:51 planmäßig Ulm verlässt und um 10:47 Stuttgart erreicht.

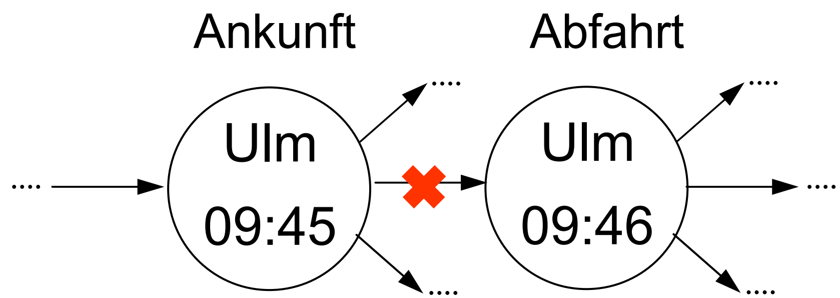


Übersetzung der Fahrplansuche in ein kürzestes Wegeproblem (3)

Fall 2: Wenn sich beide Knoten z.B. Ulm, 09:45* und Ulm, 09:51** auf eine Ankunftszeit und eine spätere Abfahrtszeit in **einem Bahnhof** (inklusive **ausreichender Umstiegszeit**) beziehen.

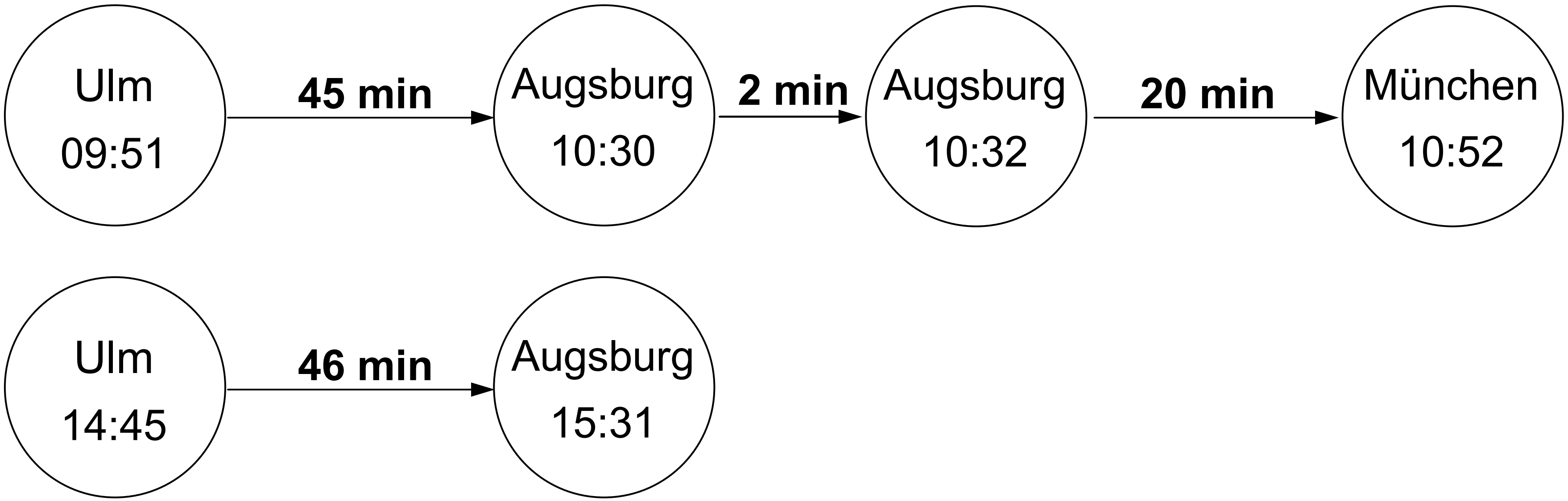


oder



Suche eines schnellsten Weges im Verbindungsgraphen

(gerichtet)



- Ein und derselbe Ort erhält mehrere Knoten im Verbindungsgraphen
- Eine Kante erhält zeitabhängig mehrere Bewertungen im Verbindungsgraphen

Suche eines schnellsten Weges im Verbindungsgraphen (1)

Konsequenz

- Der Verbindungsgraph ist so gross, dass er gar nicht als Graph (z.B. in Form einer Adjazenzmatrix) abgelegt wird.

- Es gibt nur einen Grundgraphen z.B. mit einem Knoten für Ulm, und mit Listen von Zügen (Nummer, Typ), Abfahrt- und Ankunftszeiten.

„Optimale“ Zugverbindung

Bei Vorliegen einer speziellen Verbindungssuche:

- Aus Grundgraph und Liste wird relevanter **Teil des Verbindungsgraphen aufgebaut**

  (Bei beabsichtigter Vermeidung von IC und ICE Verbindungen werden entsprechende Knoten und Kanten nicht in den relevanten Teil des Verbindungsgraph eingefügt)

- Dijkstra Algorithmus bestimmt „kürzesten“ Weg im aufgebauten Teil des Verbindungsgraphen

  ($\rightarrow$ schnellsten Weg)

3. Planare Graphen / Färbungsprobleme

- Landkartenfärbung
- Planare Graphen
- Kantenkontraktion
- Der Vier-Farben-Satz

Landkartenfärbung

➢ Das Problem:

- In Landkarten sind die Länder typischerweise farblich gekennzeichnet.
- Der Übersicht halber stellt man die Forderung, dass benachbarte Länder verschiedene Farben besitzen.
- Eine solche Darstellung nenne man **Färbung**.
- **Ziel**: Verwendung möglichst weniger Farben

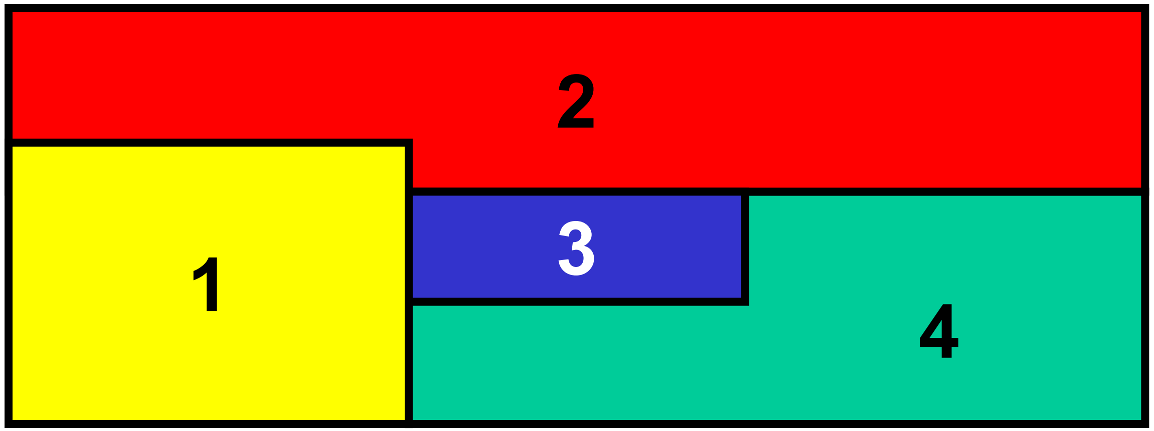
Landkartenfärbung (1)

Beobachtung 1:

- Man braucht mindestens 4 Farben

Jedes der 4 Länder stößt an jedes andere. Man braucht also mindestens 4 Farben.

➢ Beispiel 1:



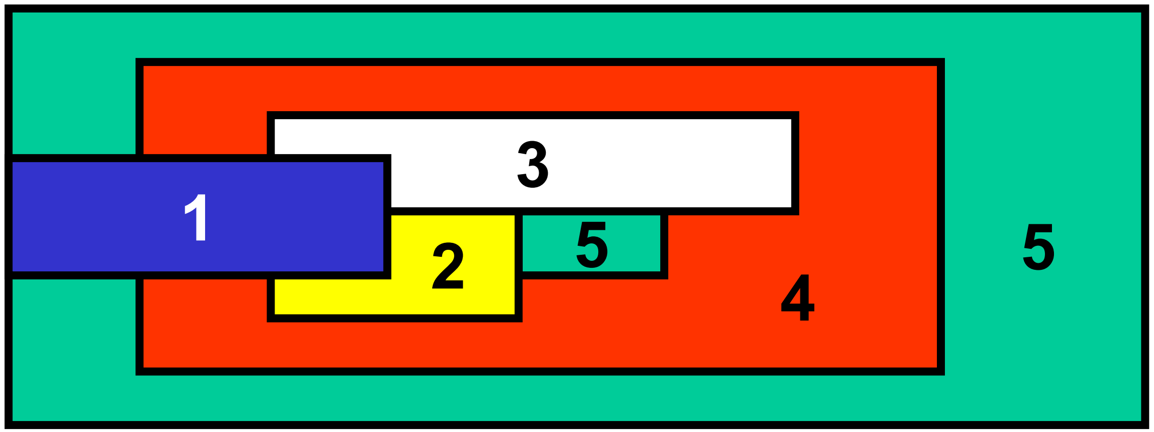
Land 5, das aus zwei Komponenten besteht, ist zu den vier anderen Ländern benachbart, so dass eine fünfte Farbe benötigt wird.

Landkartenfärbung (2)

Beobachtung 2:

- Länder müssen zusammenhängend sein, dürfen also keine abgetrennten „Kolonien“ haben.

➢ Beispiel 2:



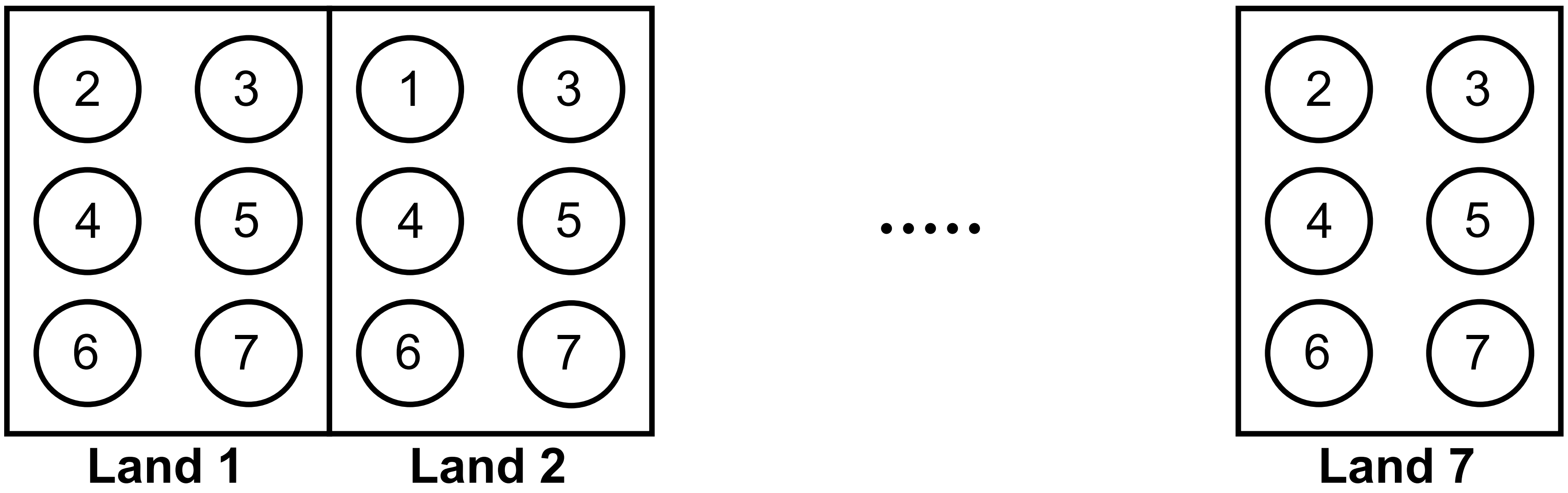
- Hinweis: Für das USA-Beispiel reichen 4 Farben, obwohl die Zusammenhangsforderung nicht erfüllt ist (vgl. Michigan)

Landkartenfärbung (3)

Im allgemeinen lassen sich Beispiele konstruieren, die beliebig viele Farben erfordern

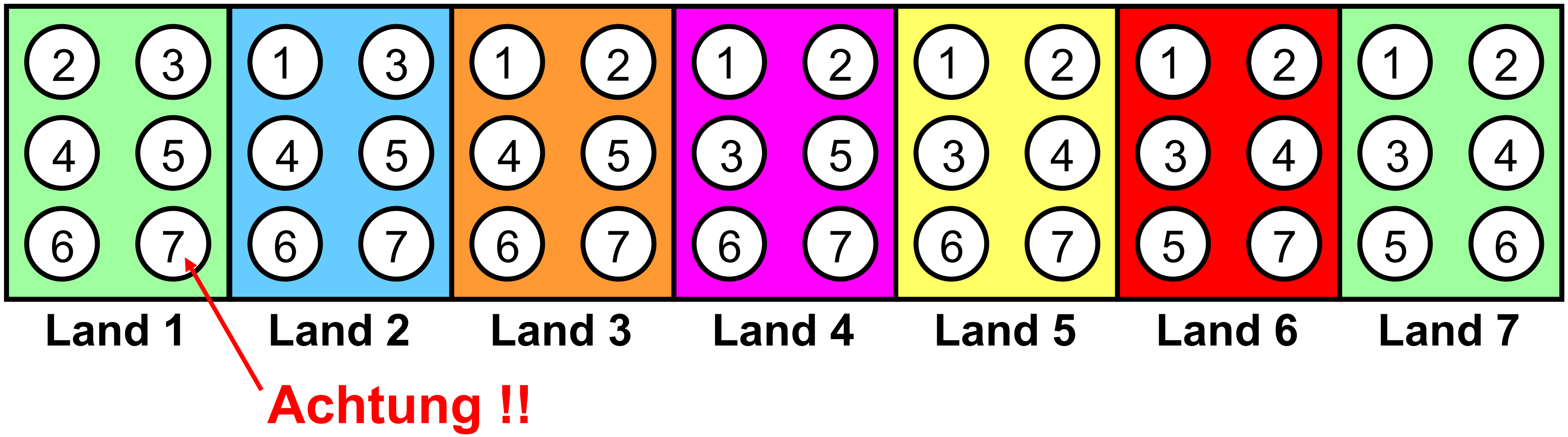
Beispiel

n=7 Länder. Jedes Land hat ein „Hauptgebiet“ und darin liegt eine Kolonie jedes anderen Landes („jedes in jedem“)



Landkartenfärbung (4)

Wenn diese (etwas merkwürdige) Landkarte mit sechs oder weniger Farben färbbar wäre, hätten mindestens zwei Länder dieselbe Farbe, z.B. Land 1 und Land 7. Der Einschluss von Land 7 im Hauptgebiet von 1 und Land 1 müssen aber verschieden gefärbt sein $\Rightarrow$ **Widerspruch**



Färbungsprobleme und Graphen

- Beobachtung 3:
  - Das Färbungsproblem kann in Form von Graphen, sog. Landkartengraphen, modelliert werden.
  - Hierzu stellt man sich vor, dass in der „Mitte“ jedes zusammenhängenden(!) Landes ein Knoten platziert wird und dass zwei Knoten genau dann durch eine Kante miteinander verbunden werden, wenn zwischen den Ländern eine gemeinsame Grenze (also mehr als ein Punkt) besteht.

- Beispiel 3:

- Färbungsproblem für Graphen:
  - Alle Knoten sind so zu färben, dass je zwei durch eine Kante verbundene Knoten verschiedene Farben besitzen.

Färbungsproblem für Graphen (1)

- Beobachtung 4:
  - Jeder Landkartengraph ist planar, also in der Ebene so darstellbar, dass sich Kanten nur in den Knoten schneiden.

- Beobachtung 5:
  - Es gilt auch die Umkehrung, d. h. zu jedem planaren Graph lässt sich ein „äquivalentes“ Landkartenproblem ableiten.

Der Vier-Farben-Satz

- Hinweis:
  - Die 5-Färbbarkeit von Landkarten bzw. planaren Graphen ist schon seit langem bekannt.

    Heawood, 1890.

  - Die 4-Färbbarkeit von Landkarten mit maximal 96 Ländern bzw. von planaren Graphen mit maximal 96 Knoten war bis 1970 nachgewiesen.

Der Vier-Farben-Satz (1)

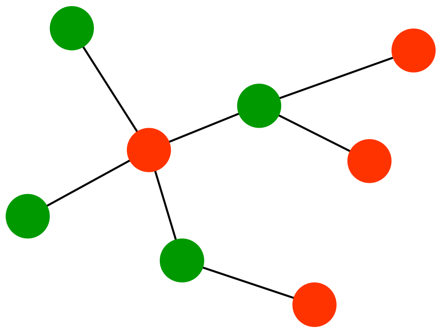
- Vier-Farben-Satz
  - 1976 bewiesen Appel und Haken, dass 4 Farben generell genügen.

- Der Beweis(!) stützte sich auf Computerberechnungen.
  - Zunächst wurde mit theoretischen Argumenten die Beweisfrage für unendlich viele Landkarten auf ca. 10.000 Standard Fälle reduziert.
  - In jedem der Standardfälle wurde die 4-Färbbarkeit nachgewiesen.
  - Inzwischen konnten die Standardfälle auf ca. 600 reduziert werden. Der Einsatz von Computern ist jedoch immer noch erforderlich.
  - Der „Beweis“ wurde als Beweis akzeptiert, obwohl kein Mensch die Argumentation vollständig nachvollziehen kann.

Der Vier-Farben-Satz (2)

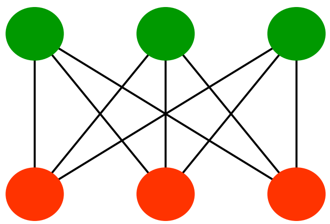
Es gibt sowohl planare als auch nicht-planare Graphen, die mit weniger als 4 Farben färbbar sind.

- Baum



2 Farben genügen

- $K_{3,3}$



2 Farben genügen auch hier

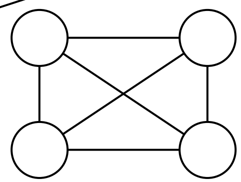
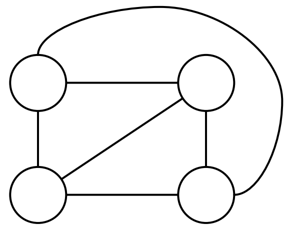
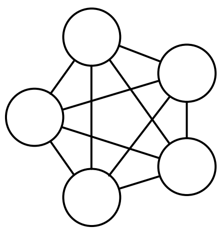
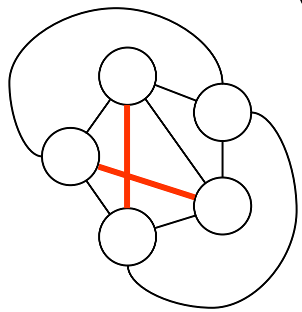
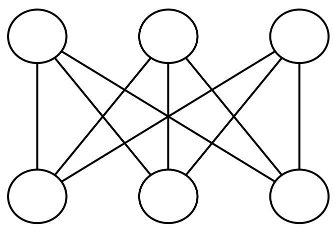
Planare Graphen

➢ Definition:

- Ein Graph heißt **planar**, wenn er so gezeichnet werden kann (ggf. mit gebogenen Kanten), so dass sich keine Kanten überkreuzen.

Graph mit 4 Knoten, bei dem jeder Knoten mit jedem verbunden ist.

➢ Beispiel:

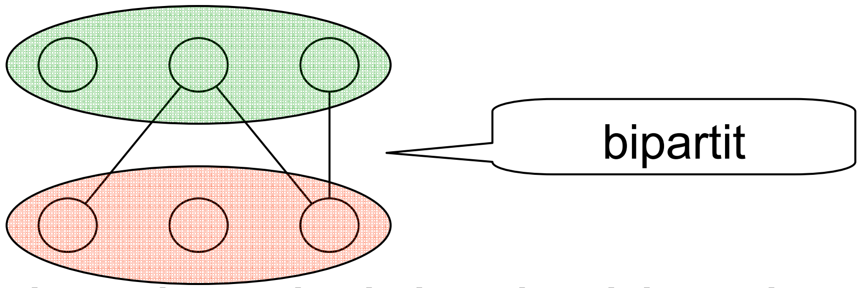
- K_4:  **ist** planar, denn er kann wie folgt gezeichnet werden: 
- K_5:  **ist nicht** planar: 
- $K_{3,3}$:  **ist nicht** planar.

Planare Graphen (1)

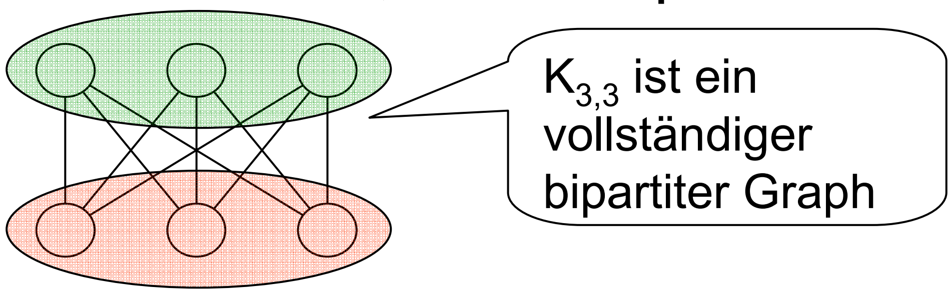
- Ein Graph G=(V,E) heißt **bipartit**, wenn die Knotenmenge aus zwei nicht-leere disjunkte Teilmengen besteht, also

$$V = V_1 \cup V_2 \text{ mit } V_1 \cap V_2 = \emptyset$$

  so dass $E \subseteq V_1 \times V_2$, d.h. Kanten nur zwischen V_1 und V_2 verlaufen.

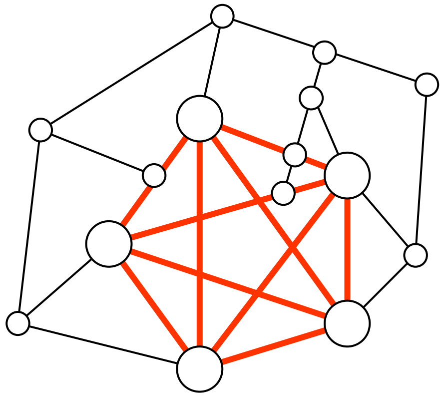


- Wenn alle solche Kanten vorhanden sind, ist der bipartite Graph **vollständig bipartit**.



Planare Graphen (2)

- Satz (Kuratowsky):
  - Ein Graph ist genau dann nicht planar, wenn er einen K_5 oder $K_{3,3}$ „als Teilgraph enthält“.



→ Es gibt bessere Algorithmen zum Testen, ob ein Graph planar ist oder nicht. Diese Algorithmen beruhen **nicht** auf Tests des Vorhandenseins von K_5 und $K_{3,3}$.

Kantenkontraktion

Problem:

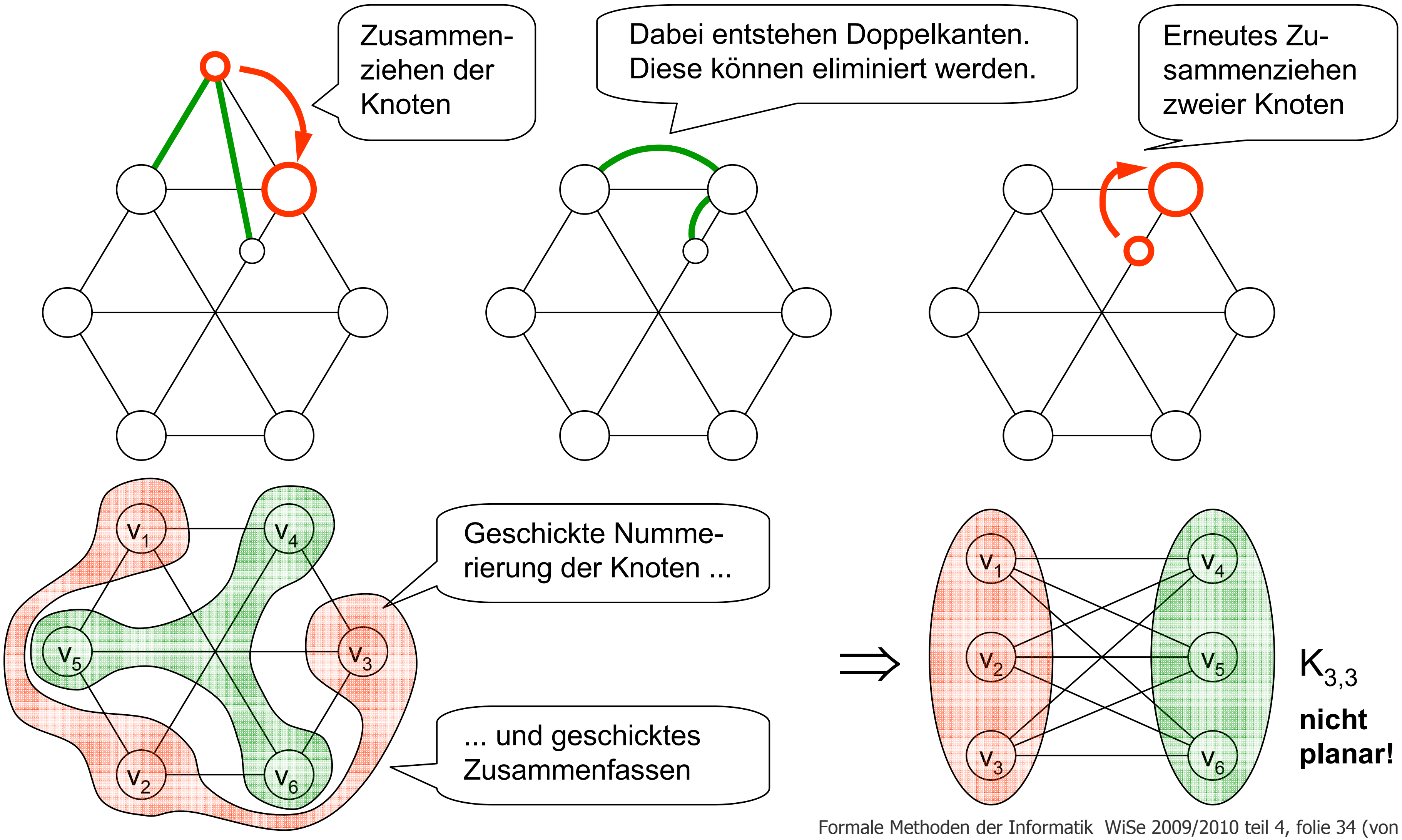
- Wie erkennt man einen Teilgraphen im Graphen?

- "Teilgraphen" ergeben sich über eine Folge von **Kantenkontraktionen**.
    - Gleichsetzen der beteiligten Knoten
    - ggf. Streichen von Mehrfachkanten

- Auswahl der Knoten, die kontrahiert werden, **nicht trivial**!

Kantenkontraktion (1)

➢ Beispiel: Ist der Graph planar?

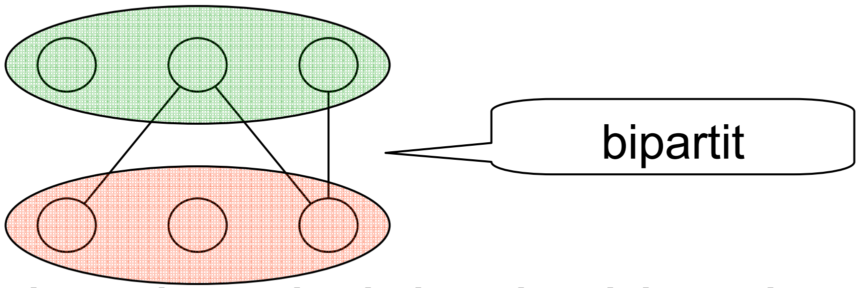


Bipartite Graphen

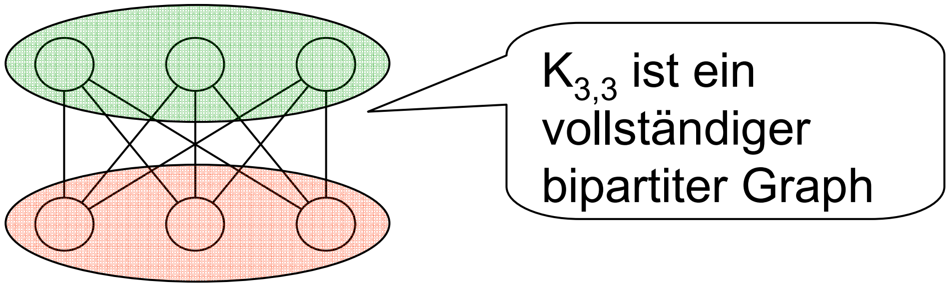
- Ein Graph G=(V,E) heißt **bipartit**, wenn die Knotenmenge aus zwei nicht-leere disjunkte Teilmengen besteht, also

$$V = V_1 \cup V_2 \text{ mit } V_1 \cap V_2 = \emptyset$$

so dass $E \subseteq V_1 \times V_2$, d.h. Kanten nur zwischen V_1 und V_2 verlaufen.



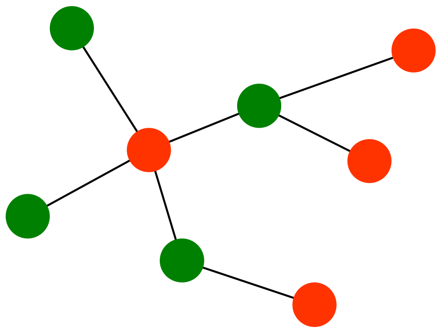
- Wenn alle solche Kanten vorhanden sind, ist der bipartite Graph **vollständig bipartit**.



Bipartite Graphen (1)

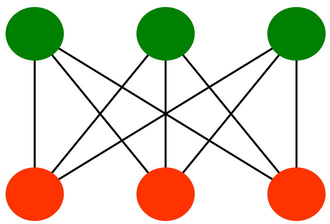
Es gibt sowohl planare als auch nicht-planare Graphen, die mit weniger als 4 Farben färbbar sind.

- Baum



2 Farben genügen

- $K_{3,3}$



2 Farben genügen auch hier

Chromatische Zahl

- Die minimale Anzahl von Farben zur Färbung eines beliebigen ungerichteten Graphen (d. h. benachbarte Knoten erhalten verschiedene Farben) heißt **chromatische Zahl** des Graphen. Abkürzung $\chi(G)$.
  - $\chi(G) = 2$, falls G Baum oder bipartit.
  - $\chi(G) \leq 4$, falls G planar.
- Algorithmus zur Bestimmung von $\chi(G)$?

Chromatische Zahl (1)

Naiver Ansatz

- Probiere, ob ein Graph mit n Knoten mit $k = 2, \ldots, n - 1$ Farben färbbar ist.
  (Mit n Farben ist immer Färbbarkeit gegeben).
- Z. B. $k = 4$ Farben
  - Erzeugung aller Zuordnungen von Farben zu Knoten.
    Davon gibt es $\underbrace{4 \cdot \ldots \cdot 4}_{\text{Knotenzahl}} = 4^n$ viele.
  - Für jede Zuordnung wird getestet, ob sie Färbung ist.
- Wenn es Färbung mit 4 Farben gibt, brauchen nicht alle Zuordnungen getestet werden.
- Wenn es keine Färbung mit 4 Farben gibt, benötigt man mindestens 4^n Berechnungsschritte.

Chromatische Zahl (2)

Naiver Ansatz 2

- Es geht etwas besser, z. B. erhält ein bestimmter Knoten Farbe 1. Dann bleiben 4^{n-1} Zuordnungen.
- Man kann sich noch weitere Verbesserungen überlegen, aber im schlimmsten Fall bleibt die Anzahl der Berechnungsschritte **exponentiell** in n.

Komplexität

➢ "Effizienter" Algorithmus

- Die Laufzeit in Abhängigkeit zur Eingabegröße n (z.B. Anzahl der Knoten oder die Seitenlänge einer Matrix) sollte ein Polynom in n sein (d.h. Laufzeit $\leq n^k$, mit k konst.)

- Beispiele: Kruskal-, Dijkstra-, Gradsequenz-Algorithmus

➢ "Ineffizienter" Algorithmus

- Die Laufzeit steigt exponentiell zu n an, z.B. 2^n oder $n! \approx 2^{n \cdot \log n}$

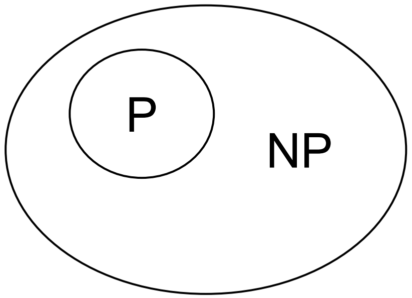
- Beispiele: Zum Isomorphieproblem und zur Bestimmung der Chromatischen Zahl sind nur ineffiziente Algorithmen bekannt.

Komplexität (1)

- Sei 1 Rechenschritt ≈ 10^{-10} sec

	Komplexität	n = 10	n = 100	n = 1000
Algor. 1	n^2	10^{-8} sec	10^{-6} sec	10^{-4} sec
Algor. 2	2^n	10^{-7} sec	10^{20} sec $3 \cdot 10^{12}$ Jahre	...

- P = Menge der effizient lösbaren Probleme
- NP = Menge aller effizient lösbaren und aller bis heute als ineffizient lösbar geltenden Probleme



$P \subsetneq NP$
oder
$P = NP$?