Algorithmen und Datenstrukturen

Große Übung vom 26.10.07

Nils Schweer

Diese Folien

- Braucht man nicht abschreiben
- Stehen im Netz unter

  www.ibr.cs.tu-bs.de/courses/ws0708/aud/index.html

Kleine Übungen

- Anmeldung online auf

  www.ibr.cs.tu-bs.de/courses/ws0708/aud/index.html
- Ca. 15-20 Teilnehmer pro Übung
- Das Online Anmeldeverfahren gibt Auskunft über den Erfolg der Anmeldung
  - Sie haben sich erfolgreich angemeldet
  - Jeder der diese Meldung bekommen hat, ist in der entsprechenden Gruppe
- Probleme?
- Nachmeldungen am Ende dieser Übung

Kleine Übungen

- Mittwoch 13:15 - 14:45 PK3.2 , PK3.3
  15:00 - 16:30 PK3.2 , PK3.3
  16:45 - 18:15 PK3.2 , PK3.3
  18:30 - 20:00 PK3.3

- Donnerstag 08:00 - 09:30 RR58.1, RR58.2
  09:45 - 11:15 RR58.1
  13:15 - 14:45 SN19.7, RR58.2
  15:00 -16:30 PK14.3, BW74.3

Mailingliste

- Natürlich auf: www.ibr.cs.tu-bs.de/courses/ws0708/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

- 6 Hausaufgabenblätter
- 14-tägig auf www.ibr.cs.tu-bs.de/courses/ws0708/aud/index.html
- Bearbeitungszeit: 14 Tage
- Abgabe: Mittwochs vor der Vorlesung im SN 19.1
- Rückgabe: In den kleinen Übungen
- 50% der Hausaufgabenpunkte sind Voraussetzung für Klausurteilnahme (für alle Bachelor- / Masterstudenten/innen)
- Besprechung der Lösungen in den kleinen Übungen

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!

Fortsetzung der Rundreise

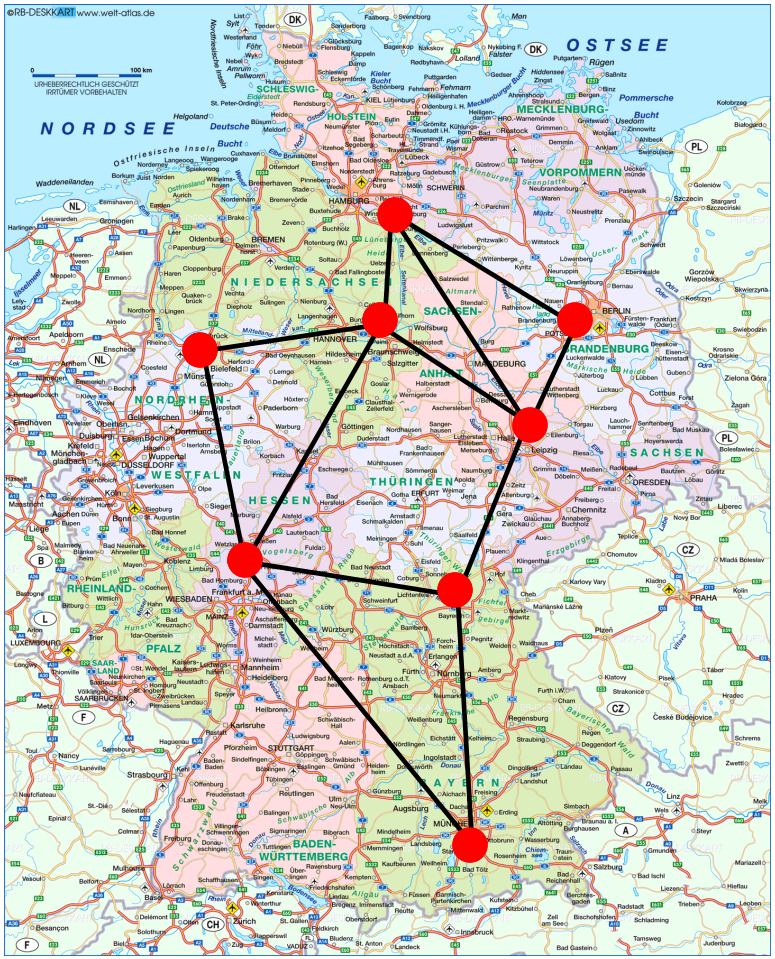
- Problem: Finde eine Rundreise, (d.h. eine Route die jeden Ort besucht und zum Startort zurückkehrt) die möglichst kurz ist.

Fortsetzung der Rundreise

Fortsetzung der Rundreise

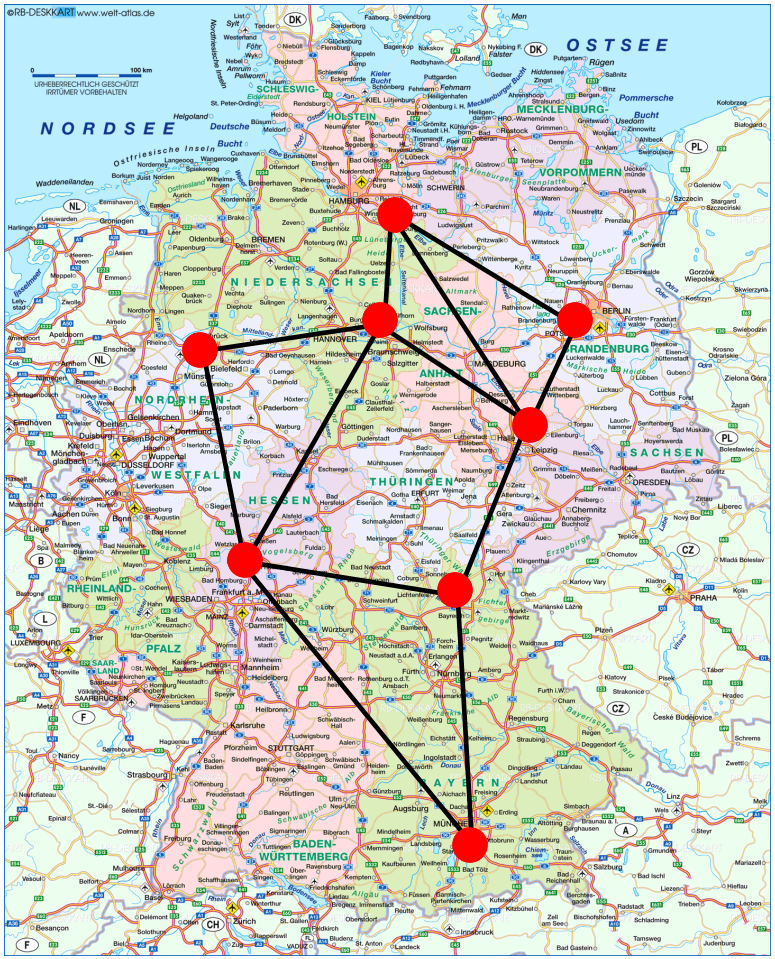
- Wie funktioniert map24.de ?
- Man kann zwischen den Städten beliebig hin und her reisen.

Phase 1: Modellierung



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

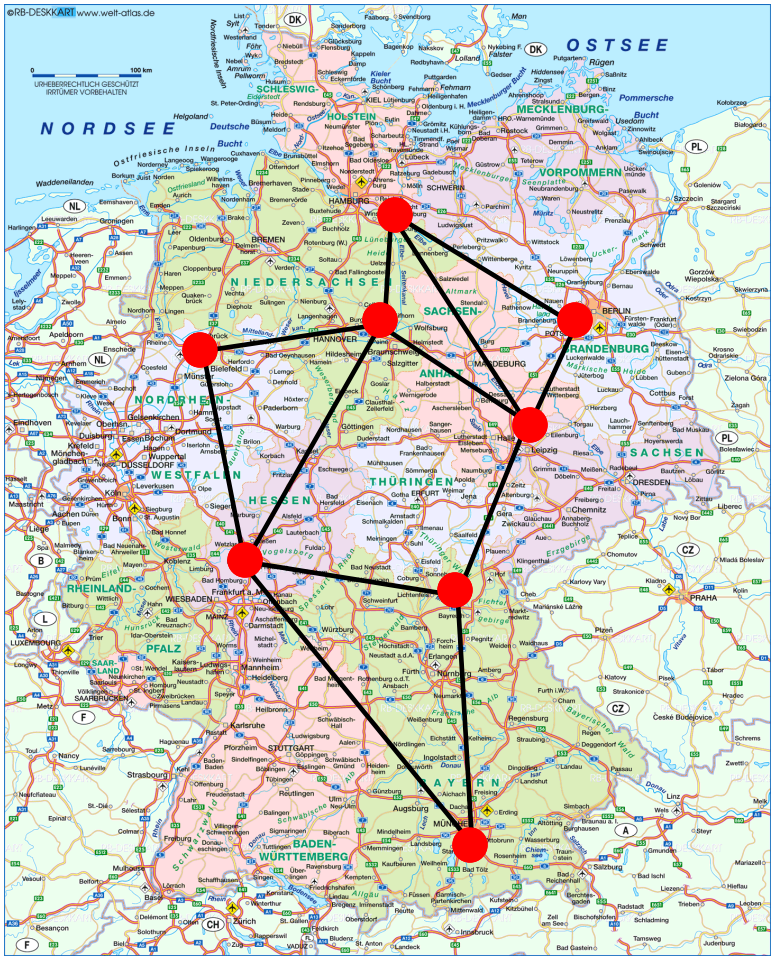
Phase 1: Modellierung



- Knoten:
  - Hamburg
  - Berlin
  - Braunschweig
  - Leipzig
  - ……
- Kanten:
  - {Hamburg, Berlin}
  - {BS, Berlin}
  - {BS, Leipzig}
  - ……

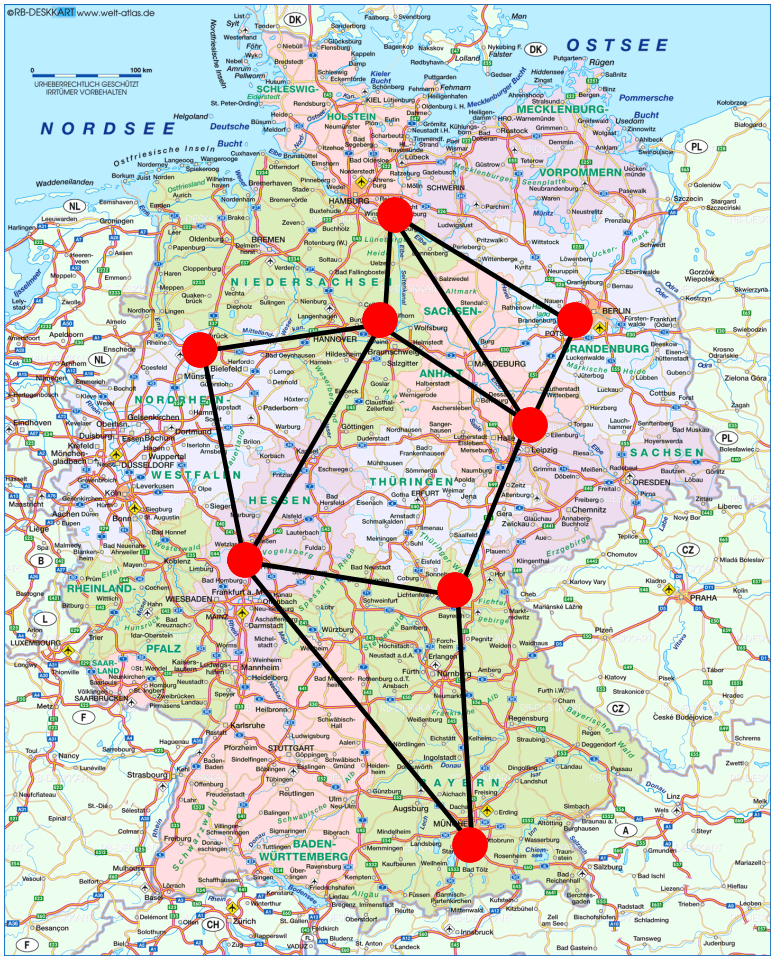
Typische Schreibweise

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



- Eine anwendbare Datenstruktur sind z.B. Listen
- Speichere alle „benachbarten Orte“ in einer Liste
  - BS_Nb:

    (Hamburg, Leipzig, Münster, Frankfurt,…)
- Die Entfernungen in einer anderen:
  - BS_Ent:

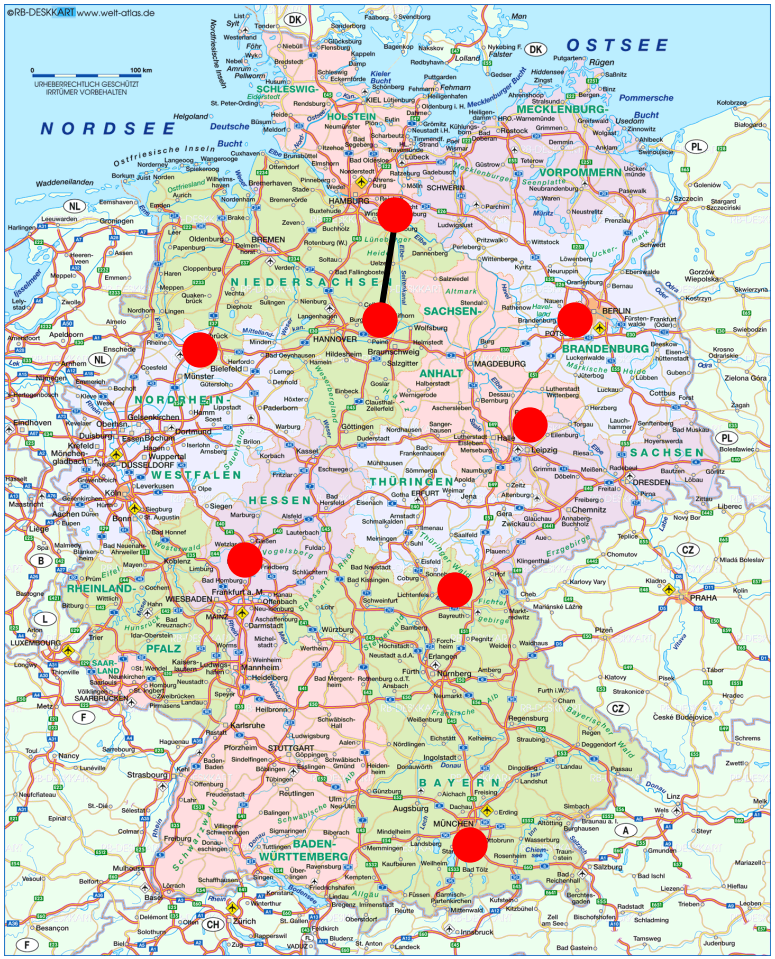
    (220, 200, 250,…)

Phase 2: Datenstruktur

- Eine anwendbare Datenstruktur sind z.B. Listen
- Speichere alle „benachbarten Orte“ in einer Liste:
  (Hamburg, Leipzig, Münster, Frankfurt,…)
- Die Entfernungen in einer anderen:
  - BS_Ent:
    (220, 200, 250,…)

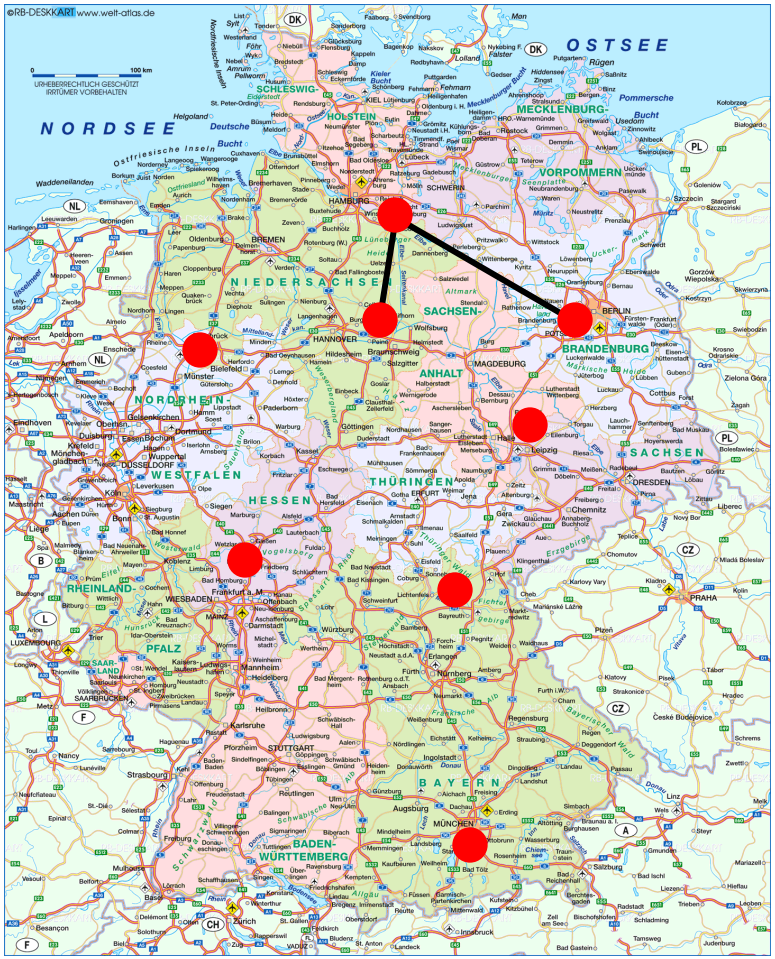
Bald in dieser Vorlesung

Phase 3: Algorithmen



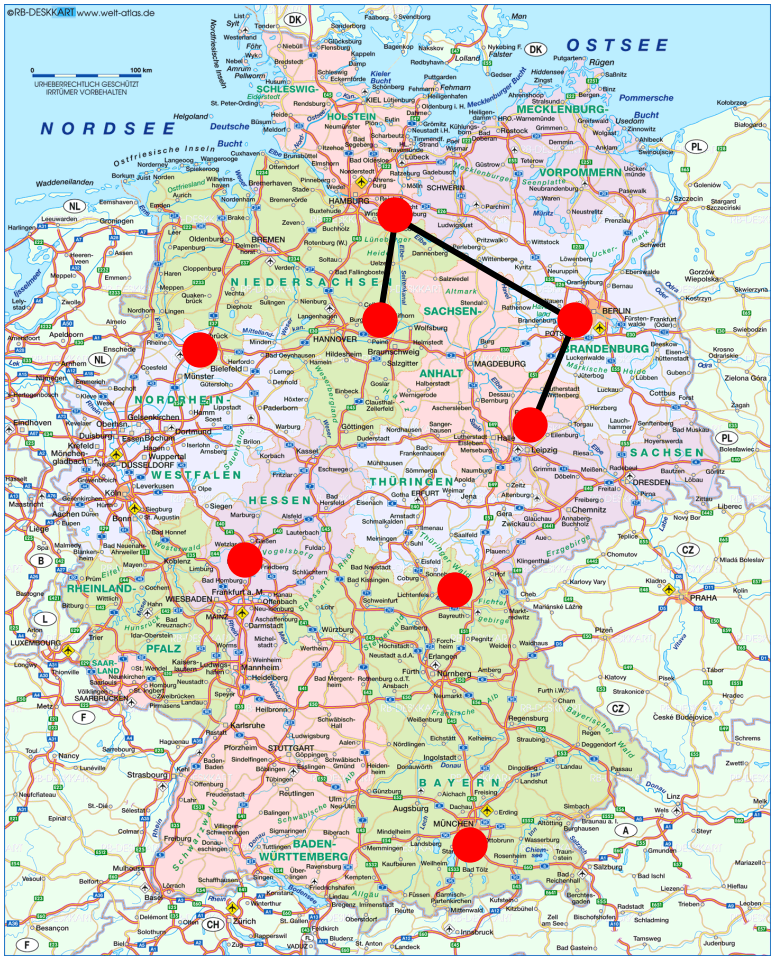
- Es ist „schwierig“ die beste Lösung zu finden
- Aber man kann schnell „ziemlich gute“ finden
- Algorithmus:
  - Wähle bel. Startort
  - Gehe zum nächstgelegenen Ort außer, dieser ist bereits besucht

Phase 3: Algorithmen



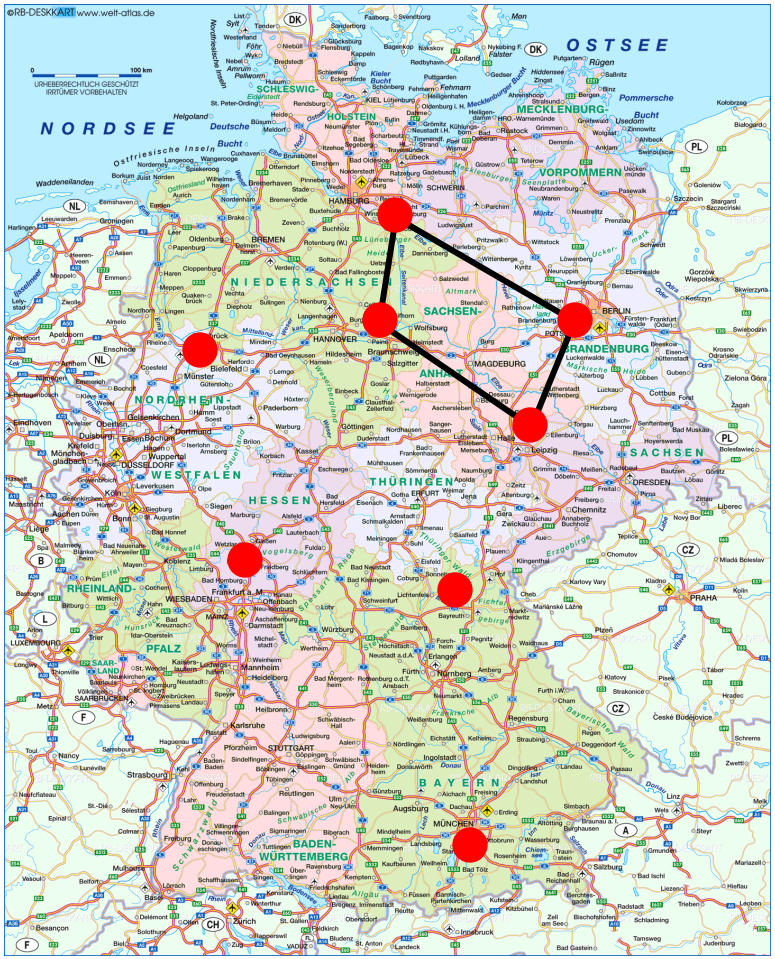
- Es ist „schwierig“ die beste Lösung zu finden
- Aber man kann schnell „ziemlich gute“ finden
- Algorithmus:
  - Wähle bel. Startort
  - Gehe zum nächstgelegenen Ort außer, dieser ist bereits besucht

Phase 3: Algorithmen



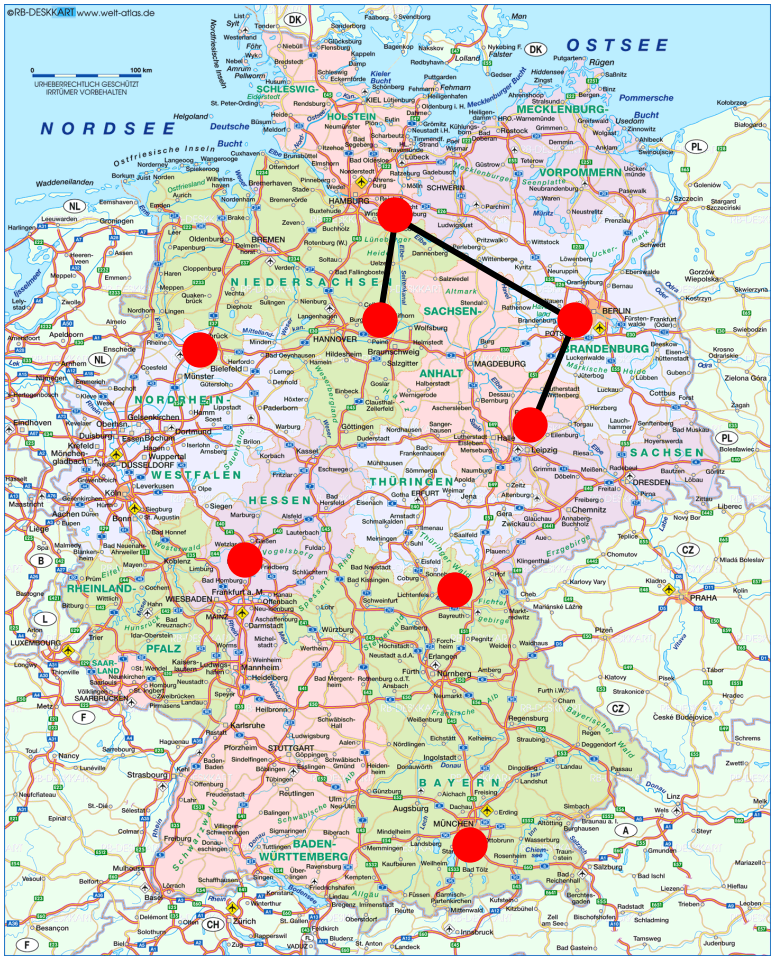
- Es ist „schwierig“ die beste Lösung zu finden
- Aber man kann schnell „ziemlich gute“ finden
- Algorithmus:
  - Wähle bel. Startort
  - Gehe zum nächstgelegenen Ort außer, dieser ist bereits besucht

Phase 3: Algorithmen



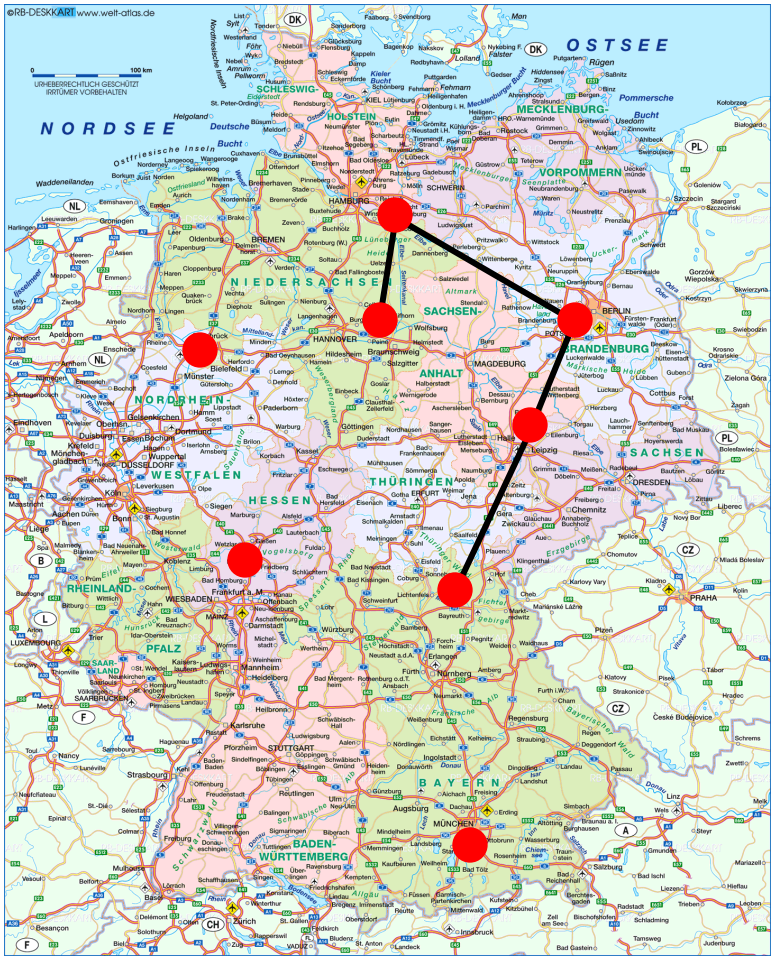
- Es ist „schwierig“ die beste Lösung zu finden
- Aber man kann schnell „ziemlich gute“ finden
- Algorithmus:
  - Wähle bel. Startort
  - Gehe zum nächstgelegenen Ort außer, dieser ist bereits besucht

Phase 3: Algorithmen



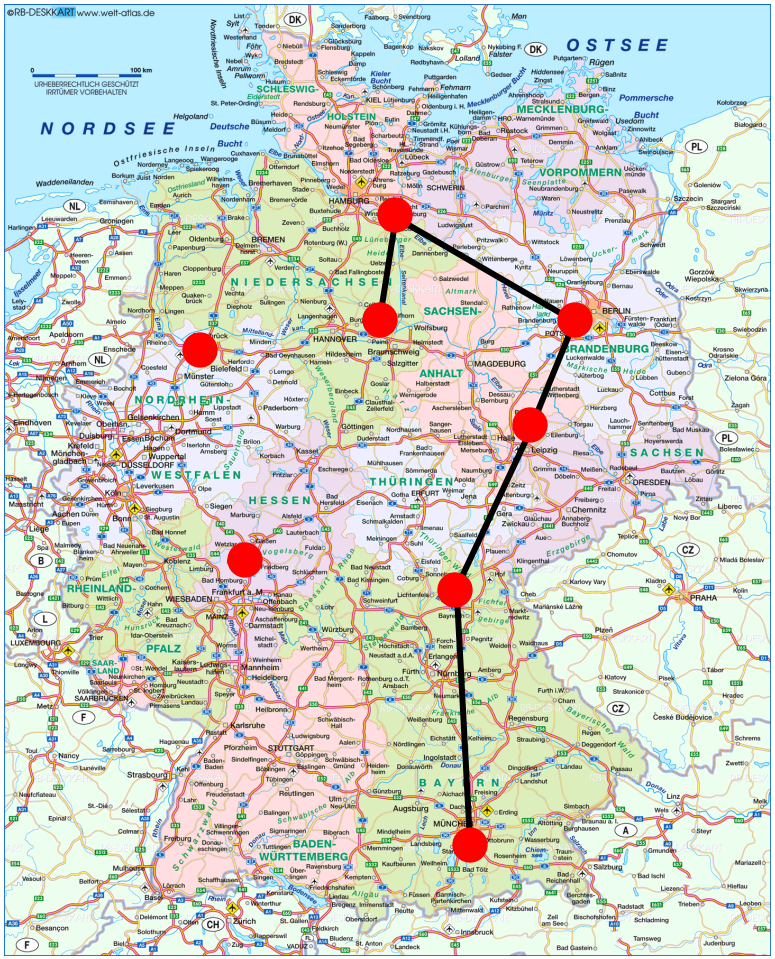
- Es ist „schwierig“ die beste Lösung zu finden
- Aber man kann schnell „ziemlich gute“ finden
- Algorithmus:
  - Wähle bel. Startort
  - Gehe zum nächstgelegenen Ort außer, dieser ist bereits besucht

Phase 3: Algorithmen



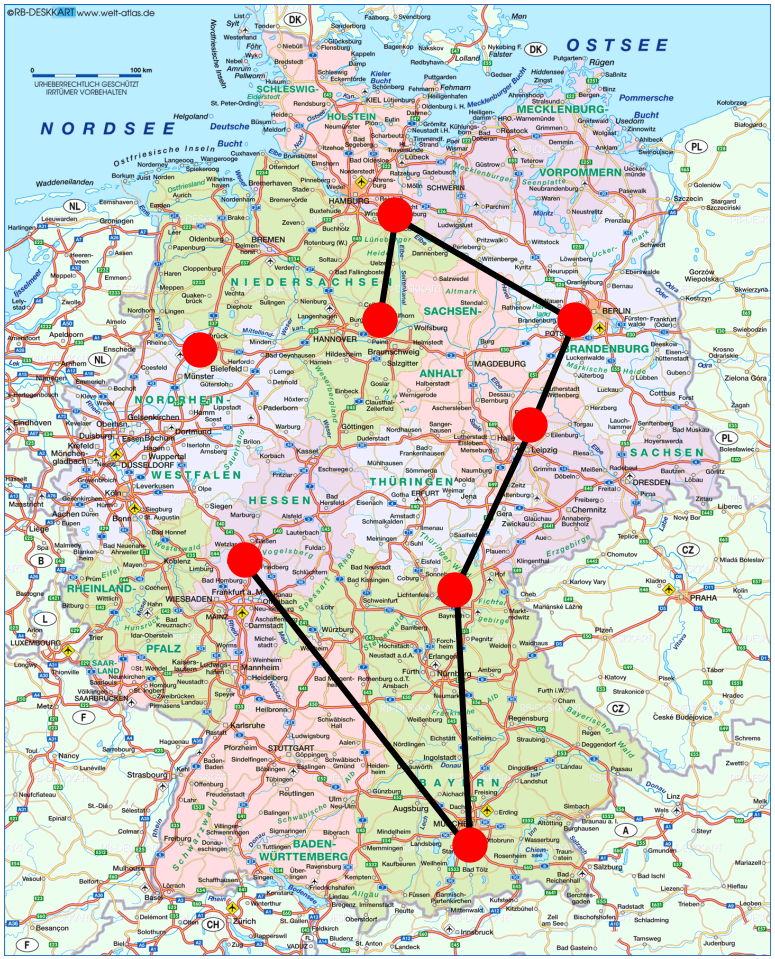
- Es ist „schwierig“ die beste Lösung zu finden
- Aber man kann schnell „ziemlich gute“ finden
- Algorithmus:
  - Wähle bel. Startort
  - Gehe zum nächstgelegenen Ort außer, dieser ist bereits besucht

Phase 3: Algorithmen



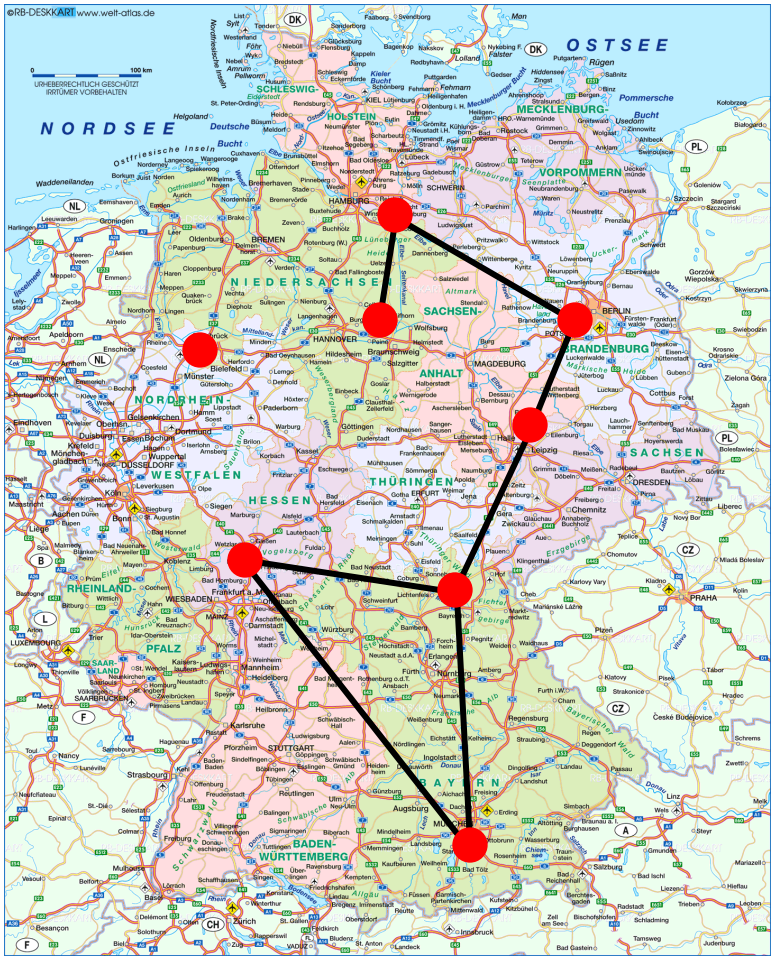
- Es ist „schwierig“ die beste Lösung zu finden
- Aber man kann schnell „ziemlich gute“ finden
- Algorithmus:
  - Wähle bel. Startort
  - Gehe zum nächstgelegenen Ort außer, dieser ist bereits besucht

Phase 3: Algorithmen



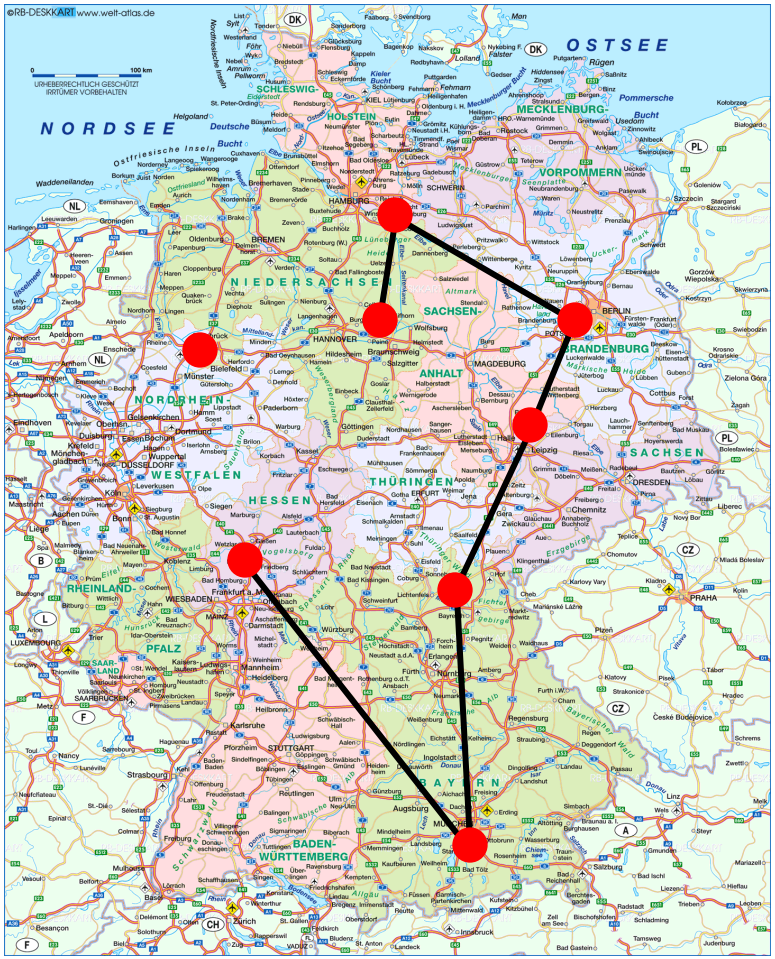
- Es ist „schwierig“ die beste Lösung zu finden
- Aber man kann schnell „ziemlich gute“ finden
- Algorithmus:
  - Wähle bel. Startort
  - Gehe zum nächstgelegenen Ort außer, dieser ist bereits besucht

Phase 3: Algorithmen



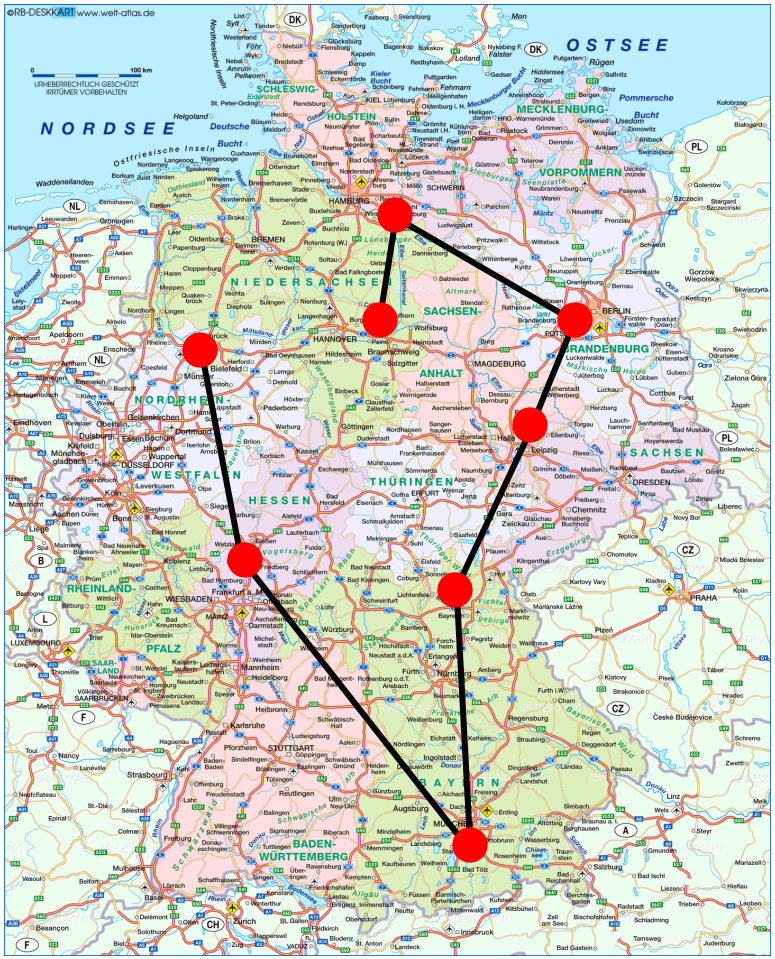
- Es ist „schwierig“ die beste Lösung zu finden
- Aber man kann schnell „ziemlich gute“ finden
- Algorithmus:
  - Wähle bel. Startort
  - Gehe zum nächstgelegenen Ort außer, dieser ist bereits besucht

Phase 3: Algorithmen



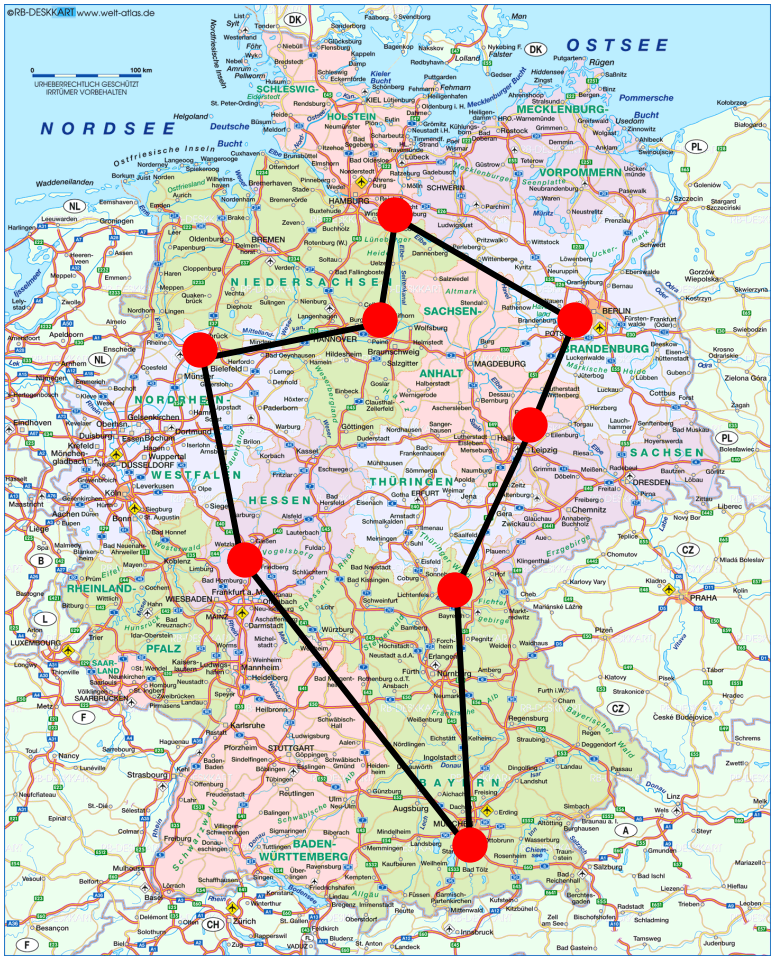
- Es ist „schwierig“ die beste Lösung zu finden
- Aber man kann schnell „ziemlich gute“ finden
- Algorithmus:
  - Wähle bel. Startort
  - Gehe zum nächstgelegenen Ort außer, dieser ist bereits besucht

Phase 3: Algorithmen



- Es ist „schwierig“ die beste Lösung zu finden
- Aber man kann schnell „ziemlich gute“ finden
- Algorithmus:
  - Wähle bel. Startort
  - Gehe zum nächstgelegenen Ort außer, dieser ist bereits besucht

Phase 3: Algorithmen



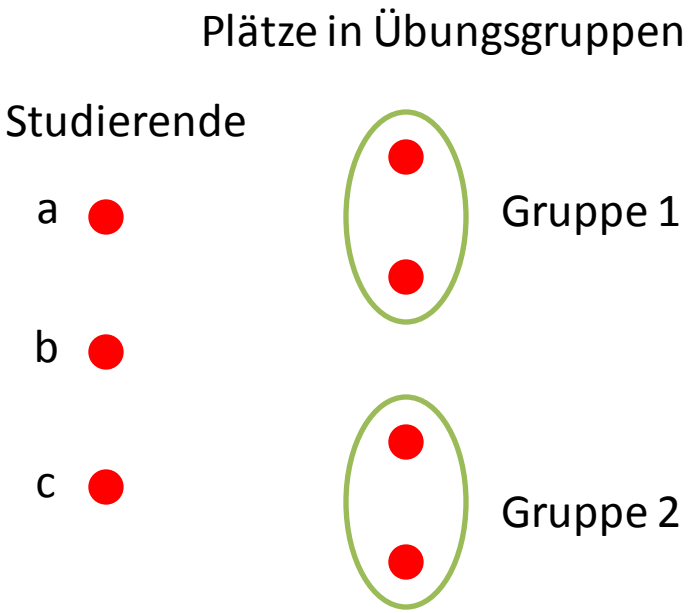
- Es ist „schwierig“ die beste Lösung zu finden
- Aber man kann schnell „ziemlich gute“ finden
- Algorithmus:
  - Wähle bel. Startort
  - Gehe zum nächstgelegenen Ort außer, dieser ist bereits besucht

Phase 3: Algorithmen

- Es ist „schwierig“ die beste Lösung zu finden
- Aber man kann schnell „zi[illegible]ute“ finden
  [illegible]ithmus:
  - Wähle bel. Startort
  - Gehe zum nächstgelegenen Ort außer, dieser ist bereits besucht

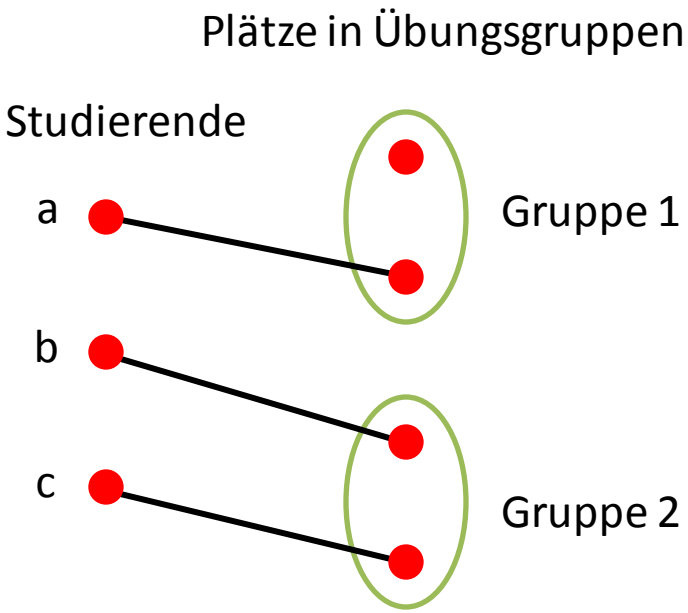
Bald in dieser Vorlesung

Das Zuordnungsproblem



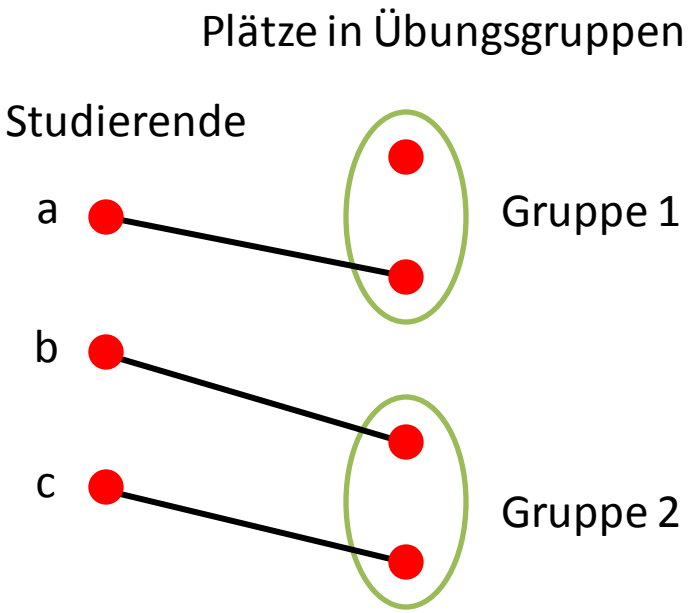
- Problem: Zuordnung von Studierenden auf Übungsgruppen
- Konkret:
  - 220 Studierende
  - 14 Übungsgruppen à 20 Plätze
- Ein kleineres Beispiel:
  - 3 Studierende
  - 2 Übungsgruppen à 2 Plätze

Das Zuordnungsproblem



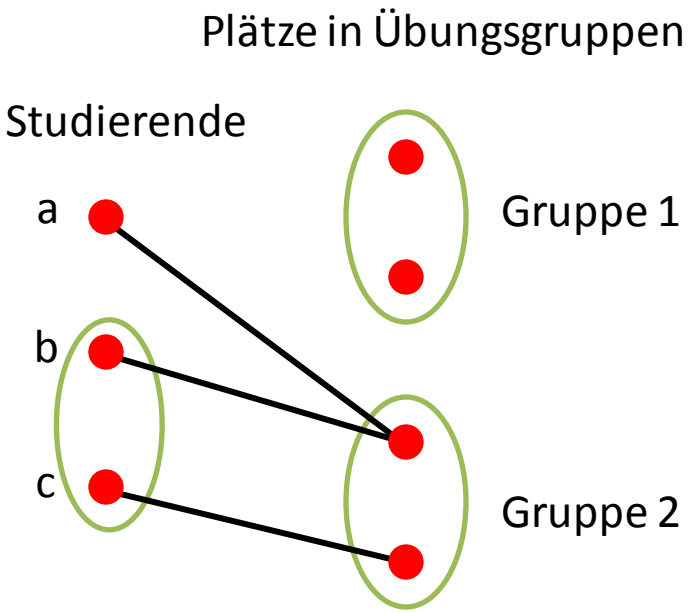
- Die Studierenden wählen jeweils einen Platz

Das Zuordnungsproblem



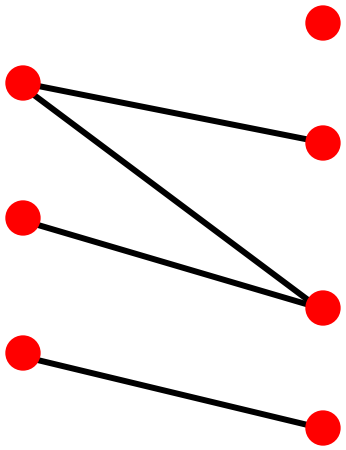
- Die Studierenden wählen jeweils einen Platz
- Aufteilung möglich
- 3 < 4 reicht das?
- Natürlich nicht!

Das Zuordnungsproblem



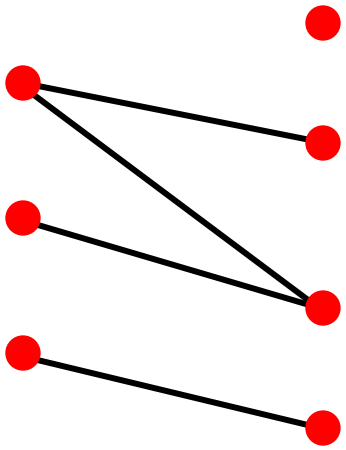
- Die Studierenden wählen jeweils einen Platz
- Aufteilung möglich
- 3 < 4 reicht das?
- Natürlich nicht!
- Wenn z.B. *a* anders wählt
- Nur 2 bekommen einen Platz

Graphen



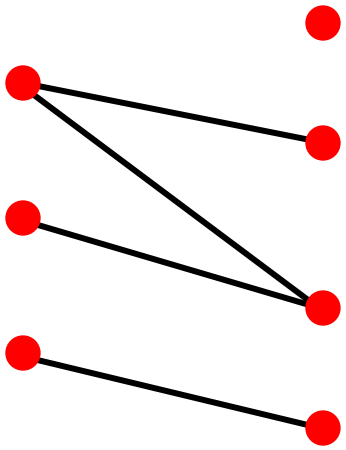
- Fazit: Graphen sind vielseitig anwendbar
- Deshalb lohnt es sich, sie genauer zu betrachten / ihre Eigenschaften zu studieren

Graphen



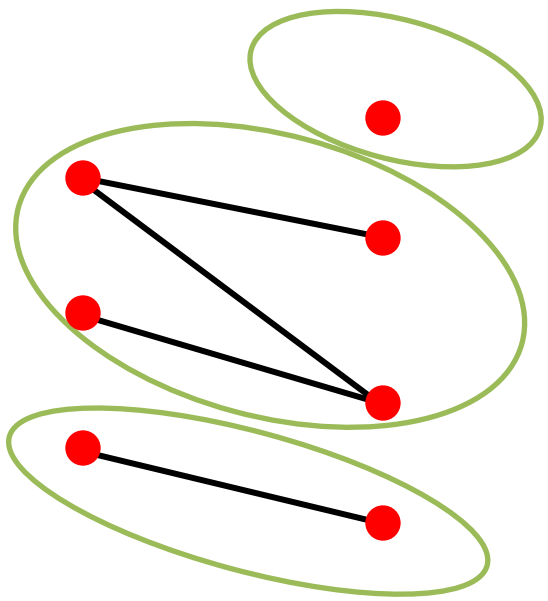
- Knoten / Kanten

Graphen



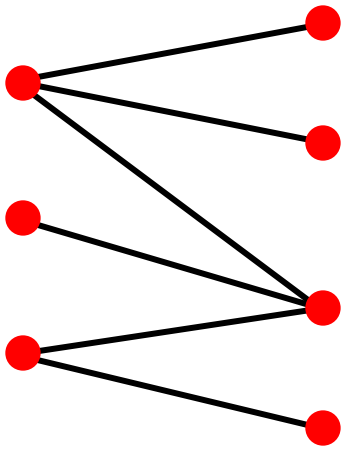
- Knoten / Kanten
- Intuitive Begrifflichkeiten, z.B. „zusammenhängend“

Graphen



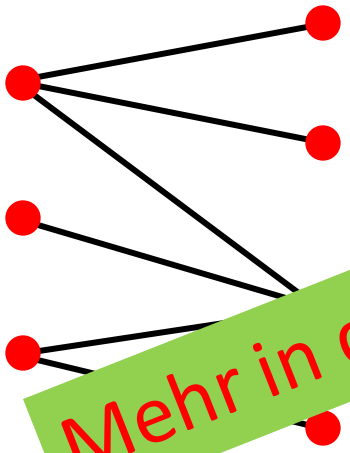
- Knoten / Kanten
- Intuitive Begrifflichkeiten, z.B. „zusammenhängend“
- 3 Teile, also nicht „zusammenhängend“

Graphen



- Knoten / Kanten
- Intuitive Begrifflichkeiten, z.B. „zusammenhängend“
- 3 Teile, also nicht „zusammenhängend“
- Nun ist der Graph „zusammenhängend“

Graphen



- Knoten / Kanten
- Intuitive Begriffe, z.B. „zusammenhängend“
- 3 Teile, also nicht „zusammenhängend“
- Nun ist der Graph „zusammenhängend“

Mehr in der nächsten Vorlesung…

Schönes Wochenende….

… bis in 14 Tagen

Freitag

Essen		Stud.	Bed.	Gäste
A: Terrine Tomatenrahmsuppe (4) mit Einlage	V	1,20	2,50	3,00
B: Putengulasch (4) mit Spätzle		1,50	2,80	3,30
C: Gebackenes Seelachsfilet mit Kartoffelsalat (3)		2,00	3,30	3,80
V: Tofuschnitte mit Tomaten überbacken, Hollandaise (4) dazu Röstkartoffeln	V	1,80	3,10	3,60
W: Truthahnschnitzel mit Austernpilzen, Pommes frites		3,00	4,40	4,90

Beilagen	Stud.	Gäste
Remoulade (3,9)	0,40	0,50
Broccoli	0,40	0,50
Salat	0,50	0,60
Bohnengemüse	0,40	0,50
Ananas	0,40	0,50