15. Januar 2019

# Graphen

# Graphen

**Definition**

Ein *(ungerichteter, schlichter) Graph* ist ein Paar $G = (V, W)$ mit

- $V$ eine endliche Menge;
- $E$ Menge von zweielementigen Teilmengen von $V$.

**Sprechweisen**

Ist $G = (V, E)$ eine Graph, dann heißen

- die Elemente von $V$ *Knoten* von $G$ (English: *vertex*),
- die Elemente von $E$ *Kanten* von $G$ (English: *edge*),
- $n_G := |V|$ die *Knotenzahl* von $G$,
- $m_G := |E|$ die *Kantenzahl* von $G$.

Für $\{u, v\} \in E$ schreiben wir auch $uv$ oder $vu$.

# Graphen (Forts.)

## Bemerkungen

- Mathematisches Modell für Kante zwischen $u, v \in V$:
  *zwei*elementige Teilmenge $\{u, v\} = \{v, u\} \subseteq V$.
- Andere verbreitete Definitionen von Graphen erlauben
  - gerichtete Kanten,
  - Schlingen,
  - Mehrfachkanten,
  - gewichtete Kanten,
  - gefärbte Kanten,
  - unendlich viele Knoten oder Kanten.
  - usw.

  Mathematisches Modell für Kanten wird angepasst:
  Z.B.: gerichtete Kante vom Knoten $u$ zum Knoten $v$
  modelliert durch $(u, v) \in V \times V$.

# Graphen (Forts.)

## Motivation

Graphen modellieren Netzwerke, z.B.

- Straßennetze
  - Knoten: Kreuzungen
  - Kanten: Straßen
- Stromnetze
  - Knoten: Umspannstationen
  - Kanten: Stromleitungen
- Computernetze
- Workflow-Diagramme

# Graphen (Forts.)

## Zeichnungen

Oft werden Graphen durch Bilder dargestellt. Beispiel:
$V = \{1, 2, 3, 4\}$,
$E = \{\{1, 4\}, \{1, 2\}, \{1, 3\}, \{2, 4\}, \{2, 3\}\}$.

# Graphen (Forts.)

Es sei $G = (V, E)$ ein Graph.

## Begriffe

- Es seien $u, v \in V$ mit $u \neq v$ und es sei $uv \in E$.
  - $u$ und $v$ heißen die *Endknoten* von $uv$.
  - $u$ und $v$ heißen *adjazent*.
  - $u$ heißt *Nachbar* von $v$ und umgekehrt.
- Für $v \in V$ ist $\Gamma(v) := \Gamma_G(v)$ die Menge der Nachbarn von $v$.
- $e \in E$ *inzident* zu $v \in V$, wenn $v$ ein Endknoten von $e$ ist.
- Zwei verschiedene Kanten heißen *inzident*, wenn sie einen gemeinsamen Endknoten haben.
- $G$ heißt *vollständiger Graph*, falls je zwei verschiedene Knoten von $G$ adjazent sind.

# Die Adjazenzmatrix

Es sei $G = (V, E)$ ein Graph mit $V = \{1, \dots, n\}$.

**Definition**

Die *Adjazenzmatrix* von $G$ ist die Matrix

$$A := \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ \vdots & & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} \in \{0,1\}^{n \times n}$$

mit

$$a_{ij} := \begin{cases} 1 & \text{falls } ij \in E, \\ 0 & \text{falls } ij \notin E. \end{cases}$$

Die *Adjazenzliste* von $G$ ist die Liste

$$\Gamma := (\Gamma(1), \Gamma(2), \dots, \Gamma(n)).$$

# Die Adjazenzmatrix (Forts.)

**Beispiel**

$V = \{1, 2, 3, 4\}$,
$E = \{\{1, 4\}, \{1, 2\}, \{1, 3\}, \{2, 4\}, \{2, 3\}\}$.

# Die Inzidenzmatrix

Es sei $G = (V, E)$ ein Graph mit $V = \{1, \dots, n\}$ und $E = \{e_1, \dots, e_m\}$.

**Definition**

Die *Inzidenzmatrix* von $G$ ist die Matrix

$$B := \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1m} \\ \vdots & & & \vdots \\ b_{n1} & b_{n2} & \cdots & b_{nm} \end{pmatrix} \in \{0,1\}^{n \times m}$$

mit

$$b_{ij} := \begin{cases} 1 & \text{falls } i \in e_j, \\ 0 & \text{falls } i \notin e_j. \end{cases}$$

Die $j$-te Spalte der Inzidenzmatrix enthält genau zwei Einsen, nämlich zu den beiden Endknoten der Kante $e_j$.

# Die Inzidenzmatrix (Forts.)

**Beispiel**

$V = \{1, 2, 3, 4\}$,
$E = \{\{1, 4\}, \{1, 2\}, \{1, 3\}, \{2, 4\}, \{2, 3\}\}$.

# Grad

Es sei $G = (V, E)$ ein Graph.

**Definition**

- Für $v \in V$ heißt $\deg(v) := |\Gamma(v)|$ der *Grad von* $v$.
- Knoten vom Grad 0 heißen *isoliert*.

**Bemerkung**

Es gilt

$$\sum_{v \in V} \deg(v) = 2m_G.$$

**Folgerung**

Die Anzahl der Knoten von $G$ mit ungeradem Grad ist gerade.

# Teilgraphen

Es sei $G = (V, E)$ ein Graph.

**Definition**
Ein Graph $G' = (V', E')$ heißt *Teilgraph* von $G$,
geschrieben $G' \leq G$,
wenn $V' \subseteq V$ und $E' \subseteq E$ ist.

**Beispiel**
Ist $V' \subseteq V$, so wird durch

$$E' := \{uv \in E \mid u, v \in V'\}$$

ein Teilgraph $(V', E')$ von $G$ definiert,
der *auf $V'$ induzierte Teilgraph von $G$*,
geschrieben $G|_{V'}$.

# Teilgraphen (Forts.)

**Beispiele**

# Kantenzüge, Kreise und Pfade

Es sei $G = (V, E)$ ein Graph und $l \in \mathbb{N}_0$.

**Definition**

- Ein *Kantenzug der Länge $l$ in $G$* ist ein Tupel $(v_0, v_1, \ldots, v_l)$ von Knoten mit $v_i v_{i+1} \in E$ für alle $i = 0, \ldots, l-1$ (heißt auch $v_0$-$v_l$-Kantenzug).
- Der Kantenzug heißt *geschlossen* falls $v_0 = v_l$ ist.
- Ein Kantenzug $(v_0, \ldots, v_l)$ heißt *Pfad der Länge $l$ in $G$*, falls die Knoten $v_0, \ldots, v_l$ paarweise verschieden sind. (heißt auch $v_0$-$v_l$-Pfad).
- Ein *Kreis der Länge $l$ in $G$* ist ein geschlossener Kantenzug $(v_0, \ldots, v_l)$, für den $l \geq 3$ und $(v_0, \ldots, v_{l-1})$ ein Pfad ist.
- Eine *Tour der Länge $l$ in $G$* ist ein geschlossener Kantenzug $(v_0, \ldots, v_l)$, für den die Kanten $v_0 v_1, v_1 v_2, \ldots, v_{l-1} v_l$ paarweise verschieden sind.

# Kantenzüge, Kreise und Pfade (Forts.)

**Beispiele**

# Zusammenhang

Es sei $G = (V, E)$ ein Graph.

**Definition**

- Die *Zusammenhangsrelation* $\sim$ auf $V$ wird definiert durch

$$u \sim v :\Leftrightarrow \text{ es gibt einen } u\text{-}v\text{-Kantenzug in } G.$$

- $G$ heißt *zusammenhängend*, falls $u \sim v$ für alle $u, v \in V$, anderenfalls *unzusammenhängend*.

- *Zusammenhangskomponenten von $G$*; die induzierten Teilgraphen $G|_U$, wobei $U$ die Äquivalenzklassen von $V$ bzgl. $\sim$ durchläuft.

- $r_G$: Anzahl der Zusammenhangskomponenten von $G$

# Zusammenhang (Forts.)

**Beispiele**

# Zusammenhang (Forts.)

Es sei $G = (V, E)$ ein Graph.

**Lemma**

Für alle $u \neq v \in V$ gilt:

- $r_G - 1 \leq r_{(V, E \cup \{uv\})} \leq r_G$.
- $r_{(V, E \setminus \{uv\})} - 1 \leq r_G \leq r_{(V, E \setminus \{uv\})}$.

**Satz**

- Untere Schranke für $m_G$: $m_G \geq n_G - r_G$.
- Obere Schranke für $m_G$: $m_G \leq \binom{n_G + 1 - r_G}{2}$.

# Zusammenhang (Forts.)

Es sei $G = (V, E)$ ein Graph.

**Folgerung**

- Ist $G$ zusammenhängend, dann ist $m_G \geq n_G - 1$.
- Ist $G$ unzusammenhängend, so gilt $m_G \leq \binom{n_G - 1}{2}$.

# Brücken

Es sei $G = (V, E)$ ein Graph, $u, v \in V$ mit $e = uv \in E$.

**Bemerkung**

Es sei $G' := (V, E \setminus \{e\})$. Dann sind äquivalent:

- $u \not\sim v$ in $G'$.
- $r_{G'} > r_G$.

**Definition**

$e$ heißt *Brücke von $G$*, wenn eine der beiden Bedingungen aus der Bemerkung erfüllt ist.

# Brücken (Forts.)

Es sei $G = (V, E)$ ein Graph, $u, v \in V$ mit $e = uv \in E$.

**Bemerkung**

Es sei $G' := (V, E \setminus \{e\})$.
Dann sind die folgenden Aussagen äquivalent.

- $e$ ist keine Brücke von $G$.
- $u \sim v$ in $G'$.
- $r_{G'} = r_G$.
- es gibt einen $u$-$v$-Kantenzug in $G$, der nicht über $e$ führt.
- es gibt einen $u$-$v$-Pfad in $G$, der nicht über $e$ führt.
- $e$ ist Teil eines Kreises in $G$.

# Brücken (Forts.)

Es sei $G = (V, E)$ ein Graph und $l \in \mathbb{N}$.

**Satz**

Ist $u \in V$ zu $l$ Brücken inzident, so besitzt $G$ mindestens $l$ von $u$ verschiedene Knoten von ungeradem Grad.

**Folgerung**

Haben in einem Graphen alle Knoten geraden Grad, so besitzt er keine Brücken.

# Distanz

Es sei $G = (V, E)$ ein Graph.

**Definition**

Es seien $v, w \in V$.

- Ist $v \sim w$, dann sei

  $$d(v, w) := \min\{l \in \mathbb{N}_0 \mid \text{in } G \text{ ex. } v\text{-}w\text{-Pfad der Länge } l\} \in \mathbb{N}_0.$$

- Ist $v, w \in V$ mit $v \not\sim w$, dann sei $d(v, w) := \infty$.
- Wir nennen $d(v, w)$ die *Distanz* zwischen $v$ und $w$.

# Distanz

Es sei $G = (V, E)$ ein Graph.

**Bemerkung**

Für alle $v, w \in V$ gelten:

- $d(v, w) = 0 \Leftrightarrow v = w$,
- $d(v, w) < \infty \Leftrightarrow v \sim w$.

$G$ ist genau dann zusammenhängend, wenn gilt:
$d(v, w) < \infty$ für alle $v, w \in V$.

# Breitensuche

Es sei $G = (V, E)$ ein Graph und $w \in V$.

Die *Breitensuche* ist ein Algorithmus, der, beginnend mit $w \in V$, alle Knoten der Zusammenhangskomponente von $w$ mit aufsteigender Distanz zu $w$ durchläuft.

**Anwendungen**

- Berechnung der Zusammenhangskomponenten von $G$.
- Berechnung der Distanzen $d(v, w)$ für $v$ in der Zusammenhangskomponente von $w$.
- Berechnung kürzester Pfade von jedem $v$ zu $w$.

# Breitensuche (Forts.)

```
BREITENSUCHE(Γ, w)
initialisiere array d[1, ..., n] mit allen Einträgen gleich ∞
initialisiere array p[1, ..., n] mit allen Einträgen gleich NIL
initialisiere leere queue Q (FIFO)
d[w] ← 0
INSERT(Q, w)
while Q ist nicht leer
do v ← EXTRACT(Q)
    for u ∈ Γ(v)
    do if d[u] = ∞
          then INSERT(Q, u)
               d[u] ← d[v] + 1
               p[u] ← v
return d, p
```

# Breitensuche (Forts.)

**Kommentare** zum Algorithmus)

- Eingabe:
  - $\Gamma$: Adjazenzliste des Graphen $G = (V, E)$ mit $V = \underline{n}$
  - $w$: Knoten $w \in V$
- Der `array` $d[1, \ldots, n]$ enthält nach der Terminierung an Position $v$ den Wert $d(w, v)$.
- Der `array` $p[1, \ldots, n]$ enthält nach der Terminierung an Position $v$ einen Knoten $u$, der auf einem $w$-$v$-Pfad der Länge $d(w, v)$ unmittelbar vor $v$ kommt.
- `queue` ist eine Warteschlange im „First-in-first-out“-Modus
- Der Aufruf $\textsc{Insert}(Q, x)$ hängt das Element $x$ an das Ende der Warteschlange.
- Der Aufruf $\textsc{Extract}(Q)$ entnimmt das Element, das am Anfang der Warteschlange steht.