

# Graphen

## Graphen und ihre Darstellungen

Ein *Graph* beschreibt Beziehungen zwischen den Elementen einer Menge von Objekten. Die Objekte werden als Knoten des Graphen bezeichnet; besteht zwischen zwei Knoten eine Beziehung, so sagen wir, dass es zwischen ihnen eine Kante gibt.

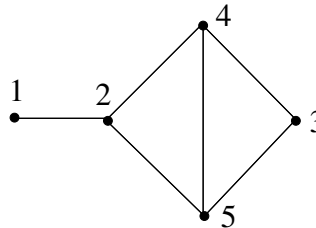
**Definition:** Für eine Menge  $V$  bezeichne  $\binom{V}{2}$  die Menge aller zweielementigen Untermengen von  $V$ . Ein *einfacher, ungerichteter Graph*  $G = (V, E)$  (kurz *Graph* genannt) besteht aus einer endlichen Menge  $V$  von *Knoten*, auch *Ecken* (*Vertex*) genannt, und einer Menge  $E \subseteq \binom{V}{2}$  von *Kanten* (*Edge*). Hier sind Kanten ungeordnete Paare von Knoten, d.h.  $\{u, v\}$  und  $\{v, u\}$  sind zwei verschiedene Schreibweisen für ein und dieselbe Kante. Im Gegensatz dazu ist endlicher *gerichteter Graph*  $G$  ein Paar  $(V, E)$  bestehend aus einer endlichen Knotenmenge  $V$  und einer Kantenmenge  $E$  von geordneten Knotenpaaren  $e = (u, v)$ , mit  $u, v \in V$ .

Ist  $e = \{u, v\}$  eine Kante von  $G$ , dann nennt man die Knoten  $u$  und  $v$  zueinander *adjacent* oder *benachbart* und man nennt sie *inzident* zu  $e$ .

Die Menge  $N(v) = \{u \in V \mid \{u, v\} \in E\}$  der zu einem Knoten  $v$  benachbarten Knoten wird die *Nachbarschaft* von  $v$  genannt. Der *Grad* eines Knotens  $v$  wird durch  $\deg(v) = |N(v)|$  definiert.

Die Anzahl der Knoten  $|V|$  bestimmt die *Ordnung* und die Anzahl der Kanten  $|E|$  die *Größe* eines Graphen.

Im Folgenden werden verschiedene Darstellungen von Graphen an einem Beispiel demonstriert.



1) Darstellung als Zeichnung:

2) Darstellung als *Adjazenzmatrix*. Jedem Knoten wird eine Zeile und eine Spalte zugeordnet und der Eintrag in der Zeile von  $u$  und der Spalte von  $v$  wird 1 gesetzt, wenn  $\{u, v\}$  eine Kante des Graphen ist (sonst 0):

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{pmatrix}$$

Während die Adjazenzmatrix eines ungerichteten Graphen symmetrisch ist (mit der Diagonalen als Symmetrieachse) sind Adjazenzmatrizen von gerichteten Graphen im allgemeinen nicht symmetrisch. Durch Verwendung beliebiger Zahlen für die einzelnen Einträge kann man auch Graphen mit Kantenbewertungen darstellen.

3) Darstellung als *Adjazenzliste*. Für jeden Knoten wird die Liste seiner Nachbarn angegeben (Liste von Listen):

1 : 2; 2 : 1, 4, 5; 3 : 4, 5; 4 : 2, 3, 5; 5 : 2, 3, 4; oder mit anderer Syntax  
(2), (1, 4, 5), (4, 5), (2, 3, 5), (2, 3, 4)

4) Darstellung als *Inzidenzmatrix*. Jedem Knoten wird eine Zeile und jeder Kante eine Spalte zugeordnet und der Eintrag in der Zeile von  $u$  und der Spalte von  $e$  wird 1 gesetzt, wenn  $u$  eine Ecke der Kante  $e$  ist (sonst 0):

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 9 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}$$

Die Darstellung mit Inzidenzmatrizen hat sich zwar für verschiedene theoretische Betrachtungen als nützlich erwiesen, ist aber für algorithmische Anwendungen eher ungeeignet. Ob man sich bei der Beschreibung und Implementierung von Graphalgorithmen für Adjazenzlisten oder für Adjazenzmatrizen entscheiden sollte, hängt zum einen von den Eigenschaften der zu bearbeitenden Graphen ab (oft werden Algorithmen für spezielle Graphklassen entwickelt), zum anderen von der Art der Zugriffe, die der Algorithmus vornehmen muss. Folgende Aspekte sind dabei zu beachten:

**1) Speicherverbrauch** Die Adjazenzmatrix verbraucht für Graphen mit  $n$  Knoten  $\Theta(n^2)$  Speicher, unabhängig von der Größe  $m = |E|$  des Graphen. Dagegen ist der Speicherbedarf einer Adjazenzliste nur  $\Theta(n + m)$  oder sogar  $\Theta(m)$ , wenn man leere Nachbarschaftslisten einfach ignoriert. Da für planare Graphen oder Graphen mit beschränktem Grad  $m = O(n)$  gilt, reduziert sich der Speicherbedarf auf  $O(n)$ .

**2) Anfragezeit** Häufig wird in Graphalgorithmen die Frage, ob zwei bestimmte Knoten adjazent sind, als Entscheidungsgrundlage für das weitere Vorgehen verwendet. Eine solche Abfrage kann mit der Adjazenzmatrix in konstanter Zeit beantwortet werden, bei Adjazenzlisten erhöht sich der Aufwand je nach Implementierung auf  $\Theta(n)$  (Listen der Nachbarn ungeordnet) oder  $\Theta(\log n)$  (geordnete Arrays mit Binärsuche oder AVL-Bäume).

**3) Updates** Das Einfügen und Streichen von Kanten erfolgt bei Adjazenzmatrizen in konstanter Zeit, bei Adjazenzlisten in linearer Zeit (oder in logarithmischer Zeit bei Verwendung von AVL-Bäumen oder ähnlichen Strukturen).

**4) Nachbarschaftsaufzählung** Um einen Nachbarn eines gegebenen Knoten zu erhalten sind die Adjazenzlisten mit konstanter Zeit (bzw. Zeit  $O(l)$  für die Liste aller  $l$  Nachbarn) im Vorteil gegen  $\Theta(n)$  bei Adjazenzmatrizen.

## Graphprobleme und Anwendungen

Da Graphen über eine sehr einfache Struktur verfügen, finden sie bei der Modellierung und algorithmischen Lösung vieler praktischer Probleme Anwendung, wie z.B.

- Modellierung von Straßen-, Flug- und Telefonnetzen
- Darstellung von Molekülen
- Interpretation von Relationen (Beziehungsgeflechten)
- Gerüste von Polyedern (lineare Optimierung)
- Entwurf von Mikrochips (VLSI-Design)

Die folgenden graphentheoretischen Aufgabenstellungen haben die die Entwicklung der Graphentheorie stark beeinflusst und unterstreichen die praktische Relevanz dieser Struktur:

**1) 4-Farben-Problem:** Man stelle sich die Welt mit einer beliebigen politischen Landkarte vor. Wir definieren einen Graphen, indem wir jedem Land einen Knoten zuordnen und zwei Knoten mit einer Kante verbinden, wenn sie einen gemeinsamen Grenzabschnitt haben. Wie viele Farben braucht man, um die Länder so einzufärben, dass benachbarte Länder verschiedene Farben haben. Man hat (mit Computerhilfe) bewiesen, dass vier Farben immer ausreichen! Einen Beweis ohne Computer gibt es bis heute nicht.

**2) Eulersche Graphen:** Man charakterisiere jene Graphen, bei denen man die Kanten so durchlaufen kann, dass man jede Kante einmal benutzt und man am Schluss wieder am Ausgangspunkt steht. ("Haus vom Nikolaus"-Prizip). Der Ausgangspunkt für diese Frage war das von Euler gelöste sogenannte Königsberger Brückenproblem.

**3) Hamiltonsche Graphen:** Dies sind solche Graphen, die man so durchlaufen kann, dass man jeden Knoten genau einmal besucht bis man zum Ausgangsknoten zurückkehrt. Während man für das vorherige Problem effiziente algorithmische Lösungen kennt, ist dieses algorithmisch schwer (NP-vollständig).

**4) Travelling Salesman Problem (TSP):** Oft hat man es mit bewerteten Graphen zu tun, das heißt Kanten und/oder Knoten haben zusätzliche Informationen wie Gewichte, Längen, Farben etc.

Ein Beispiel ist das TSP. Wir haben  $n$  Städte. Für jedes Paar  $\{u, v\}$  von Städten kennt man die Kosten, um von  $u$  nach  $v$  zu kommen. Man entwerfe für einen Handelsreisenden eine geschlossene Tour, die alle Städte besucht und minimale Gesamtkosten hat. Auch dies ist ein algorithmisch schweres Problem.

**5) Planare Graphen:** Welche Graphen lassen sich so in der Ebene zeichnen, dass sich Kanten nicht schneiden, also sich höchstens in Knoten berühren? Wie kann man sie charakterisieren und algorithmisch schnell erkennen?

**6) Flussprobleme:** Angenommen ein (Informations)-Netzwerk wird durch einen Graphen mit Kantenbewertung beschrieben, wobei diese Werte eine Obergrenze für die Übertragungskapazitäten der Kanten angeben. Wie groß ist dann der maximale (Informations)-Fluss zwischen zwei gegebenen Knoten  $s$  und  $t$ ?

## Grundlegende Eigenschaften und Standardbeispiele

**Satz:** Für jeden Graph  $G = (V, E)$  gilt  $\sum_{v \in V} \deg(v) = 2|E|$ , d.h.  $\sum_{v \in V} \deg(v)$  ist eine gerade Zahl.

**Beweis:** Bei Betrachtung der Inzidenzstruktur zwischen Knoten und Kanten ergibt sich die Aussage durch doppeltes Abzählen: Für jeden Knoten  $v$  ist die Anzahl inzidenter Kanten  $\deg(v)$  und jede Kante ist zu ihren zwei Eckknoten inzident. Damit erhalten wir  $\sum_{v \in V} \deg(v) = \sum_{e \in E} 2 = 2|E|$ .

**Folgerung:** Die Anzahl der Knoten mit ungeraden Grad ist in jedem Graphen gerade.

**Definition:** Seien  $G = (V, E)$  und  $G' = (V', E')$  zwei Graphen. Eine Abbildung  $\varphi : V \rightarrow V'$  wird *Graphhomomorphismus* genannt, falls für alle Kanten  $\{u, v\} \in E$  auch  $\{\varphi(u), \varphi(v)\} \in E'$  gilt. Ist darüber hinaus  $\varphi$  eine bijektive Abbildung und  $\varphi^{-1}$  auch ein Graphhomomorphismus, so nennt man  $\varphi$  einen *Graphisomorphismus* (und  $G, G'$  zueinander *isomorph*).

Die folgenden Standardbeispiele beschrieben formal gesehen nicht einzelne Graphen, sondern Isomorphieklassen.

1. Mit  $K_n$  ( $n \geq 1$ ) bezeichnet man den **vollständigen Graphen** der Ordnung  $n$ , d.h. eine Knotenmenge  $V$  mit  $|V| = n$  und der vollen Kantenmenge  $\binom{V}{2}$ .
2. Mit  $C_n$  ( $n \geq 3$ ) bezeichnet man den **Kreis** der Länge  $n$ , d.h. eine Knotenmenge  $\{v_1, v_2, \dots, v_n\}$  mit der Kantenmenge  $\{\{v_1, v_2\}, \{v_2, v_3\}, \dots, \{v_{n-1}, v_n\}, \{v_n, v_1\}\}$ .
3. Mit  $Q_n$  ( $n \geq 1$ ) bezeichnet man den **n-dimensionalen Würfel** mit Knotenmenge  $\{0, 1\}^n$  (Menge aller n-Tupel über  $\{0, 1\}$ ) wobei zwei Tupel dann und nur dann adjazent sind, wenn sie sich an genau einer Stelle unterscheiden.
4. Mit  $K_{n,m}$  ( $n, m \geq 0$ ) bezeichnet man den **vollständigen, bipartiten Graphen**, dessen Eckenmenge  $V$  die disjunkte Vereinigung von zwei Mengen  $A$  und  $B$  mit  $|A| = n$ ,  $|B| = m$  ist und dessen Kantenmenge aus allen Paaren  $\{a, b\}$  mit  $a \in A$ ,  $b \in B$  besteht.

**Definition:** Man nennt  $G' = (V', E')$  einen *Untergraph* von  $G = (V, E)$ , wenn  $V' \subseteq V$  und  $E' \subseteq E$  gilt. Ist außerdem  $E' = E \cap \binom{V'}{2}$ , so wird  $G'$  als *induzierter Untergraph* von  $G$  bezeichnet.

**Definition:** Ein Graph  $G = (V, E)$  wird *bipartit* genannt, wenn er Untergraph eines vollständigen, bipartiten Graphen ist. Etwas anschaulicher kann man formulieren,

daß ein Graph genau dann bipartit ist, wenn man die Knoten so mit zwei Farben einfärben kann, daß keine gleichfarbigen Ecken benachbart sind.

**Definition:** Das *Komplement* eines Graphen  $G = (V, E)$  ist der Graph  $\overline{G} = (V, \binom{V}{2} \setminus E)$ .

## Zusammenhang

**Definition:** Eine Folge von paarweise verschiedenen Ecken  $v_1, v_2, \dots, v_k$  eines Graphen  $G = (V, E)$  repräsentiert einen *Weg der Länge  $k - 1$* , falls  $\{v_i, v_{i+1}\} \in E$  für alle  $1 \leq i < k$ . Ist außerdem  $\{v_k, v_1\} \in E$ , so repräsentiert die Folge auch einen *Kreis der Länge  $k$* .

Man sagt, daß  $v$  von  $u$  *erreichbar* ist, falls ein Weg  $v_1, v_2, \dots, v_k$  mit  $v_1 = u$  und  $v_k = v$  in  $G$  existiert. Die Länge eines kürzesten Weges zwischen zwei Knoten nennt man ihren *Abstand* in  $G$ .

**Lemma:** Die Relation der Erreichbarkeit in der Knotenmenge  $V$  eines Graphen ist eine Äquivalenzrelation, d.h. sie ist reflexiv, symmetrisch und transitiv.

**Definition:** Die Knotenmenge  $V$  wird durch die Erreichbarkeitsrelation in Äquivalenzklassen zerlegt, wobei zwei Knoten  $u, v$  genau dann in derselben Klasse liegen wenn  $v$  von  $u$  erreichbar ist. Diese Klassen nennt man die *Zusammenhangskomponenten* (kurz *Komponenten*) des Graphen.  $G$  wird *zusammenhängend* genannt, wenn er genau eine Komponente hat.

Stellt man  $G$  durch eine Zeichnung dar, so kann man diesen Begriff anschaulich erklären: Zwei Knoten gehören zur selben Komponente, wenn man sie durch einen geschlossenen Kantenzug verbinden kann, wobei die Kanten nur in Knoten und nicht auf Kantenschnitten in der Zeichnung gewechselt werden dürfen.

**Satz:** Sind zwei Graphen isomorph so sind jeweils die Ordnung, die Größe, die sortierten Gradfolgen und die Anzahl der Komponenten der beiden Graphen gleich.

**Definition:** Seien  $u, v$  Knoten in einem ungerichteten Graphen  $G = (V, E)$ . Sind  $u$  und  $v$  in einer gemeinsamen Zusammenhangskomponente von  $G$ , so definieren wir ihren *Abstand*  $d(u, v)$  als Länge eines kürzesten Weges (Anzahl der Kanten des Weges) von  $u$  nach  $v$ . Gehören sie zu verschiedenen Komponenten, so setzen wir  $d(u, v) = \infty$ .

**Definition:** Der *Durchmesser*  $D(G)$  des Graphen ist definiert als das Maximum über alle paarweisen Abstände zwischen Knoten.

**Satz:** Ein Graph ist genau dann bipartit, wenn alle in ihm als Untergraph enthaltenen Kreise gerade Länge haben.

**Beweis:** Zunächst überlegt man sich, dass wir den Graphen als zusammenhängend voraussetzen können, ansonsten führt man den folgenden Beweis für jede Zusammenhangskomponente.

Sei  $G = (V, E)$  bipartit, das heißt,  $V = A \cup B$  mit  $A \cap B = \emptyset$  und Kanten verlaufen nur zwischen Knoten aus  $A$  und Knoten aus  $B$ . Sei des weiteren  $C$  ein Kreis in  $G$ .  $C$  benutzt abwechselnd Knoten aus  $A$  und  $B$  und hat somit gerade Länge.

Wir zeigen die andere Richtung. Wir fixieren einen beliebigen Knoten  $u \in V$ . Wir definieren:  $A = \{v \in V \mid d(u, v) \text{ gerade}\}$ ,  $B = V \setminus A$ . Zu zeigen, es gibt keine Kanten zwischen Knoten aus  $A$  (bzw. aus  $B$ ). Wir führen einen indirekten Beweis:

Wir nehmen an, es gibt eine Kante  $\{v, w\}$ ,  $v, w \in B$  (für  $A$  analog) und finden einen Widerspruch zur Annahme, dass alle Kreise gerade Länge haben. Wir betrachten kürzeste Wege von  $u$  zu  $v$  und zu  $w$ . Diese Wege haben gleiche Länge! (wegen der Kante zwischen  $v$  und  $w$ ) Sei  $x$  der letzte gemeinsame Knoten auf beiden Wegen. Dann bilden die beiden Wegabschnitte von  $x$  nach  $v$  bzw. nach  $w$  zusammen mit der Kante  $\{v, w\}$  einen Kreis ungerader Länge.