Algorithmen und Datenstrukturen

Große Übung vom 29.10.09

Nils Schweer

Diese Folien

- Braucht man nicht abzuschreiben
- Stehen im Netz unter

  www.ibr.cs.tu-bs.de/courses/ws0910/aud/index.html

Kleine Übungen

- Anmeldung online auf

  www.ibr.cs.tu-bs.de/courses/ws0910/aud/index.html
- Ca. 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?

Kleine Übungen

Mittwoch

- 13:15 - 14:45 Uhr, BW 74.4, BW 74.5
- 15:00 - 16:30 Uhr, SN 23.2 , BW 74.4
- 16:45 - 18:15 Uhr, RR 58.2, RR 58.1

- Donnerstag
- 13:15 - 14:45 Uhr, BW 74.4, BW 74.5
- 15:00 - 16:30 Uhr, BW 74.4, BW 74.5

- Freitag
- 09:45 - 11:15 Uhr, BW 74.4, BW 74.3
- **13:15 - 14:45 Uhr, IZ 161, SN 23.3**

Mailingliste

- Natürlich auf: www.ibr.cs.tu-bs.de/courses/ws0910/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
- 14-tägig auf www.ibr.cs.tu-bs.de/courses/ws0910/aud/index.html
- Bearbeitungszeit: 14 Tage
- Abgabe: Mittwochs bis 11:25 vor dem IZ 262

Hausaufgaben

Hausaufgaben

- 5 Hausaufgabenblätter
- 14-tägig auf www.ibr.cs.tu-bs.de/courses/ws0910/aud/index.html
- Bearbeitungszeit: 14 Tage
- Abgabe: Mittwochs bis 11:25 vor dem im IZ 262
- 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!

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)
44	1, 2	1		UE_0 *		
45	3, 4			HA_1		
46	5, 6	2	1			
47	7, 8			HA_2	HA_1	
48	9, 10	3				
49	11, 12		2	HA_3	HA_2	HA_1
50	13, 14	4				
51	15, 16		3	HA_4	HA_3	HA_2
52	Weihnachtsferien					
53						
1	17,18	5				
2	19, 20		4	HA_5	HA_4	HA_3
3	21, 22	6				
4	23, 24		5		HA_5	HA_4
5	25, 26	7	6			HA_5

*Bearbeitung in der ersten kleinen Übung; keine Bewertung

Das Rundreise-Problem

Fortsetzung der Rundreise

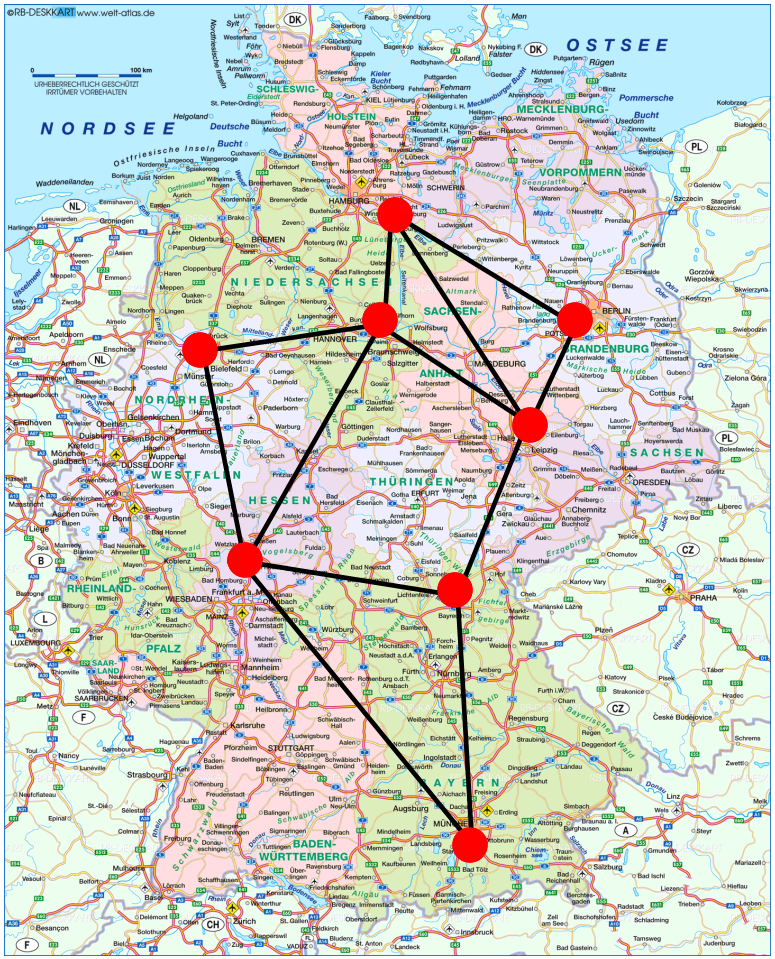
- Problem: Finde eine Rundreise, (d.h. eine Route die jeden Ort besucht und zum Startort zurückkehrt) die möglichst kurz ist.

Fortsetzung der Rundreise

Fortsetzung der Rundreise

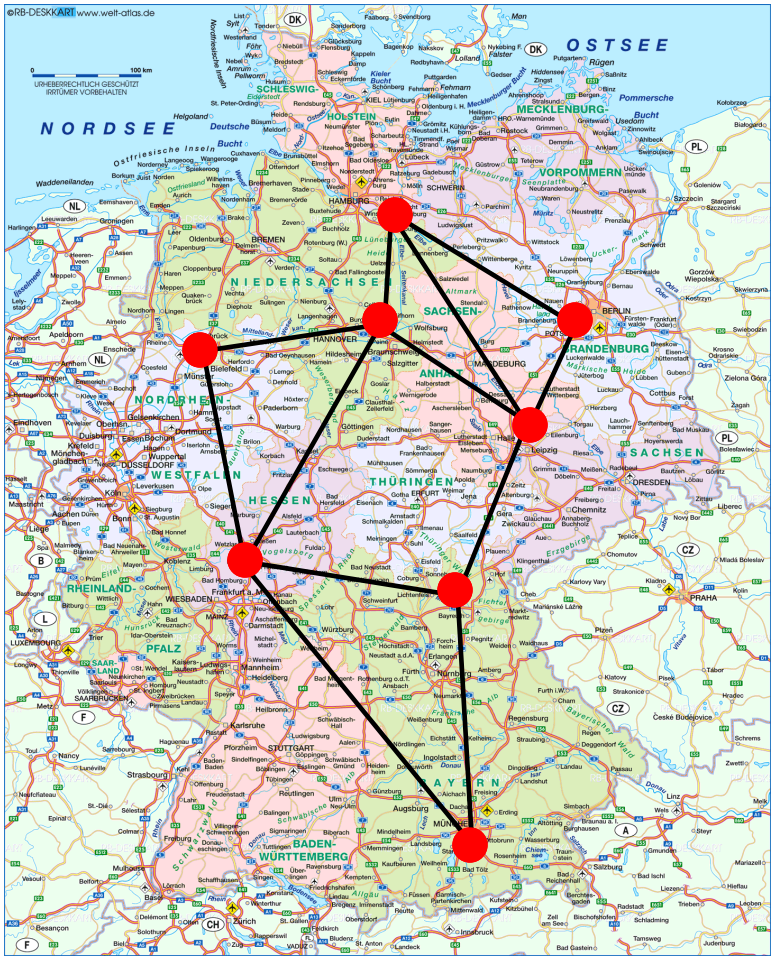
- Wie funktioniert map24.de?
- Man kann zwischen den Städten beliebig hin und her reisen.

Phase 1: Modellierung



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

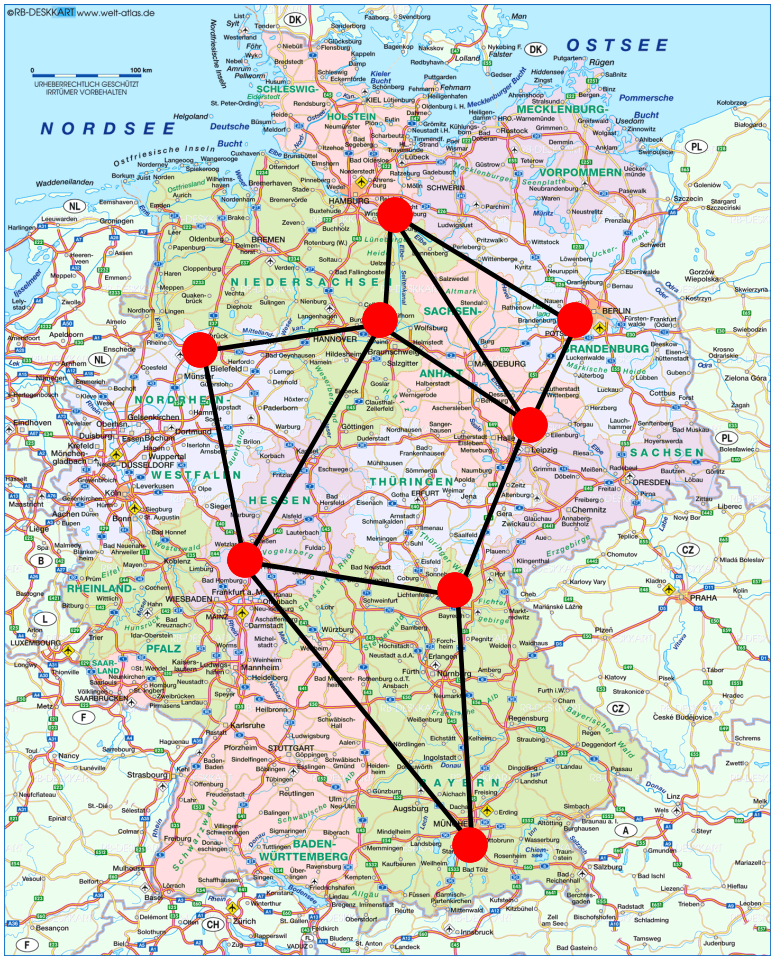
Phase 1: Modellierung



- 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:
  (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 „zi[illegible]ute“ finden
  [illegible]ithmus:
  - Wähle bel. Startort
  - Gehe zum nächstgelegenen Ort außer, dieser ist bereits besucht

Bald in dieser Vorlesung

Mehrheitsbestimmung

Szenario: Wahl

- N Wähler schreiben den Namen des favorisierten Kandidaten auf jeweils einen Zettel
- Gewinner ist, wer **mehr als die Hälfte** aller Stimmen hat, also mehr als N/2

Aufgabe: Auszählen der Stimmen

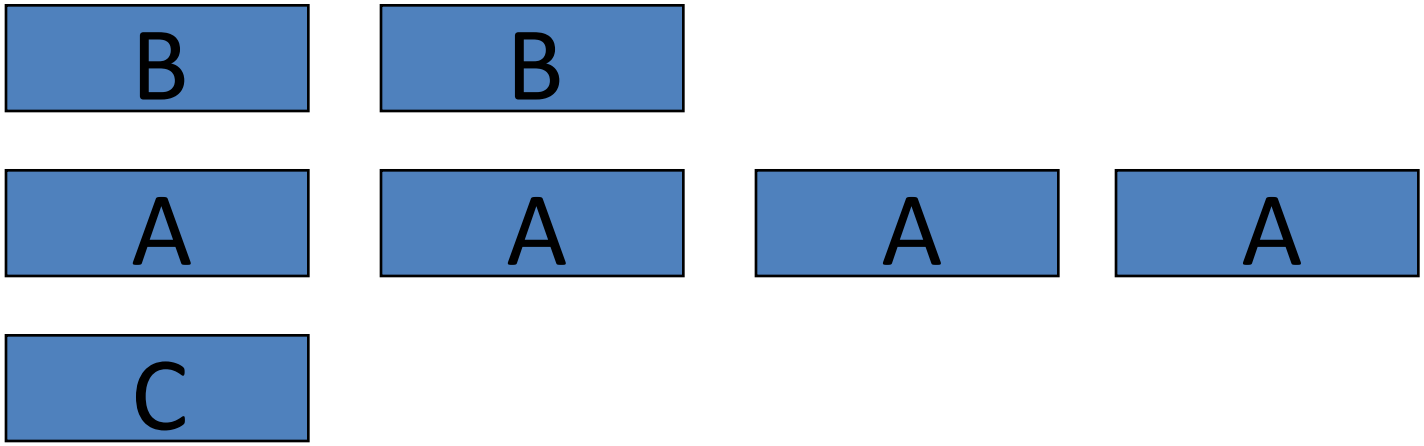
- Hat jemand mehr als die Hälfte aller Stimmen bekommen?
- Vorgehensweise (Algorithmus)?
- Ziel: Korrektes Ergebnis + Aufwand minimieren!

Erster Ansatz: Stimmen aller Kandidaten zählen

- Führe eine Liste mit Kandidaten
- Prüfe für jeden Wahlzettel
  - Sei A der Name auf Wahlzettel
  - Falls A in der Liste -> erhöhe Stimmenanzahl um 1
  - Falls A nicht in der Liste -> füge A hinzu
- Prüfe ob ein Kandidat mehr als N/2 Stimmen hat

Beispiel:

B, B, A, A, C, A, A



Kritik

- Der Ansatz funktioniert
- Aufwand?
  - Für jeden Wahlzettel muss man einmal die komplette Liste anschauen, d.h. ca. N/2 Vergleiche pro Wahlzettel (zu Beginn weniger, zum Schluss mehr)
  - Also insgesamt für N Wahlzettel ca. $N^2/2$ Vergleiche

Geht das effizienter?

Anderer Ansatz

- Verwende einen Stapel, anfangs leer.
- Phase 1: Für jeden Wahlzettel
  - Sei X der Name auf dem Wahlzettel und Y der oberste Name auf dem Stapel
  - Falls (Stapel leer oder X gleich Y)
    -> lege Wahlzettel auf den Stapel
  - Sonst
    -> Entferne X und Y

Anderer Ansatz (Fortsetzung)

- Phase 2:
- Falls (der Stapel leer ist)
  -> Ausgabe: Niemand hat die Mehrheit
- Sonst
  - Sei T der Name der am Ende oben auf dem Stapel liegt
  - Zähle, wie oft T unter allen Wahlzetteln vorkommt
  - Kommt T mehr als N/2 mal vor, so hat T die absolute Mehrheit; andernfalls hat niemand die absolute Mehrheit erreicht

Beispiel:

B, B, A, A, C, A, A

Am Ende von Phase 1 liegt ein A oben auf dem Stapel.

In Phase 2 zählt der Algorithmus, wie oft A insgesamt vorkommt, und erkennt A somit richtig als Mehrheitselement.

- Verwende einen Stapel, anfangs leer.
- Phase 1: Für jeden Wahlzettel
  - Sei X der Name auf dem Wahlzettel und Y der oberste Name auf dem Stapel
  - Falls (Stapel leer oder X gleich Y)
    -> lege Wahlzettel auf den Stapel
  - Sonst
    -> Entferne X und Y

Stapel

Wieso funktioniert dieser Algorithmus?

- Phase 2:
- Falls (der Stapel leer ist)
  -> Ausgabe: Niemand hat die Mehrheit
- Sonst
  - Sei T der Name der am Ende oben auf dem Stapel liegt
  - Zähle, wie oft T unter allen Wahlzetteln vorkommt
  - Kommt T mehr als N/2 mal vor, so hat T die absolute Mehrheit; andernfalls hat niemand die absolute Mehrheit erreicht

Wieso funktioniert dieser Algorithmus?

- Wir beobachten: Der Algorithmus könnte nur dann einen Fehler machen, wenn die Eingabe ein Mehrheitselement X enthält, am Ende von Phase 1 aber der Stapel leer ist oder ein anderes Element als X oben am Stapel liegt.

Kann dieser Fall jemals eintreten?

Wieso funktioniert dieser Algorithmus?

- Beweis durch Widerspruch: Angenommen, am Ende von Phase 1 ist der Stapel leer oder ein anderes Element als X liegt oben auf dem Stapel, es gibt aber ein Mehrheitselement X

Beweis

- Was ist mit jedem X in Phase 1 passiert?
  1. Als X bearbeitet wurde, lag gerade ein anderes Element Y oben am Stapel. X wurde nicht auf den Stapel gelegt und Y vom Stapel entfernt.
  2. Als X bearbeitet wurde, wurde es auf den Stapel gelegt, aber später aufgrund eines anderen Elementes Z wieder vom Stapel entfernt.

Abschluss des Beweises

- Jedem X lässt sich somit eindeutig ein Element Y oder Z zuordnen
- Es gibt daher mindestens so viele Elemente ungleich X, wie es Elemente X gibt
- Das kann nicht sein, wenn X wirklich das Mehrheitselement ist

Der Algorithmus ist also immer korrekt!

Ist Phase 2 überhaupt nötig?

- Ja, denn z.B. bei
  A,B,A,C,C
  liegt nach Phase 1 ein C oben
- Phase 2 erkennt, dass C keine Mehrheit hat

Aufwand?

- Wieder N Wähler
  - In Phase 1: N Vergleiche
    - Aktueller Name mit Name auf Stapel
  - In Phase 2: höchstens N Vergleiche
  - Insgesamt: höchstens 2N Vergleiche
  - Vorher: $N^2/2$

Was haben wir gesehen?

- Es lohnt sich über Algorithmen nachzudenken (von $N^2/2$ auf 2N)
- Pseudo-Code:

  - Verwende einen Stapel, anfangs leer.
  - Phase 1: Für jeden Wahlzettel
    - Sei X der Name auf dem Wahlzettel und Y der oberste Name auf dem Stapel
    - Falls (Stapel leer oder X gleich Y)
      -> lege Wahlzettel auf den Stapel.
    - Sonst
      -> Entferne X und Y

Was haben wir gesehen?

- Es lohnt sich über Algorithmen nachzudenken (von 2N auf N^2)
- Pseudo-Code:

```
:
:
Verwende einen Stapel, anfangs leer.
Phase 1: Für jeden Wahlzettel
    - Sei X der Name auf dem Wahlzettel und Y der
                oberste Name auf dem Stapel
    - Falls (Stapel leer oder X=Y) -> lege Wahlzettel auf
                den      Stapel.
    - Sonst
        -> Entferne X und Y
```

- Analyse von zwei Algorithmen

Schönes Wochenende….

… nächste Übung in 14 Tagen