

Algoritmen & Datastructuren

Herfst 2018

Vorlesing 2

# Graphen & Theorie

Funktionsgraph:  
(aus Schule)

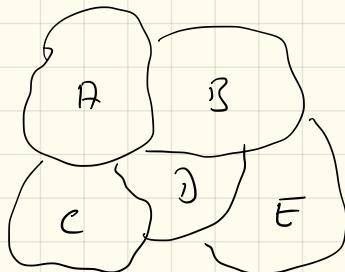


hat nichts mit  
Graphentheorie zu tun

## Beispiel: Kartenfärbung

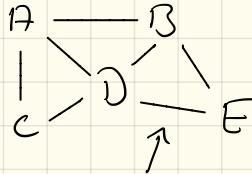
Länder

- färbt jedes Land
- benachbarte Länder  
haben unterschiedliche  
Farben



Wie viele Farben braucht man?

Abstraktion:

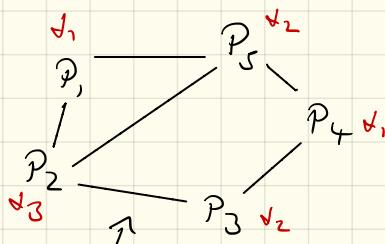


benachbart

## Beispiel: Prüfungszeitplan

- n Prüfungen
- k Termine
- verschiedene Termine wenn  $\geq 1$  Student  
scheitert

Abstraktion:



Termine:  
 $d_1, d_2, d_3$

Konflikt besteht  
(1 Student schreibt beide)  
⇒ verschiedene Turniere

Ist das gleiche Problem nur zuvor!

Die Abstraktion macht das leichter

II

## Graphentheorie

Definition Graph: ein graph  $G = (V, E)$   
bestehend aus

Knoten  $V = \{v_1, \dots, v_n\}$  vertices

Kanten  $E = \{e_1, \dots, e_m\}$  edges

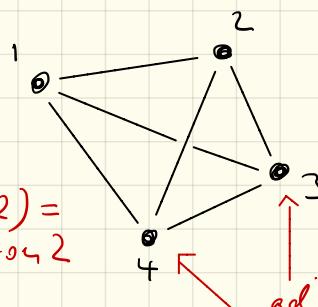
jede Kante ist ein ungeordnetes Paar

$$e_k = \{v_i, v_j\}$$

Nachbarn von 2:

$$\{1, 4, 3\}$$

Degrad von 2 = deg(2) =  
Anzahl Nachbarn von 2  
= 3



4 Knoten

6 Kanten

$\{1,2,3, \{2,3\}, \dots\}$ , etc.

deg = degree

Schlußfolgerung:  $\deg(v_1) + \dots + \deg(v_n) = 2m$

Vorlesung Beispiel  $3+3+3+3 = 12 \quad \checkmark$

Beweis: Folgen

Graphenfärbung:

Eingabe:  $G = (V, E)$ ,  $k$  Farben

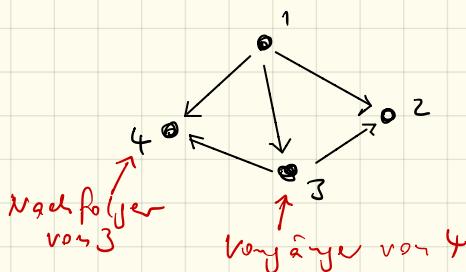
Ziel: färbe Knoten so daß Nachbarn unterschiedliche Farben haben (wenn es geht)

im allgemeinen Sinn

Definition gewichteter Graphs:  $G = (V, E)$

wobei zuver, nur sind Knoten gewichtet

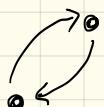
Paare:  $e_k = (v_i, v_j)$



Knoten:  $(1,4), (1,2), (1,3)$   
 $(3,4), (3,2)$

Fuß Kopf

1 hat keine Vorgänger  
4 hat keine Nachfolger



ist erlaubt



"Schleife"  
ist erlaubt, wenn häufig ausgeschlossen

## Beispiele geschichteter Graphs:

- WWW:

Knoten: Webseiten

Kanten: Hyperlinks

- Straßennetz

- Soziale Netzwerke

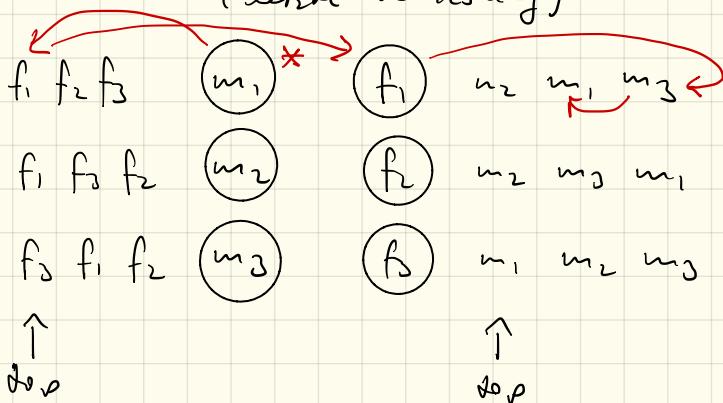
Knoten: Nutzer

Kanten: "is 2 Follower von"

---

Algorithmen abstrahieren Computerprogramme  
Datenstrukturen abstrahieren den Computer speziell

Beispiel: Gale-Shapley Algorithmus  
(lebende Verteilung)



Anfang  
Befürdeung

Zugriff als folge  
völlig  
Algorithmus  
ist ein  
Graph

# < Schmausfall 1 Pad >

Wege, Pfeile, Zahlen, Werte, Distanz

## Einschub: Induktion

Ziel: zeige eine Aussage  $A(n)$  für alle  $n \geq 1$ .

Beispiele für  $A(n)$ : a.)  $1 + 2 + \dots + n = \frac{1}{2}n(n+1)$

b.) wenn es einen Weg von  $u \rightarrow v$  der Länge  $n$  gibt, dann gibt es auch einen Pfeil der Länge  $\leq n$

Induktion (allgemeines Prinzip):

Induktionsanfang  $(n=1)$  1.) zeige  $A(1)$  Induktionshypothese

Induktionsabschluss  $(n-1 \rightarrow n)$  2.) zeige  $A(n-1) \Rightarrow A(n)$   
(aus  $A(n-1)$  folgt  $A(n)$ )

Beispiel Beweis a.)

$$1 = \frac{1}{2} \cdot 1 \cdot (1+1) \quad \checkmark$$

$$2.) 1 + \dots + n-1 = \frac{1}{2}(n-1)n \quad (\text{Ind. Hypothese})$$

$$\Rightarrow 1 + \dots + n-1 + n = \frac{1}{2}(n-1)n + n = \frac{1}{2}n(n+1) \quad \checkmark$$

Beweis S.) =

Sei Weg der Länge  $n$ :  $u \xrightarrow{\omega_0} \omega_1, \dots, \omega_{n-1}, v \xrightarrow{\omega_n}$

1.)  $n=1$ :  $u \rightarrow v$  ist Pfeil ✓

2.)  $n-1 \rightarrow n$ :

Fall 1: der Weg ist ein Pfeil  $\rightarrow$  fertig

Fall 2: der Weg ist kein Pfeil

$\Rightarrow \omega_i = \omega_j$  für  $0 \leq i < j \leq n$

also  $u \xrightarrow{\omega_0} \omega_1 \xrightarrow{\omega_1} \dots \xrightarrow{\omega_i} \begin{matrix} \xrightarrow{\omega_i} \\ (= \omega_j) \end{matrix} \xrightarrow{\omega_j} \dots \xrightarrow{\omega_{n-1}} v$

Kürze Weg als:  $u, \omega_1, \dots, \omega_i, \omega_{i+1}, \dots, v$

- ist immer noch Weg

- hat Länge  $\leq n-1$

$\Rightarrow$  nach Ind. Hypothese gibt

es einen Pfeil  $u \rightarrow v$

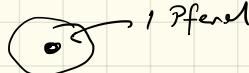
der Länge  $\leq n-1 \leq n$

✓

## Trübles bei der Induktion

Beispiel: In einer Gruppe von  $n$  Pferden haben alle die gleiche Farbe

Anfang ( $n=1$ ):

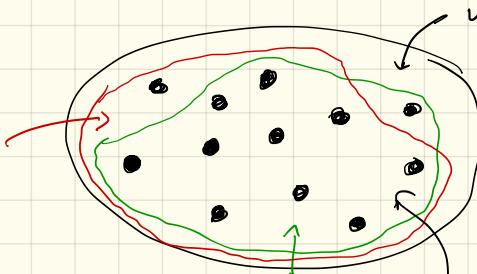


1 Pferd



Schritt ( $n-1 \rightarrow n$ ):

$n-1$  Pferde



$n-1$  Pferde

Da Überlappung  
ist die Farbe  
gleich  
 $\Rightarrow$  Behauptung

Induktionshypothese:  
alle haben gleiche Farbe

Wo ist das Problem?