

BRIGITTE LUTZ-WESTPHAL¹

Wie komme ich optimal zum Ziel? Unterricht über Kürzeste-Wege-Algorithmen für Graphen

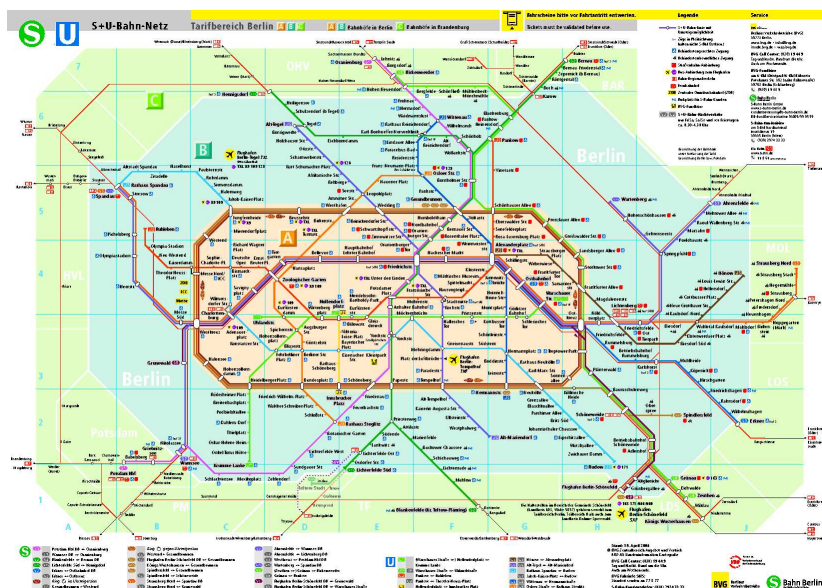
¹Konrad-Zuse-Zentrum für Informationstechnik Berlin (ZIB), Takustraße 7, D-14195 Berlin, lutz-westphal@zib.de.
Arbeitsplatzadresse: Technische Universität Berlin, Fakultät II - Mathematik und Naturwissenschaften, Institut für Mathematik, Sekr. MA 6-2, Straße des 17. Juni 136, D-10623 Berlin, westphal@math.tu-berlin.de.

Wie komme ich optimal zum Ziel? Unterricht über Kürzeste-Wege-Algorithmen für Graphen

1 Eine Frage aus dem Alltag

Die Frage nach dem kürzesten, schnellsten oder billigsten Weg taucht im Alltag häufig auf. Z. B. haben Schülerinnen und Schüler ein großes Interesse an einem möglichst schnellen Schulweg. Sie möchten vom Klassenzimmer auf möglichst kurzem Wege zum Brötchenverkauf kommen oder es wird eine Reise geplant, bei der die Fahrtkosten möglichst gering gehalten werden sollen. In der im Folgenden skizzierten Unterrichtseinheit dienen diese Fragen als Ausgangspunkt um ein klassisches Problem der kombinatorischen Optimierung kennen zu lernen. Die algorithmische Lösung, die die Schülerinnen und Schüler dabei selber erarbeiten, kommt in vielen realen Anwendungen vor, z. B. bei Routenplanern, Auto-Navigationssystemen oder bei der Wegberechnung für Datenpakete im Internet.

Die hier vorgestellten Unterrichtsideen wurden im Rahmen des von der Volkswagenstiftung geförderten Projektes „Diskrete Mathematik für die Schule“² (Konrad-Zuse-Zentrum Berlin/Technische Universität Berlin) entwickelt und in Unterrichtsversuchen in Berlin und Tübingen erprobt.³ Sie sind für den Einsatz ab Klasse 11 gedacht, eignen sich mit leichten Modifikationen aber auch schon in der Mittelstufe.



Zum Einstieg in diesen Themenbereich benötigt man einen U-Bahn-Netzplan (der Berliner U- und S-Bahnplan im Internet: http://www.bvg.de/images/sunetz_aktuell.pdf) oder auch einen Buslinienplan oder Ähnliches. Worauf es beim Einstiegsbeispiel ankommt ist, dass die Strecken zwischen den Stationen nicht allzu unterschiedlich lang sind. Die Schülerinnen und Schüler bekommen die Aufgabe herauszufinden, welches der kürzeste Weg von Station A nach Station B ist. Dabei sollte man möglichst Stationen wählen, die den Schülerinnen und Schülern bekannt sind (z. B. die Schulhaltestelle) und die durch mehrere verschiedene Wege verbunden sind. Im Idealfall gibt es mehrere gleichlange Wege oder auch einen überraschenden, der sozusagen „hintenherum“ geht, aber kürzer ist als alle anderen.

2 Problemanalyse/Modellierung

Was heißt hier kürzester Weg? Im Allgemeinen möchte man ja so wenig Zeit wie möglich in Bus und U-Bahn verbringen. Die Fahrzeiten gehen jedoch aus so einem Netzplan gar nicht hervor. Auch die Länge der eingezeichneten Strecken zwischen den Stationen sagt nichts über die Fahrzeit aus, zumal die meisten solcher Netzpläne die geographischen Gegebenheiten nur sehr verzerrt wiedergeben.

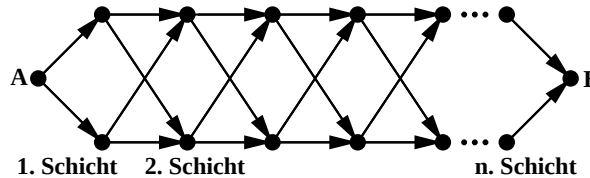
Die Problemstellung scheint zunächst vollkommen klar zu sein, doch dann stellt sich heraus, dass Entscheidungen getroffen werden müssen: Optimieren wir die Fahrzeit, die Strecke, die Kosten oder die Anzahl der Stationen? Je nach dem, für welches Kriterium man sich entscheidet, muss man sich um das entsprechende Datenmaterial (Fahrpläne, Entfernungstabellen o. ä.) bemühen, was eine interessante Zusatzaufgabe für die Klasse sein kann.

Stationen zählen ist das, was man „von Hand“ und spontan tut, wenn man vor solch einem Netzplan steht und sich für eine Route entscheiden muss. Die Optimierung der Anzahl der Stationen ist zumindest im großstädtischen Schnellbahnnetz oder in einem Innenstadt-Busnetz sinnvoll, da dort die Fahrzeiten zwischen den Stationen wenig variieren. Somit hat man eine erste Modellierung des Problems gefunden und kann nun die Suche nach Lösungen beginnen. Wie man vorgehen muss, um z. B. Fahrzeiten auch berücksichtigen zu können, wird in einer späteren Phase des Unterrichts thematisiert.

3 Algorithmen suchen

Das Abzählen von Stationen und Suchen nach guten Verbindungen aktiviert die Lerngruppe und setzt gleich die ersten Diskussionen in Gang. Meist tritt auch die Situation auf, dass jemand anderes einen besseren Weg gefunden hat als man selber, so dass sich schnell herausstellt, dass das Verfahren „draufgucken und abzählen“ nicht unbedingt zuverlässig arbeitet.

Wie müsste denn ein zuverlässiges Verfahren aussehen? Eine Möglichkeit ist, alle A-B-Wege zu suchen und dann den oder (falls es mehrere gleich kurze gibt) einen kürzesten Weg auszuwählen. Dieses Verfahren ist korrekt, hat aber einen Haken: Die Anzahl möglicher Wege kann unter Umständen gigantisch groß werden, wie man an folgendem Beispiel sehen kann⁴.



Mit jeder „Schicht“ des Graphen verdoppelt sich die Anzahl der A-B-Wege. Mit n Schichten hat man also 2^n Wege. Allgemein ausgedrückt, die Anzahl der verschiedenen Wege wächst exponentiell! Der Graph mit 50 Schichten hat nur 102 Knoten, aber 1,1 Billionen A-B-Wege! Dieses Phänomen der ins Unermessliche wachsenden Kombinationen bei recht kleinen Grundmengen (hier die Anzahl der Knoten) nennt man die „kombinatorische Explosion“.

Selbst mit Hilfe sehr schneller Computer kommt man mit dem Verfahren der Enumeration (also „Abzählen“ aller Möglichkeiten) nicht in annehmbarer Zeit zum Ziel. Nimmt man etwa an, ein Computer kann 1 000 000 solcher Wege in einer Sekunde berechnen, so bräuchte er bei 25 Schichten etwa eine halbe Minute, bei 30 Schichten knappe 18 Minuten, bei 40 Schichten fast 13 Tage, bei 50 Schichten mehr als 35 Jahre! Lassen Sie Ihre Schülerinnen und Schüler solche Beispiele ausrechnen und die kombinatorische Explosion selbst erfahren.

Da also die Enumeration für die meisten größeren Beispiele nicht praktikabel ist, muss man sich eine Vorgehensweise zum Auffinden eines kürzesten Weges überlegen, die effizienter arbeitet.

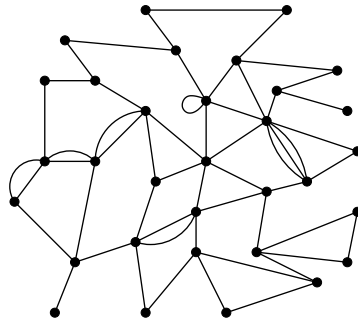
4 Graphen

An dieser (oder auch anderer) Stelle im Unterricht kann man einen Exkurs über Graphen machen. Neben der Einführung einiger Fachbegriffe kommt dadurch eine weitere Abstraktionsebene hinzu. Dass die Überlegungen zu kürzesten Wegen in der U-Bahn leicht auf andere Realsituationen übertragbar sind, liegt auf der Hand, wenn klar geworden ist, dass das Nahverkehrsnetz als Graph betrachtet werden kann und was Graphen alles darstellen können (siehe Arbeitsblatt 1).

⁴Dieses Beispiel ist dem Buch „Das Geheimnis des kürzesten Weges“ von Peter Gritzmann und Rene Brandenberg, Springer, 2002, entnommen

Der Liniennetzplan ist eigentlich schon ein Graph:

Definition: Ein Graph ist ein Gebilde aus Knoten und Kanten (wobei die Kanten stets in Knoten beginnen und enden).



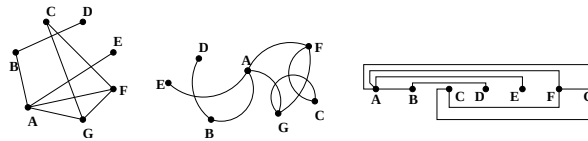
Eine weitere Überlegung bringt wichtige Grapheneigenschaften zutage. Wir suchen in dieser Unterrichtseinheit nach Strategien, wie Computer das kürzeste-Wege-Problem lösen können. Dafür ist es notwendig, dass der Computer den zugrundeliegenden Graphen kennt. Wie kann man Graphen so darstellen, dass sie in den Computer eingegeben werden können? Die Angabe von Koordinaten für die Knoten ist nicht immer sinnvoll und beinhaltet mehr Information als wir benötigen. Für unsere Fragestellung reicht es zu wissen, welcher Knoten mit welchem verbunden ist. Eine der gebräuchlichen Datenstrukturen für Graphen ist die Adjazenzmatrix:

	A	B	C	D	E	F	G
A	0	1	0	0	1	1	1
B	1	0	0	1	0	0	0
C	0	0	0	0	0	1	1
D	0	1	0	0	0	0	0
E	1	0	0	0	0	0	0
F	1	0	1	0	0	0	1
G	1	0	1	0	0	1	0

Die Schülerinnen und Schüler können selbst herausfinden, wie man aus solchen Tabellen (Arbeitsblatt 2) die nötigen Informationen herausliest. Zeichnen sie nun den Graphen, so erleben sie direkt wichtige Eigenschaften von Graphen: Die Position der Knoten spielt keine Rolle, die Länge und Form der Kanten ebensowenig. Ein Graph kann verschiedene Darstellungen besitzen.

Definition: Zwei Graphen heißen isomorph, wenn sie - bei entsprechender Benennung der Knoten - die gleiche Matrix besitzen.

Zum Nachweis der so definierten Isomorphie (Arbeitsblatt 3) werden daher auch immer Knotennamen benötigt, z. B. so:

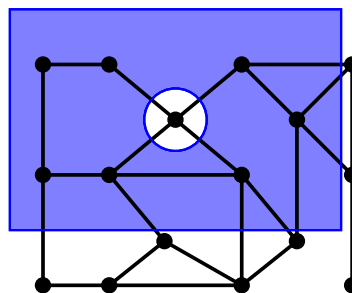


Will man noch tiefer in die Graphentheorie vordringen, so ist das Entdecken und Beweisen des Satzes über die Anzahl der ungeraden Knoten sehr reizvoll (Arbeitsblatt 4). Dass die Anzahl der Knoten mit ungeradem Grad (Grad: Anzahl der Kanten, die von einem Knoten ausgehen, wobei Schlingen doppelt gezählt werden) stets gerade ist, ist ein wichtiger Satz, der u. a. im Zusammenhang mit der Optimierung von Touren der Müllabfuhr benötigt wird.

5 Die Froschperspektive des Computers

Wie wir inzwischen schon festgestellt haben, brauchen wir zur Lösung des kürzeste-Wege-Problems ein Verfahren, das schneller geht als die Enumeration aller möglichen Wege. Wir wollen nun also versuchen, ein geschickteres Verfahren zu finden, das in beliebigen Graphen einen kürzesten A-B-Weg konstruiert.

Die Hauptschwierigkeit beim Erfinden von Graphenalgorithmien liegt darin einzusehen, dass die nächstliegende Vorgehensweise, die durch genaues Anschauen des Graphen gefunden wird, im allgemeinen Fall nicht funktioniert. Das kommt daher, dass man beim Draufgucken meist viele Arbeitsschritte gleichzeitig tut, ohne sich dessen bewusst zu sein. Eine gute Hilfe zum Auffinden dieser einzelnen Schritte ist die Vorstellung, dass der Computer einen Graphen aus der Froschperspektive „sieht“. Als Veranschaulichung dafür hat sich die Lochblende sehr bewährt. Der Computer sieht immer nur den Knoten, auf dem er sich gerade befindet, und die Kanten, die von diesem Knoten ausgehen. Was sich jenseits dieses Guckloches befindet, kann er nicht sehen.



Also setzen wir das Guckloch auf den Startknoten und sehen lediglich, wie viele Kanten von diesem Knoten ausgehen. Mehr Informationen, z. B. in welcher Richtung sich

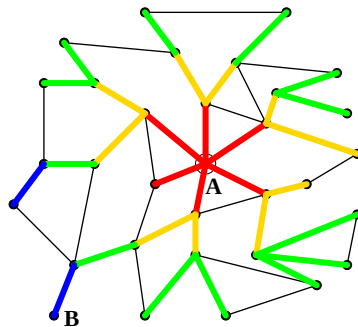
der Zielknoten befindet, bekommen wir nicht. Wie muss man nun vorgehen, um einen kürzesten Weg zum Ziel zu finden?

Eine Möglichkeit, diese „Froschperspektive“ des Computers noch direkter zu erfahren, ist, in einem größeren Unterrichtsraum oder noch besser im Schulhof einen Graphen auf den Boden zu zeichnen und eine Schülerin oder einen Schüler zu bitten, sich mit einer bunten Kreide in der Hand auf einen Knoten zu hocken. Welche Anweisungen müssen gegeben werden, damit diese Person den besten Weg findet? Gibt es Dinge, die sie sich notieren muss? Welche Knoten oder Kanten müssen markiert werden, damit die Wegekonstruktion des Algorithmus nachvollziehbar wird? Kehren bestimmte Anweisungen immer wieder, d. h. gibt es Schleifen im Algorithmus?

Es ist äußerst wichtig, sich an dieser Stelle so viel Zeit im Unterricht zu nehmen, dass die Schülerinnen und Schüler kreativ ans Werk gehen können, und versuchen, eigene Verfahren zu entwickeln. Das Produkt soll eine Schritt-für-Schritt-Anweisung sein, die einen kürzesten Weg zwischen den jeweiligen Start- und Zielknoten produziert. Diese Anweisung soll umgangssprachlich formuliert werden. Ziel dabei ist, die Grundideen des Algorithmus exakt zu fassen. Programmiersprachen werden hier nicht benötigt, falls sie nicht von der Lerngruppe bereits beherrscht werden. Erst die konkrete Implementierung erfordert die Ausformulierung in einer Programmiersprache. Dies kann idealerweise in fächerübergreifender Zusammenarbeit mit dem Informatikunterricht geschehen

6 Der erste Algorithmus: die Breitensuche

Bei der Optimierung über die Anzahl der Stationen wird ein Verfahren herauskommen, das der Breitensuche oder „Breadth-First-Search (BFS)“ ähnelt. Die Breitensuche lässt sich so veranschaulichen (die verschiedenen Farben symbolisieren die einzelnen Iterationen des Algorithmus):



Bei der Breitensuche wird von je zwei gleichlangen Wegen immer nur die zuerst aufgefundene Alternative gewählt, so dass das Endprodukt ein eindeutiger kürzester Weg von A nach B ist. Lässt man den Algorithmus so lange arbeiten, bis jeder Knoten besucht wurde, so erhält man kürzeste Wege vom Startknoten zu allen anderen Knoten.

Ein Beispiel für eine Formulierung der Breitensuche:

Gegeben: ein Graph mit Knotennamen, ein Startknoten S (einen Zielknoten braucht man nicht unbedingt anzugeben, da die Breitensuche kürzeste Wege zu allen anderen Knoten konstruiert).

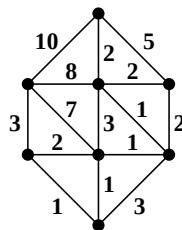
Gesucht: kürzeste Wege (bzgl. Kantenzahl) vom Startknoten zu allen anderen Knoten.

1. Wähle einen Startknoten.
2. Markiere alle Kanten, die von diesem Knoten ausgehen.
3. Schreibe alle Nachbarknoten des Startknotens in eine Liste.
4. Streiche den ersten Knoten aus der Liste.
5. Markiere alle von diesem Knoten ausgehenden Kanten, sofern sie noch nicht markiert sind, und nicht im Endknoten auf bereits markierte Kanten treffen.
6. Schreibe die Knoten, die am Ende dieser neu markierten Kanten sind, hinten in die Liste.
7. Wiederhole 4., 5. und 6. so lange, bis die Liste leer ist.

Man kann den Detaillierungsgrad noch erhöhen, wenn man überlegt, wie eigentlich die konstruierten Wege abgespeichert werden, und ob die zurückgelegten Distanzen auch notiert werden sollen. Mit solchen Überlegungen kommt man schnell zur Frage nach den passenden Datenstrukturen und kann, wenn genügend Zeit vorhanden ist, einen Ausflug in das Grenzgebiet zwischen Mathematik und Informatik machen. Z. B. ist die hier für die Breitensuche verwendete Liste, die nach dem Prinzip „first in first out“ (FIFO) funktioniert, eine elementare Datenstruktur, eine sogenannte „Queue“.

7 Gewichtete Graphen: eine Verfeinerung des Modells

Will man nicht Wege im U-Bahnnetz optimieren, sondern z. B. Bahn- oder Autofahrten, so spielt die Fahrzeit zwischen zwei Orten eine wesentliche Rolle (wahlweise auch die Kosten oder die Streckenlänge, die Modellierung ist dieselbe). Diese Größen gibt man in Form von sogenannten Kantengewichten an (die als Einträge in der Adjazenzmatrix auftauchen):



8 Der (gierige) Algorithmus von Dijkstra

Die Breitensuche konstruiert für gewichtete Graphen keine kürzesten Wege. Ein anderer Algorithmus wird gebraucht. Dies ist wieder eine Stelle, an der die Schülerinnen und Schüler ganz selbständig „forschen“ können, zumal sie inzwischen schon einige Erfahrungen im Umgang mit Graphen gesammelt haben. Der Algorithmus, der hier vermutlich gefunden wird, ist ein berühmter Algorithmus der kombinatorischen Optimierung: der Algorithmus von Dijkstra. Er findet an verschiedenen Stellen Anwendung, z. B. bei der Fahrplanauskunft, bei Routenplanern, in Auto-Navigationssystemen, aber auch überall dort, wo kürzeste Wege als Teilproblem auftauchen, etwa bei der Optimierung von Briefträgertouren.

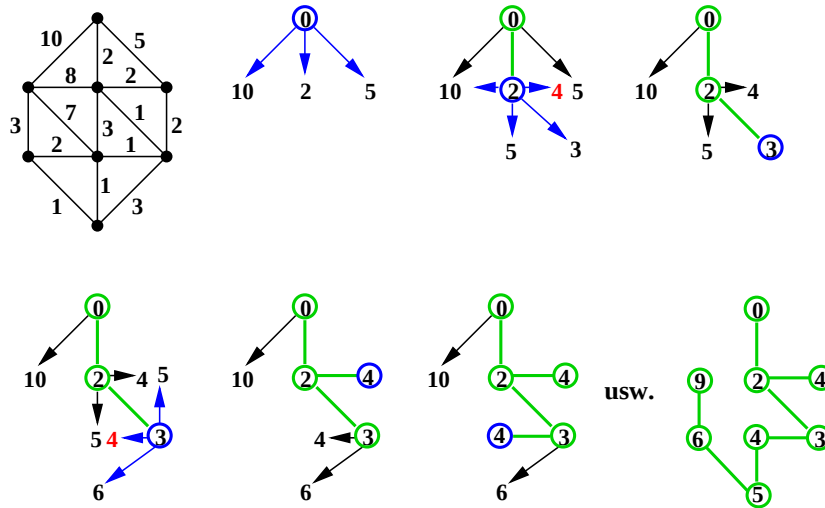
Ein Beispiel für eine Formulierung des Algorithmus von Dijkstra:

Gegeben: ein gewichteter Graph mit Knotennamen, ein Startknoten

Gesucht: Kürzeste Wege vom Startknoten zu allen anderen Knoten

(„Distanz“ bedeutet hier stets Weglänge zum Startknoten.)

1. Zu Beginn bekommen alle Knoten die Distanz unendlich.
2. Wähle einen Startknoten. Der Startknoten bekommt die permanente Distanz 0. Er ist nun der aktive Knoten.
3. Berechne die Distanzen (temporäre Distanzen) aller noch nicht mit permanenten Distanzen versehenen Nachbarknoten des aktiven Knotens: Kantengewicht der verbindenden Kante + Distanz des aktiven Knotens.
4. Ist diese neu berechnete Distanz für einen Knoten kleiner als die bereits vorhandene, so wird die vorherige Distanz gelöscht, die kleinere Distanz notiert, und der aktive Knoten wird als Vorgänger dieses Knotens gespeichert. Ein vorher gespeicherter Vorgänger dieses Knotens wird gelöscht (Mit dieser Vorgängerliste lässt sich am Ende der Weg rekonstruieren). Ist die neu berechnete Distanz größer, als die bereits vorhandene, ändert sich nichts an Distanz und Vorgänger.
5. Wähle einen Knoten mit minimaler temporärer Distanz, diese Distanz wird nun unveränderlich (permanente Distanz). Dieser Knoten wird nun zum aktiven Knoten.
6. Wiederhole 3.-6. so lange, bis alle Knoten permanente Distanzmarken haben.



Der Algorithmus von Dijkstra ist ein gieriger oder Greedy-Algorithmus, weil er immer lokal optimal entscheidet. Glücklicherweise funktioniert es beim kürzeste-Wege-Problem, dass mit Hilfe der gierigen Methode ein globales Optimum gefunden wird. Bei anderen Problemstellungen versagt die Greedy-Strategie, beispielsweise beim Traveling-Salesman-Problem. (Verweis auf Artikel von Gritzmann/Brandenberg)

9 Zusammenfassung

Das Erfinden und Formulieren von Algorithmen ist eine sehr komplexe Aufgabe, die wichtige Kompetenzen wie Modellieren, Problemlösen und das mathematische Argumentieren und Kommunizieren fördert. Da die Ausgangsfrage aus dem Alltag der Schülerinnen und Schüler kommt, ergibt sich ein direkter Zugang zum Problem. Dazu kommt, dass man mit wenig Theorie (im Sinne von Definitionen und komplizierten Rechenverfahren) dennoch tief in die Materie eindringen kann. Rein mathematische Fragestellungen, die u. a. zu Resultaten der Graphentheorie oder Kombinatorik führen können, erwachsen wie von selbst aus der Beschäftigung mit einer Anwendung.

Unterrichtsverlauf (Beispiel)

- 1./2. Stunde: Einarbeitung in das Problem, erste Heuristiken, Enumeration, kombinatorische Explosion
- 3./4. Stunde: Graphen
- 5./6. Stunde: Breitensuche
- 7./8. Stunde: der Algorithmus von Dijkstra
9. Stunde: Ausblick: NP-schwere Probleme

10 Literatur:

P. Gritzmann und R. Brandenburg, Das Geheimnis des kürzesten Weges. Ein mathematisches Abenteuer, Springer Berlin, Heidelberg, 2. Auflage 2002

V. Heun, Grundlegende Algorithmen. Einführung in den Entwurf und die Analyse effizienter Algorithmen, Vieweg Braunschweig, 2000

Th. Cormen, Ch. Leiserson und R. Rivest, Introduction to Algorithms, The MIT Press, Cambridge, Massachusetts, 1990

Unterrichtssoftware:

Wird im Rahmen des Matheon-Projekts G6 entwickelt:

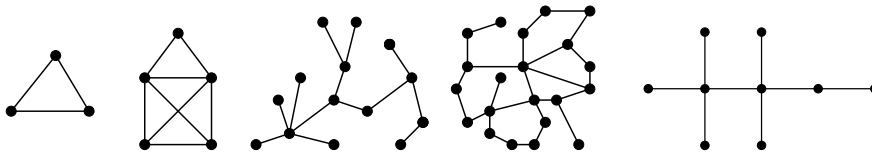
<http://www.math.tu-berlin.de/didaktik/tiki-index.php?page=Matheon+G6>

Modellierung durch Graphen

Um Fragestellungen wie z.B. die Optimierung von Wegen bearbeiten zu können, braucht man geeignete mathematische Modelle.

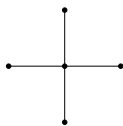
In der Kombinatorische Optimierung verwendet man dafür **Graphen**.

Ein Graph ist ein Gebilde aus Knoten und Kanten. Es werden im weitesten Sinne Beziehungen von verschiedenen Objekten zueinander dargestellt. Hier ein paar kleine Beispiele:

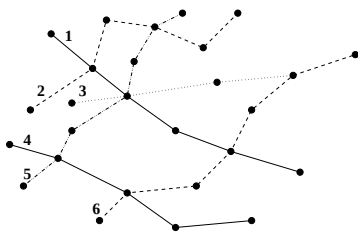


1. Was könnten diese Graphen darstellen? Geben Sie mehrere verschiedene Möglichkeiten an.

2. Dieser Graph soll eine Straßenkreuzung darstellen. Manchmal braucht man allerdings genauere Informationen über eine Kreuzung, z.B. welche Möglichkeiten es zum Abbiegen gibt. Zeichnen Sie einen Graphen, der die Kreuzung von zwei vierspurigen Straßen mit allen Abbiegemöglichkeiten darstellt.



3. Sie sehen hier ein U-Bahn-Liniennetz. Die Zahlen geben die Liniennummern an. Zeichnen Sie einen Graphen, der darstellt, welche Linien sich kreuzen.



Adjazenzmatrizen

	A	B	C	D	E	F	G
A	0	1	0	1	0	1	0
B	1	0	0	1	1	0	0
C	0	0	0	1	0	0	1
D	1	1	1	0	1	1	1
E	0	1	0	1	0	0	0
F	1	0	0	1	0	0	1
G	0	0	1	1	0	1	0

	A	B	C	D	E	F	G
A	0	0	1	1	0	0	1
B	0	0	1	0	0	0	0
C	1	1	0	0	1	0	1
D	1	0	0	0	0	1	0
E	0	0	1	0	1	0	1
F	0	0	0	1	0	0	0
G	1	0	1	0	1	0	0

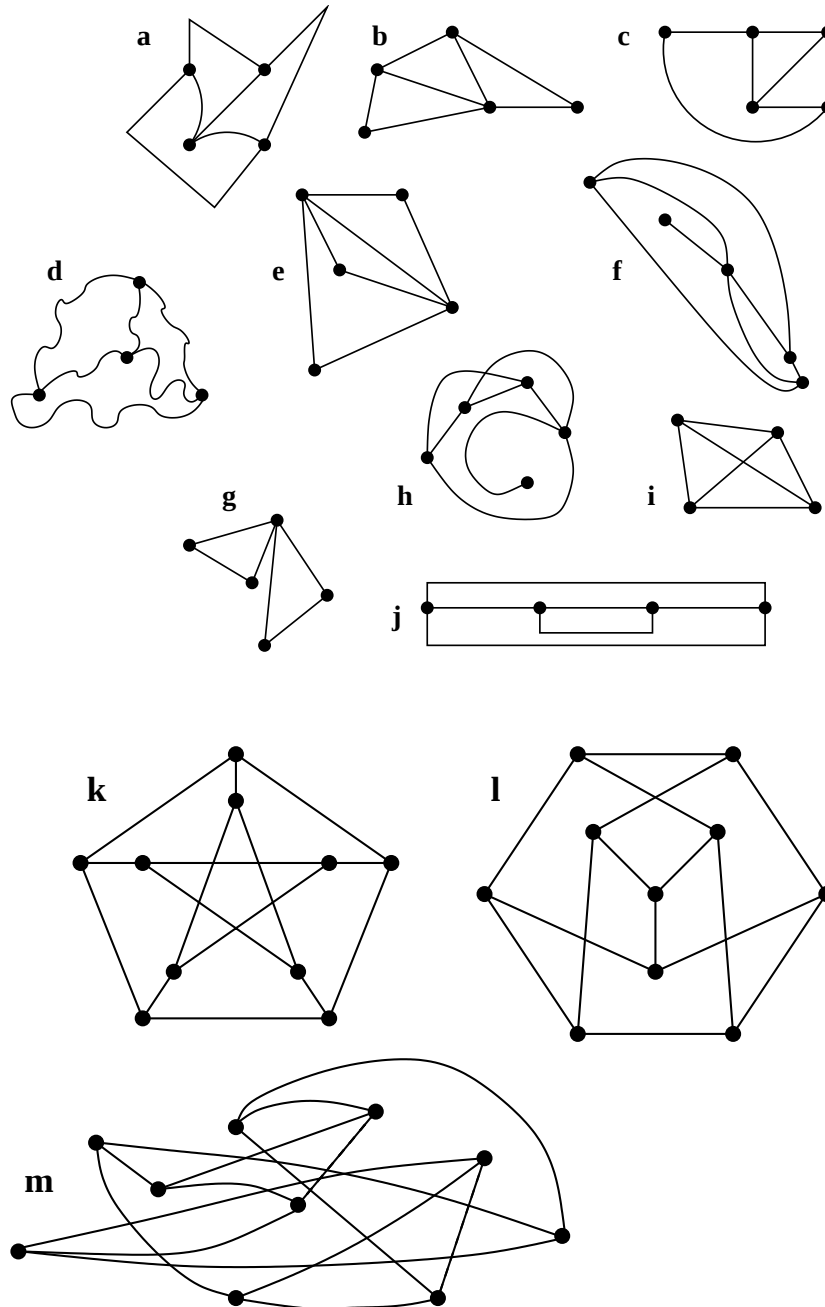
	A	B	C	D	E	F	G
A	0	0	0	0	1	1	1
B	0	0	0	1	1	0	0
C	0	0	0	0	0	0	0
D	0	1	0	0	0	1	1
E	1	1	0	0	0	1	1
F	1	0	0	1	1	0	1
G	1	0	0	1	1	1	0

	A	B	C	D	E	F	G
A	0	1	1	1	1	1	1
B	1	0	1	1	1	1	1
C	1	1	0	0	1	1	1
D	1	1	0	0	0	1	1
E	1	1	1	0	0	1	1
F	1	1	1	1	1	0	1
G	1	1	1	1	1	1	0

Graphen: Isomorphie

Welche dieser Graphen sind isomorph?

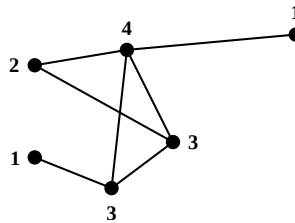
Welche Kriterien müssen überprüft werden? Stellen Sie eine Liste zusammen.



Graphen: „Handshaking Lemma“ und Anzahl der Knoten mit ungeradem Grad

Die Zahl der Kanten, die von einem Knoten abgehen (wobei Schlingen doppelt gezählt werden), nennt man den **Grad des Knotens**.

Hier ein Beispiel (die Zahlen an den Knoten geben den Grad an):



Aufgabe 1: Zwischen der Summe aller Knotengrade in einem Graphen und der Anzahl der Kanten in demselben Graphen gibt es einen Zusammenhang, das sogenannte „Handshaking Lemma“ (*Ein Lemma ist ein kleines mathematische Resultat bzw. ein Hilfssatz*).

- a) Versuchen Sie diesen Zusammenhang anhand von eigenen Beispielen zu finden.
- b) Formulieren Sie die Aussage und begründen (bzw. beweisen) Sie diese.
- c) Welche Interpretation des Graphen legt der Name des Lemmas nahe?

Aufgabe 2: Zeichnen Sie einen Graphen, der fünf Knoten mit ungeradem Grad hat. Hinter dieser Aufgabe steckt ein Satz über die Anzahl ungerader Knoten. Formulieren und beweisen Sie den Satz.

