Graphen – einige Definitionen

Ein ***Graph*** $G = (V, E)$ besteht aus einer Menge von ***Knoten*** V (engl. Vertices) und ***Kanten*** E (engl. Edges).

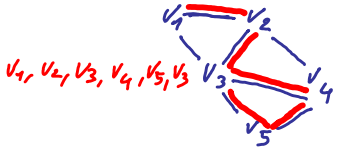
Eine Kante ist ein Paar (v, w) mit $v, w \in V$. Ist dieses Paar geordnet, so heisst die Kante ***gerichtet*** (directed).

Die Kante (v, w) ***verbindet*** die Knoten v und w (v ist mit w verbunden), falls $(v, w) \in E$ gilt. In einem ***ungerichteten*** Graphen mit Kante (v, w) ist w mit v **und** v mit w verbunden (engl. v is adjacent to w).

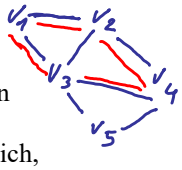
Manchmal hat eine Kante eine dritte Komponente. Man bezeichnet diese als ***Gewicht*** (weight), sie wird aber je nach Bedeutung z.B. auch ***Kosten oder Distanz*** genannt.

Ein Knoten v hat ***Grad*** n (degree n) $d(v) = n$, wenn er mit n Nachbarn verbunden ist. In einem gerichteten Graphen unterscheidet man zusätzlich die Anzahl der eingehenden und ausgehenden Kanten. Man spricht dann auch vom ***Eingangsgrad*** $d^-(v)$ (in-degree) und vom ***Ausgangsgrad*** $d^+(v)$ (out-degree) eines Knotens v.

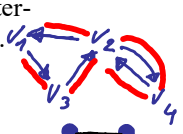
Ein ***Pfad*** (path) ist eine Sequenz von Knoten $v_1, v_2, v_3, \ldots v_n$ mit $(v_i, v_{i+1}) \in E$, $1 \leq i < n$.
Die ***Länge eines Pfades*** entspricht seiner Anzahl Kanten, also $n-1$.
Ein Pfad der Länge 0 - also ein Pfad ohne Kanten - ist demnach ein einzelner Knoten.



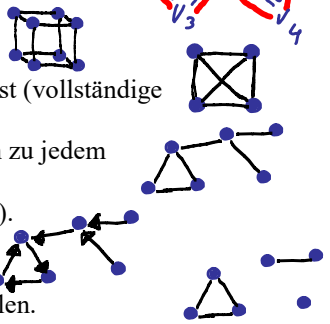
Ein ***einfacher Pfad*** (simple path) enthält lauter verschiedene Knoten, mit der Ausnahme, dass der erste und letzte Knoten gleich sein dürfen.



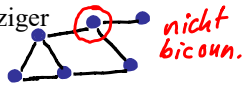
Eine ***Schlinge*** (loop) ist eine Kante (v, v) von einem Knoten v auf sich selbst.
Ein ***Zyklus*** (cycle) in einem gerichteten Graphen ist ein Pfad der Mindestlänge 1, so dass $w_1 = w_n$; solch ein Zyklus ist einfach, falls er durch einen einfachen Pfad beschrieben wird.
Weil in einem ungerichteten Graphen (u, v) und (v, u) dieselbe Kante bezeichnet, verlangt man hier zusätzlich, dass die Kanten in einem Zyklus unterschiedlich sind. In einem gerichteten Graphen sind (u, v) und (v, u) unterschiedlich (ev. auch mit unterschiedlichen Gewichten), so dass Sie zusammengenommen einen Zyklus bilden.
Ein Graph heisst ***schlicht***, wenn er weder Parallelkanten noch Schlingen aufweist.



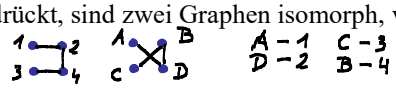
In ***regulären Graphen*** hat jeder Knoten dieselbe Anzahl Nachbarn, also $(\forall v \in V: d(x) = const)$.
Ein ungerichteter Graph ist ***vollständig***, falls jeder Knoten mit jedem anderen Knoten verbunden ist (vollständige Graphen sind somit Spezialfälle von regulären Graphen).
Ein ungerichteter Graph ist ***zusammenhängend*** (connected) falls es einen Pfad von jedem Knoten zu jedem anderen Knoten gibt. *Achtung*: dies ist nicht unbedingt ein vollständiger Graph.
Ein gerichteter Graph mit dieser Eigenschaft heisst ***stark zusammenhängend*** (strongly connected).
Ist ein gerichteter Graph nicht stark zusammenhängend, aber als ungerichteter Graph betrachtet zusammenhängend, dann heisst er ***schwach zusammenhängend*** (weakly connected).
Nicht zusammenhängende Graphen erkennt man daran, dass sie in zwei oder mehrere Teile zerfallen.



Ein zusammenhängender, ungerichteter Graph ist ***zweifach zusammenhängend*** (biconnected) falls kein einziger Knoten entfernt werden kann, ohne dass der Graph auseinander fällt.



Zwei Graphen G = (V, E) und G' = (V', E') sind ***isomorph***, falls die Knoten von G derart numeriert werden können, dass G identisch zu G' wird. Umgangssprachlich ausgedrückt, sind zwei Graphen isomorph, wenn sie gleich sind. Diese Eigenschaft ist nicht einfach festzustellen.



Eine ***Nachbarschaftsmatrix*** oder ***Verbindungsmatrix*** (adjacency matrix) A gibt an, welche Knoten mit welchen anderen benachbart / verbunden sind. Für jede Kante (u, v) wird $A[u][v]$ auf `true` gesetzt, alle anderen Einträge dieser Matrix sind `false`. Falls es sich um gewichtete Kanten handelt, so wird das Gewicht anstelle des bool'schen Wertes eingesetzt. Keine Verbindung wird dann entweder mit einem sehr grossen oder sehr kleinen Gewicht angegeben. Der Speicherbedarf für eine Verbindungsmatrix ist immer $O(|V|^2)$.

Viele Graphen haben nur eine dünn besetzte Verbindungsmatrix. Die meisten Einträge werden daher nicht verwendet und die ganze Matrix verschwendet viel Platz. Ein ökonomischerer Umgang mit Speicher haben die ***Nachbarschaftslisten*** oder ***Verbindungslisten*** (adjacency lists). Für jeden Knoten unterhält man eine Liste mit benachbarten Knoten. Der Speicherbedarf für Verbindungslisten ist nur noch $O(|V| + |E|)$, d.h. linear in bezug auf die Grösse des Graphen. Graphalgorithmen mit ***linearer Laufzeit*** haben also eine Komplexität von $O(|V| + |E|)$.