

# Kap. 6: Graphen

Professor Dr. Petra Mutzel  
 Lehrstuhl für Algorithm Engineering, LS11

17. VO 6. Juni 2006

## Motivation

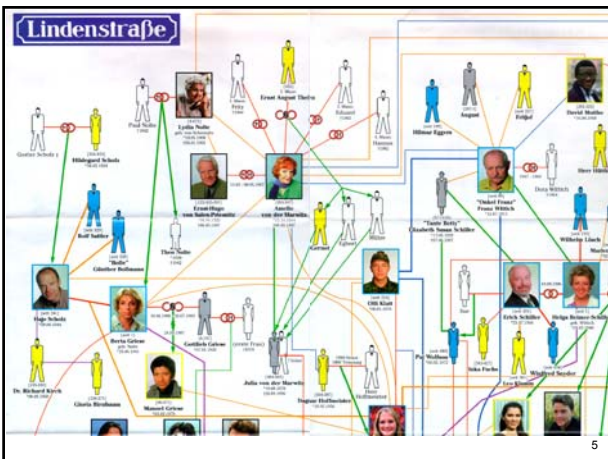
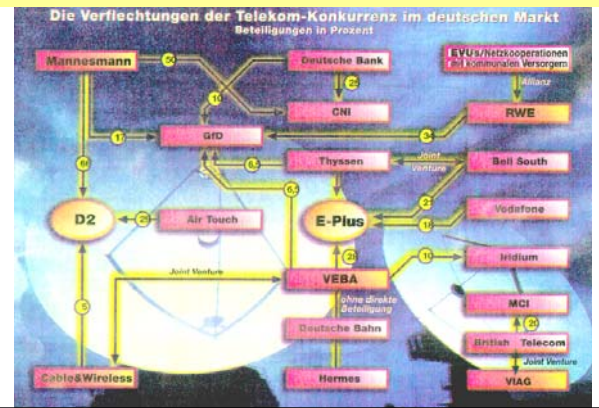
„Warum soll ich heute hier bleiben?“  
 Graphen sind wichtig und machen Spaß!

„Was gibt es heute Besonderes?“  
 schöne Bilder

## Überblick

- Motivation
- Einführung von Graphen
- Datenstrukturen
- Traversieren von Graphen:
  - Breitensuche (BFS)
  - Tiefensuche (DFS) [nächstes Mal]

Graph: Objekte und Beziehungen zwischen den Objekten



**US-Präsident Bush gelingt Schlag gegen Bin Ladens Finanznetz**

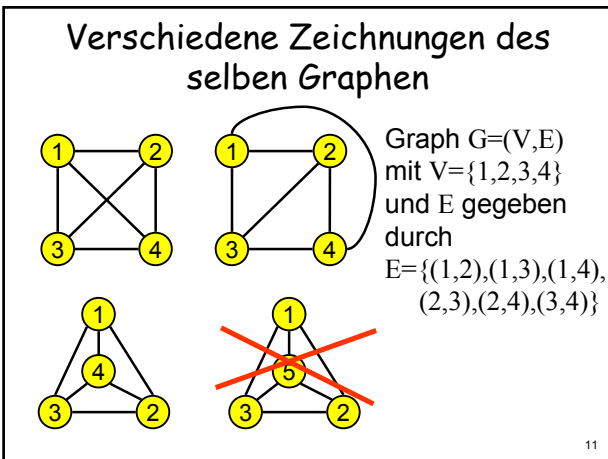
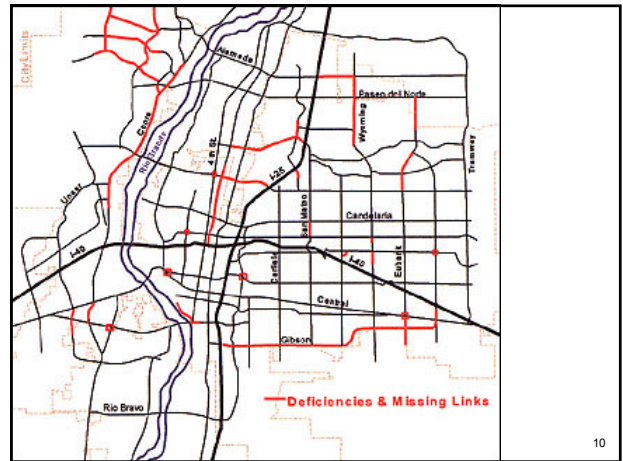
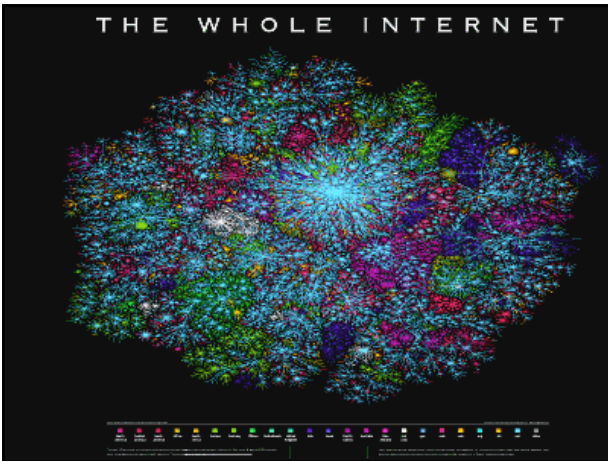
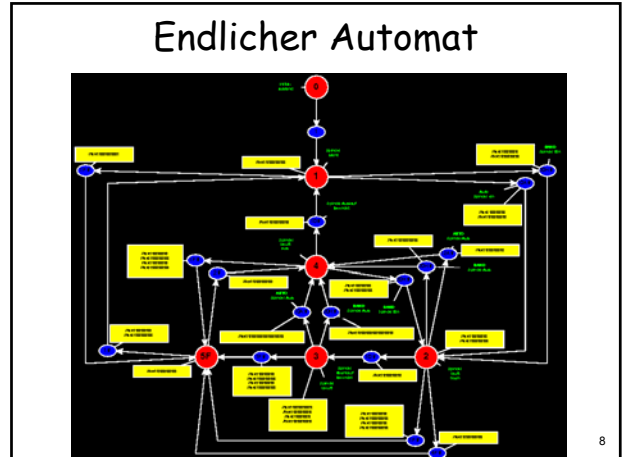
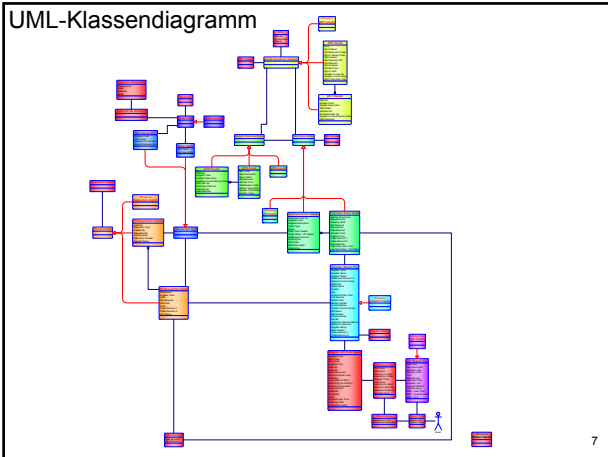
In der Schweiz und in Somalia wird gegen Verdächtige ermittelt

Washington/Wien. Weltweit wurden am Donnerstag die Verhaftungsversuche von 62 Einzelpersonen und Gruppen mit vermuteten Verbindungen zu Bin Laden eingeleitet. Dabei handelt es sich um 43 Männer und 19 Frauen.

160 deutsche Konten gesperrt

In Deutschland sind derzeit noch 160 Konten von Verdächtigen gesperrt.

**Kölner Stadtanzeiger**  
 09.11.2001



- ### Motivation
- Graphen modellieren diskrete Strukturen
  - hilfreich zur Analyse und Optimierung
- Straßen-, Bahnnetze: kürzeste Wege
  - Modellierung von Prozessen, z.B. Geschäftsprozesse, Betriebsabläufe
  - Proteininteraktionsnetzwerke in der molekularen Biologie
- 12

## Kap. 6.1 Definition (Graph)

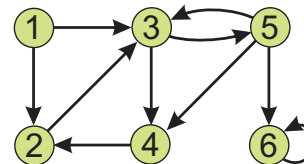
Graph  $G=(V,E)$  besteht aus

- einer Menge  $V$  von Knoten
- einer (Multi-)menge  $E$  von Kanten, die Paaren von Knoten entsprechen.
- Bei Multimenge kann ein Paar  $(v,w)$  mehrfach in  $E$  vorkommen  $\rightarrow$  Mehrfachkanten
- Eine Kante  $(v,v)$  heißt Schleife (self-loop)
- Annahmen:
  - $V$  und  $E$  sind endliche Mengen
  - Mehrfachkanten erlaubt

13

## Gerichtete Graphen

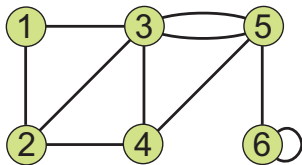
- Sind die Paare in  $E$  geordnet:  $E \subseteq V \times V \rightarrow$  **gerichteter Graph** (Digraph)
- Kanten heißen dann: **gerichtete Kanten** (Bögen, directed edges, arcs)
- Maximale Kantenanzahl eines Digraphen ohne Schleifen und Mehrfachkanten:  $|E| \leq |V|(|V|-1)$



14

## Ungerichtete Graphen

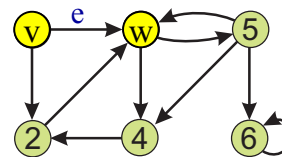
- Sind die Paare in  $E$  ungeordnet  $\rightarrow$  (**ungerichteter Graph**)
- Kanten heißen dann: **Kanten** (edges)
- Maximale Kantenanzahl ohne Schleifen und Mehrfachkanten:  $|E| \leq \frac{1}{2} |V|(|V|-1)$



15

## Definitionen (Nachbarn)

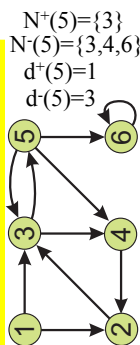
- Sei  $e=(v,w)$  eine Kante in  $E$ , dann sagen wir:
  - $v$  und  $w$  sind **adjazent**
  - $v$  (bzw.  $w$ ) und  $e$  sind **inzident**
  - $v$  und  $w$  sind **Endpunkte** von  $e$
  - $v$  und  $w$  sind **Nachbarn**
  - $e$  ist eine **ausgehende** Kante von  $v$  und eine **eingehende** Kante von  $w$  (falls  $G$  Digraph)



16

## Definitionen für gerichtete Graphen $G=(V,A)$

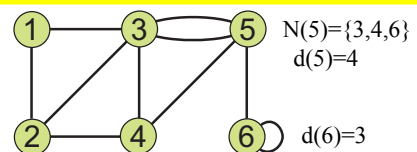
- Eingehende Nachbarmenge von  $v \in V$ :  $N^+(v) := \{u \in V \mid (u,v) \in A\}$
- Ausgehende Nachbarmenge von  $v \in V$ :  $N^-(v) := \{w \in V \mid (v,w) \in A\}$
- $A^+(v) :=$  Menge der eingehenden Kanten von  $v$
- $A^-(v) :=$  Menge der ausgehenden Kanten von  $v$
- $A(v) := A^+(v) \cup A^-(v)$
- Eingangsgrad  $d^+(v) := |A^+(v)|$
- Ausgangsgrad  $d^-(v) := |A^-(v)|$
- **Knotengrad**  $d(v) := d^+(v) + d^-(v)$



17

## Definitionen für ungerichtete Graphen $G=(V,E)$

- **Nachbarmenge** von  $v \in V$ :  $N(v) := \{w \in V \mid (v,w) \in E\}$
- Menge der zu  $v$  inzidenten Kanten  $E(v) := \{(u,v) \mid (u,v) \in E\}$
- **Knotengrad**  $d(v)$  ist die Anzahl der zu  $v$  inzidenten Kanten, wobei eine Schleife 2 Mal gezählt wird



18

## Lemma (gerade Knotengrade)

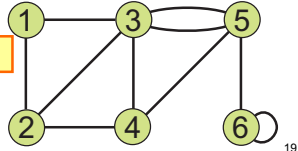
- In einem ungerichteten Graphen  $G=(V,E)$  ist die Anzahl der Knoten mit ungeradem Knotengrad gerade.
- Summiert man über alle Knotengrade, so zählt man jede Kante genau zweimal:  

$$\sum_{v \in V} d(v) = 2 |E|$$

L.S.: gerade

R.S.: gerade

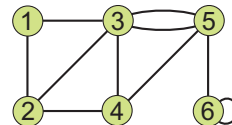
also auch die Anzahl der ungeraden Summanden



19

## Definitionen (Wege)

- Sei  $G=(V,E)$  gerichtet oder ungerichtet:
- Ein **Kantenzug** (walk) **der Länge k** ist eine nicht-leere Folge  $v_0, e_1, v_1, e_2, \dots, e_k, v_k$  von abwechselnd Knoten und Kanten aus  $G$  mit  $e_i=(v_{i-1}, v_i)$  für  $i=1, \dots, k$ .
- Man schreibt auch:  $v_0, v_1, \dots, v_k$
- Ein **Weg** (path) ist ein Kantenzug in dem alle Knoten verschieden sind.



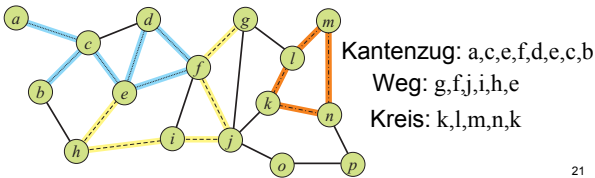
Kantenzug: 1,3,2,4,3,5

Weg: 1,3,2,4,5

20

## Definitionen (Kreis)

- Sei  $G=(V,E)$  gerichtet oder ungerichtet:
- Ist  $v_0, e_1, v_1, e_2, \dots, e_{k-1}, v_{k-1}$  ein Weg mit  $k \geq 3$  und  $e_k=(v_{k-1}, v_0)$  eine Kante aus  $G$ , dann ist  $v_0, e_1, v_1, e_2, \dots, e_{k-1}, v_{k-1}, e_k, v_0$  ein **Kreis der Länge k** in  $G$ .



Kantenzug: a, c, e, f, d, e, c, b

Weg: g, f, j, i, h, e

Kreis: k, l, m, n, k

21

## Darstellung von Graphen im Rechner: Statische Graphen

- Im Folgenden sei  $V=\{v_1, v_2, \dots, v_n\}$

- 1. Möglichkeit: Adjazenzlisten

- Idee:** Speichere für jeden Knoten seine Nachbarmenge in einer Liste

- Realisierung:** z.B. Knoten in Array und Nachbarkanten jedes Knotens als einfach verkettete Liste

22

## Darstellung von Graphen im Rechner: Statische Graphen

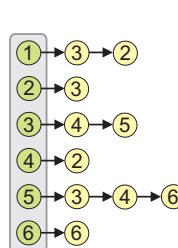
- 2. Möglichkeit: Adjazenzmatrix

- Idee:** Eine  $V \times V$  Matrix enthält 0/1-Einträge für jedes Knotenpaar  $\{u, v\}$

- Realisierung:**
- Sei  $M=(m_{ij})$  eine  $n \times n$  Matrix mit  $m_{ij}=1$  falls  $(v_i, v_j) \in E$ , und  $m_{ij}=0$  sonst.
- bei Mehrfachkanten schreibe statt 1 die Anzahl der Kanten

23

## Darstellung gerichteter Graphen



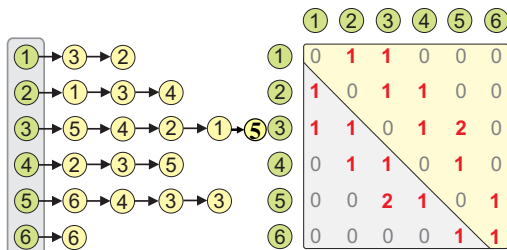
Adjazenzlisten

	1	2	3	4	5	6
1	0	1	1	0	0	0
2	0	0	1	0	0	0
3	0	0	0	1	1	0
4	0	1	0	0	0	0
5	0	0	1	1	0	1
6	0	0	0	0	0	1

Adjazenzmatrix

24

## Darstellung ungerichteter Graphen



Adjazenzlisten

Adjazenzmatrix

ist symmetrisch: nur Speicherung der oberen Hälfte

25

## Diskussion

HIER: ab jetzt Adjazenzlisten

für dünne Graphen vorzuziehen!

### Adjazenzliste:

- Speicherplatzverbrauch: linear:  $\Theta(|V|+|E|)$
- Zeit für Aufbau: linear:  $\Theta(|V|+|E|)$
- Abfrage, ob Kante  $(u,v)$  existiert:  $\Theta(d(v))$
- Iteration über alle Nachbarn von  $v \in V$ :  $\Theta(d(v))$

### Adjazenzmatrix:

- Speicherverbrauch immer quadratisch:  $\Theta(|V|^2)$
- Zeit für Aufbau: immer quadratisch:  $\Theta(|V|^2)$
- Abfrage, ob Kante  $(u,v)$  existiert:  $\Theta(1)$
- Iteration über alle Nachbarn von  $v \in V$ :  $\Theta(|V|)$

26

## Diskussion

- Die **Dichte** (density) eines Graphen  $G$  ist das Verhältnis  $|E| / |V|$ .
- $G$  heißt **dünn**, falls seine Dichte  $O(1)$  ist
- $G$  heißt **dicht**, falls seine Dichte  $\Omega(|V|)$  ist.

27

## Darstellung von Graphen im Rechner: Dynamische Graphen

- Dynamisch unter den Operationen:
  - Hinzufügen neuer Knoten und Kanten
  - Entfernen von Knoten und Kanten

- **Idee:** für gerichtete Graphen:
- Inzidenzlisten: speichere ein- und ausgehende Knoten bei  $v$
- Knoten in doppelt verketteter Liste (damit Entfernen in konstanter Zeit)
- Inzidenzlisten in doppelt verketteten Listen

28

## Realisierung Dynamische Graphen: Liste für die Knoten

```

struct Node
  var Node prev // Vorgänger Knotenliste
  var Node next // Nachfolger in Knotenliste
  var Edge outHead // Listenanfang ausgeh. Kanten
  var Edge inHead // Listenanfang eingeh. Kanten
  var int index // fortlaufender Index
end struct
    
```

29

## Realisierung Dynamische Graphen: Liste für die Kanten

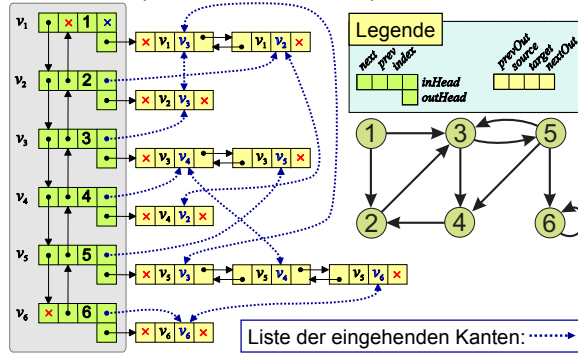
```

struct Edge
  var Edge prevOut //Vorgänger in Liste ausg. Kanten
  var Edge nextOut //Nachfolger in Liste ausg. Kanten
  var Edge prevIn // Vorgänger in Liste eing. Kanten
  var Edge nextIn // Nachfolger in Liste eing. Kanten
  var node source // Anfangsknoten der Kante
  var node target // Endknoten der Kante
end struct
    
```

Achtung: jeder Kanteneintrag ist genau einmal in Liste enthalten

30

## Darstellung von Graphen im Rechner: Dynamische Graphen



## Analyse Dynamischer Graphen

- Speicherplatzverbrauch: linear:  $\Theta(|V|+|A|)$
- Zeit für Aufbau: linear:  $\Theta(|V|+|E|)$
- Abfrage, ob Kante  $(u,v)$  existiert:  $\Theta(d(v))$
- Iteration über alle Nachbarn von  $v \in V$ :  $\Theta(d(v))$
- Iteration über alle ausg. Kanten von  $v \in V$ :  $\Theta(d^-(v))$
- Iteration über alle eing. Kanten von  $v \in V$ :  $\Theta(d^+(v))$
- Einfügen eines Knotens bzw. Kante:  $\Theta(1)$
- Entfernen einer Kante:  $\Theta(1)$
- Entfernen eines Knotens:  $\Theta(d(v))$

Man geht davon aus, dass man jeweils Zeiger auf die Knoten und beim Entfernen auch auf die Kanten gegeben hat 32

## Kap. 6.3 Traversieren von Graphen

Traversieren: systematisches Durchwandern von Graphen

HIER: ungerichtete Graphen

**Def.:** Der **graphentheoretische Abstand** zweier Knoten  $u, v$  eines ungerichteten Graphen  $G$  ist die Länge des kürzesten Weges von  $u$  nach  $v$ , falls ein solcher existiert, sonst  $\infty$ .

Achtung: hierbei werden keine vorgegebenen Kantenlängen bzw. Kantengewichte berücksichtigt. 33

## Breitensuche (BFS)

Die Folien zu diesem Teil finden Sie bei der folgenden Vorlesung. Dort wurde BFS wiederholt, diesmal mit Animationen.

34