Renate Thies

Unterrichtsentwurf im Fach Informatik

Schule: Evangelisch Stiftisches Gymnasium Gütersloh
Klasse: 13 IF GK
Datum: 4.9.2007

Thema der Unterrichtsreihe:
Einführung in Graphen – exemplarische Algorithmen aus der Graphentheorie

Thema der Stunde:
„Viele Wege führen nach Rom – aber welcher ist der kürzeste?“ – Auf der Suche nach dem Algorithmus von Dijkstra

Ziel der (Doppel)-Stunde:
Die Schüler[1] sollen ausgehend von einem „realen“ Problem für das graphentheoretische Problem „Ermittlung kürzester Wege“ sensibilisiert und motiviert werden und erkennen, dass das reale Problem „Ermittlung kürzester Wege“ graphentheoretisch modelliert und dann gelöst werden kann. Weiter sollen Sie den Algorithmus von Dijkstra (möglichst) selbstständig entdecken und verbalisieren können. (Minimales Stundenziel)

Die Schüler sollen den Algorithmus von Dijkstra[2] im Hinblick auf die Implementierung in Pseudocode formulieren und auf verschiedenen Graphen durchführen können. (Mögliches erweitertes Stundenziel)

Bemerkungen zur Lerngruppe
In der Lerngruppe, bestehend aus 13 Schülern, herrscht eine sehr angenehme Lernatmosphäre. In der Regel sind alle Schüler interessiert und motiviert bei der Sache, und es kommt kaum zu Ablenkungen.

Da bereits zwei Schüler parallel Informatik-Vorlesungen an der Uni/FH besuchen, verfügen diese teilweise über besonders gute Fähigkeiten des „informatischen bzw. algorithmischen Denkens“. Im Bereich der Graphentheorie haben allerdings auch diese Schüler keine Vorerfahrungen, weshalb gesonderte leistungsdifferenzierende Angebote in dieser Stunde nicht nötig sind. Dennoch muss insbesondere in Sicherungsphasen sichergestellt werden, dass alle Schüler die grundlegenden Zusammenhänge erfassen.

[1] In dieser Planung wird das generische Maskulinum verwendet, dabei ist die weibliche Schreibweise implizit mitgemeint.
[2] Es sei erwähnt, dass es „den“ Algorithmus von Dijkstra nicht gibt. Abhängig von Art und Weise der Benennung markierter Knoten (hängend, farbig, markiert, etc.) und der Entscheidung, ob nur die kürzeste Länge oder auch der Weg entscheidend ist, variiert die Darstellung des Algorithmus. In dieser Unterrichtsstunde sollen die Schüler also die zentrale Idee des Dijkstra-Algorithmus erkennen und formulieren. In Abhängigkeit von den Ideen der S. können dabei unterschiedliche Versionen „entdeckt“ werden.

Aufbau der Unterrichtsreihe

14.08.D	„Über wie viele Brücken kannst du gehen?“ – Entdeckung von Graphen am Königsberger Brückenproblem sowie Einführung und erste Übungen zu grundlegenden Begriffen und Modellierung der Datenstruktur „Graph“ (Klassendiagramm)
21.08.E	Analyse der Datenstruktur Graph aus den Zentralabiturvorgaben und kritischer Vergleich mit eigenen Modellierungen.
22.08.D	„Wenn das Naheliegende das Beste ist“ – Minimal aufspannende Bäume und Algorithmen zu deren Erstellung
28.08.E	entfällt wg. Kursfahrt
29.09.D	entfällt wg. Kursfahrt
03.09.E	Wiederholung und „Was man mit Graphen noch so alles zeigen kann…“ – Graphen als Hilfsmittel zur Lösung von Problemen
04.09.D	„Viele Wege führen nach Rom“ – Auf der Suche nach dem Algorithmus von Dijkstra
10.09.E	Ggf. Dijkstra in Pseudocode oder Ideen zur Implementierung
11.09.D	Beginn der Implementierungsphase

(Länge variiert in Abhängigkeit von Schülerleistungen)

Lernziele

Zum Erreichen des Stundenzieles ist das Erreichen der folgenden *Teillernziele* vorgesehen:

Die Schüler sollen

- den Zusammenhang zwischen Graphen und Straßenkarten erkennen.
- eine Straßenkarte als Graph modellieren können (verbal).
- ihre Fähigkeit, Probleme zu lösen, schulen, indem Sie den kürzesten Weg in einem gegebenen Graphen finden und…
- … über ihr Vorgehen zum Auffinden dieses Weges reflektieren und es beschreiben können.
- erste Ideen entwickeln, welche Anforderungen an einen Algorithmus zum Auffinden des kürzesten Weges gestellt werden sollten.
- einen gegebenen Graphen mit gegebenen Hilfsmitteln nachbilden können.
- erste Ideen für den Algorithmus von Dijkstra entwickeln, indem Sie den Graph/das Netz schrittweise hochheben und genau beobachten, was passiert.
- ihre Beobachtungen verbalisieren können, um so den Algorithmus von Dijkstra (teilweise) umgangssprachlich beschreiben zu können.

optional:

- den Algorithmus von Dijkstra in Pseudocode angeben können[3]

Methodisch-didaktischer Kommentar:

[3] Hier wird nicht erwartet, dass die S. dies völlig eigenständig ausführen können, da die S. bisher wenig Erfahrung im Pseudocode haben.

Der Algorithmus von Dijkstra

Mit Hilfe von Graphen können Ausschnitte der Wirklichkeit modelliert werden, um diese einfacher zu verstehen und zu analysieren. Landkarten sind Beispiele, die mit Hilfe von Graphen modelliert werden können. Orte bzw. Kreuzungen werden durch Knoten dargestellt und die Verbindungswege werden durch Kanten repräsentiert. Aber auch diverse andere Netzwerke können mit Hilfe von Graphen übersichtlich dargestellt werden.

Aufgrund der vielen Anwendungssituationen für Graphen existieren auch viele Algorithmen für Graphen. Ein bedeutendes Problem, das mit Hilfe von Graphen gelöst werden kann, ist die Berechnung des Kürzesten Weges zwischen zwei Orten bzw. Knoten („Single Source shortest Path“ SSSP). Ein klassischer Graphalgorithmus zur Lösung dieses Problems ist der sogenannte „Algorithmus von Dijkstra“, der nach seinem Entdecker Edsger Wybe Dijkstra benannt wurde.

Für weitere Informationen zum Dijkstra-Algorithmus sei auf die verwendete Literatur verwiesen.

Selbstständiges Entdecken des Algorithmus von Dijkstra – ein Wagnis?!

Im Sinne eines konstruktivistischen Menschenbildes ist das selbstständig entdeckende Lernen eindeutig dem durch Vorgaben geleiteten Lernprozess vorzuziehen.

Im Gegensatz zu anderen Problemen, wie zum Beispiel der Entwicklung des Algorithmus von Kruskal[4], den die Schüler eigenständig (siehe beigefügte ABs) selbstständig entwickeln konnten, stellt der Algorithmus von Dijkstra Schüler und Lehrer vor eine Herausforderung: die menschliche Intuition ermöglicht uns – im Gegensatz zum Computer – relativ schnell Irrwege auszuschließen und den kürzesten Weg zu finden.

In der heutigen Stunde sollen die Schüler zunächst ein Problem selbst aufwerfen und dann zu diesem „eigenen“ Problem selbstständig den Algorithmus von Dijkstra entdecken, um so einen möglichst eigenständigen Lernprozess zu durchlaufen und Wissen möglichst nachhaltig zu festigen.

Der Problemaufwurf („Bestimmung eines kürzesten Weges“) kann noch leicht von den Schülern entwickelt werden, die eigenständige Entdeckung des Algorithmus stellt den Lehrenden hier jedoch vor eine große Herausforderung: Die von mir gesichtete Literatur bietet dazu wenig Anregung – der Algorithmus von Dijkstra wird in der Regel vermittelt, indem die zentrale Idee in einer Geschichte (Ameisen[5] oder chinesischen Kaiser[6]) vorgegeben wird.

Das Konzept der heutigen Stunde basiert grundlegend auf den Ausführungen von Prof. Dr. rer. nat. Peter Sanders und Dipl.-Inform. Johannes Singler im Algorithmus der Woche[7]. Sanders und Singler bieten mit dem Bindfadenverfahren eine Möglichkeit, den Algorithmus anschaulich und „begreifbar“ zu visualisieren.

Beim Bindfadenverfahren werden alle Kanten durch Fäden, alle Knoten durch Knoten (hier: Knetmasse) dargestellt. Es entsteht ein dreidimensionales Netz, das man anheben kann und so den kürzesten Weg zweier Punkte über die gespannte Schnur

[4] zur Generierung eines minimalen Spannbaums aus einem gegebenen Graphen
[5] Nach Gallenbacher, Jens: Abenteuer Informatik
[6] Nach http://mandalex.manderby.com/d/dijkstra.php?id=93
[7] http://www-i1.informatik.rwth-aachen.de/~algorithmus/algo7.php

zwischen diesen beiden Punkten ermitteln kann. Mit Hilfe dieses Verfahrens sollen die Schüler, den Algorithmus von Dijkstra anhand von Leitfragen selbst entdecken

Mit Hilfe gezielter Leitfragen und Arbeitsblätter sollen die Schüler dazu angeleitet werden, über die Vorgänge beim „Anheben" des Netzes nachzudenken und so in die Lage versetzt werden, die zentralen Ideen des Algorithmus von Dijkstra zu verbalisieren. Ziel der Gruppenarbeit kann dabei nicht sein, dass die Schüler den Algorithmus in Pseudocode formulieren, vielmehr sollen Sie in der Lage sein, ihn zu beschreiben.

Zum Beispiel so:

1. *Am Anfang: Entfernung vom Startknoten zu sich selbst: Null, alle anderen Knoten: unbekannte Entfernung: daher: Entfernung unendlich.*
2. *[Man hebt den Startknoten hoch und misst die Entfernung zu allen Nachbarn (und notiert diese in der Tabelle)].*
3. *Man hebt den liegenden Knoten mit der kürzesten Entfernung zum Startknoten hoch.*
4. *Dann betrachtet man alle liegenden Nachbarn von hängenden Knoten (und notiert deren Entfernung zum Startknoten in der Tabelle)…*
5. *… und nimmt den Nachbarn hoch, der die kürzeste Entfernung zum Startknoten hat.*
6. *Wenn noch Knoten liegen: Beginne wieder bei Schritt 3*

Die Entwicklung des Pseudocodes muss dann im Plenum erfolgen und erfordert ggf. stärker leitende Impulse seitens des Lehrers, insbesondere auch deshalb, da gerade der Sprung vom Beispiel und der verbalen Formulierung an dieser Stelle noch einmal ein hohes Abstraktionsvermögen voraussetzt und außerdem die Schüler nicht sehr erfahren im Formulieren von Pseudocode sind.

Sollte es gelingen, dass die Schüler mit Hilfe der Arbeitsblätter und Leitfragen den Algorithmus von Dijkstra verbalisieren können, so ist der Lernertrag der heutigen Stunde bereits sehr hoch. Die Schüler haben dann die grundlegende Idee des Algorithmus weitgehend selbstständig erarbeitet.

Ich bin mir jedoch auch dessen bewusst, dass dieser Schritt des selbstständigen Entdeckens ein Wagnis ist – nur, wenn die Arbeitsblätter und Leitfragen präzise und eindeutig formuliert werden und die Schüler sich exakt an die Vorgaben halten, kann der Erkenntnisprozess gelingen.

Gelingt es mir nicht, mit Hilfe der vorbereiteten Materialien, die Schüler zu einem eigenständigen Erkenntnisprozess anzuregen, so muss der Gruppenarbeit eine Plenumsphase folgen, in der durch (stark) leitende Impulse des Lehrers die Ideen der Schüler kanalisiert und so zum Algorithmus von Dijkstra gelenkt werden. Dies würde dann in etwa einer Vorgabe des Algorithmus durch eine Geschichte gleichkommen und ist daher keinesfalls negativ zu bewerten. Sollte der Erkenntnisprozess jedoch gelingen sollte, so haben die Schüler die zentrale Idee des Algorithmus selbst erarbeitet und damit im Sinne des konstruktivistischen Menschenbildes einen deutlich nachhaltigeren Lernprozess erlebt.

Die Frage, ob ein Lehrer dieses Risiko eingehen darf, ist sicherlich berechtigt, allerdings schätze ich das Risiko als sehr gering ein und verfahre daher nach dem Motto „Einen (gut überlegten) Versuch ist es wert!".

Verwendete Literatur:

Gallenbacher, Jens: **Abenteuer Informatik – IT zum Anfassen von Routenplaner bis Online-Banking**. (Kapitel 1). Spektrum Akademischer Verlag. München. 2007

Sanders, Peter; Singler, Johannes; Kupfererz, Stephan: **7. Algorithmus der Woche: Kürzeste Wege (Wie komme ich am schnellsten von einem Ort zu einem anderen?).**

http://www-i1.informatik.rwth-aachen.de/~algorithmus/algo7.php. (Stand 27.08.2007)

Tobias Stamm: **Mandalex**.

http://mandalex.manderby.com/d/dijkstra.php (Stand 27.08.2007)

Landesinstitut für Schule und Weiterbildung: **Gymnasiale Oberstufe – Informatik, Lehrplanentwurf**. Frechen: Ritterbach Verlag , 1998

NAME: Renate Thies LERNGRUPPE: 13 IF GK	DATUM: 04.07.09 ZEIT: 8.10- 9.45Uhr	FACHLEHRER/in: Herr Pelkmann	STUNDENTHEMA: „Viele Wege führen nach Rom – aber welcher ist der kürzeste?“ – Auf der Suche nach dem Algorithmus von Dijkstra

STUNDENZIEL (DOPPELSTUNDE):

Die Schüler sollen ausgehend von einem „realen“ Problem für das graphentheoretische Problem „Ermittlung kürzester Wege“ sensibilisiert und motiviert werden und erkennen, dass das reale Problem „Ermittlung kürzester Wege“ graphentheoretisch modelliert und dann gelöst werden kann. Weiter sollen Sie den Algorithmus von Dijkstra (möglichst) selbstständig entdecken und verbalisieren können. *(Minimales Stundenziel)* Die Schüler sollen den Algorithmus von Dijkstra im Hinblick auf die Implementierung in Pseudocode formulieren und auf verschiedenen Graphen durchführen können. *(Mögliches erweitertes Stundenziel)*

PHASEN	INHALTLICHE SCHWERPUNKTE / OPERATIONEN	SOZIAL- / AKTIONS-FORMEN	MEDIEN	ANMERKUNGEN ZUM LERNPROZESS
Begrüßung	Begrüßung und kurze Vorstellung der Gäste	LV		
Einleitung und Motivation	– L zeigt Bild von Navigationssystem, S. nennen dessen Funktionen „Kürzesten Weg von A nach B bestimmen“ – Reduktion des Problems durch L auf begrenzten Kartenausschnitt – Erarbeitung des Zusammenhangs zwischen Straßenkarte, Problemstellung und aktueller Unterrichtsreihe: – Ziel dieser Phase: S erkennen: Straßennetz kann als Graph modelliert werden; es müssten alle Orte als Knoten und Straßen als Kanten modelliert werden. – L gibt reduzierten Graphen vor – S formulieren Problem: => Gesucht: kürzester Weg auf Graph	LV/UG	Beamer, Präsentation	Lebensweltbezug des Themas der heutigen Stunde ergibt sich aus der gerade beendeten Studienreise des Kurses und erhöht somit hoffentlich die Motivation der Schüler. Die konkrete Abbildung des Straßennetzes als Graph erfolgt im Plenum und wird nach Vorgabe der Idee vom L vorgegeben, da die S bereits Graphen modellieren können und dieser Lerngegenstand somit nicht mehr im Zentrum der Stunde steht. Beschränkung auf Kartenausschnitt => Graph mit wenigen Knoten, zwecks didaktischer. Reduktion.
1. Phase der Erarbeitung	– Einleitung der GA: AA: S sollen kürzesten Weg zwischen Immstadt und Oppenheim bestimmen – Erweiterter AA: S sollen ihr Vorgehen rekapitulieren und beschreiben	GA	AB 1 und Anlage	Zunächst sollen die Schüler ohne Hinweise das Problem lösen (=> Förderung von Problemlösestrategien). Ziel ist dabei nicht das korrekte Ergebnis, sondern dass die S sich mit dem Thema befassen und so das Problem durchdringen Erweiterter AA zunächst verschlossen, damit S. zunächst „frei“ eine Lösung suchen, bevor Sie ihr eigenes Vorgehen rekapitulieren
Zwischen-präsentation und erste	– Eine/Einige Gruppen präsentieren (den kürzesten Weg und) ihr Vorgehen zur Ermittlung des kürzesten Weges – Thematisieren: Schwierigkeiten: viele Möglichkeiten, Auswertung aller Wege -> Hohe Anzahl, etc.	SV/UG	Beamer ggf. Tafel	Ich erwarte, dass die S. ihr Vorgehen nicht algorithmisch bzw. systematisch beschreiben können. Mögliche Schülerantworten könnten sein: alle Wege abschreiten, Distanzen vergleichen, längere Wege verwerfen. Ziel: Erkenntnis: Vorgehen für große Karten bzw. Distanzen (z.B. GT-Rom) unbrauchbar

Überleitung zur 2. Phase der Erarbeitung	– Im UG werden erste Ideen gesammelt, welche Kriterien an eine gute Strategie zur Ermittlung kürzester Wege (auf Graphen) gestellt werden: – L gibt Hinweis auf bekannten Algorithmus/Strategie – L weist auf nötige Sorgfalt in der GA-Phase hin.	UG	Tafel	S. sollen für die Notwendigkeit eines Algorithmus sensibilisiert werden, und Kriterien diskutieren, die dieser Algorithmus erfüllen muss (alle Möglichkeiten überprüfen, ggf. weitere)
2. Phase der Erarbeitung	– S erhalten Arbeitsmaterial und erarbeiten/spielen damit selbstständig den Algorithmus – AB IV muss nicht von allen Gruppen vollständig ausgefüllt werden – wenn die S nach einigen Zeilen das Prinzip verstanden haben, kann die GA ggf. beendet werden.	GA	AB II AB III AB IV , ggf. AB V	S erhalten Arbeitsmaterial, das Sie zum Algorithmus von Dijkstra leitet (siehe auch meth.-did. Kommentar). Falls nicht alle Gruppen AB II erfolgreich bearbeiten können, wird die GA kurz unterbrochen und AB II besprochen, um sicherzustellen, dass alle S. das Ziel dieses ABs erreicht haben. (Unterbrechung optional, da der L aufgrund der kleinen Lerngruppe den Leistungsfortschritt der S. beobachten kann. Im Idealfall kommt es nicht zu dieser Unterbrechung.
Sicherung	– S stellen Ergebnisse der GA vor, dabei wird… – … der Algorithmus an einem Graphen von L/S gemeinsam im Plenum vorgeführt und verbal erläutert – Erste Ideen/Ansätze zu Dijkstra werden notiert.	SV, UG Ggf. LV	Graph/Netz, Folie, ggf. Tafel	Erwartete Schülerlösung: siehe did. Kommentar

1. Mögliches Stundenende

HAUSAUFGABE ZUR STUNDE: keine

HAUSAUFGABE ZUR NÄCHSTEN STUNDE:

Die Schüler sollen den im Unterricht verbalisierten Algorithmus zusammenfassend aufschreiben.	Wird nur erteilt, wenn dies im Unterricht noch nicht geschehen ist. Sollte der Algorithmus schon im Unterricht notiert worden sein, so ist eine Vergabe von HA zur Sicherung ist an dieser Stelle nicht mehr von Nöten. Eine Hausaufgabe zu Übungszwecken böte sich vielleicht an, allerdings besteht in diesem Stadium der Algorithmus-Entwicklung die Gefahr, dass nicht alle S den Algorithmus zuhause alleine eigenständig durchführen können, insbesondere durch die fehlende Visualisierung. Weiterführende Aufgaben könnten die S. ebenfalls überfordern, daher wird in diesem Fall auf eine Hausaufgabe verzichtet.

2. Mögliches Stundenende (erwünschtes Ende)

3. Phase der Erarbeitung und Sicherung	– Der Algorithmus von Dijkstra wird in Pseudocode formuliert – Dabei: ggf. leitende Impulse und Vorgaben durch den L notwendig.	SV oder UG	Tafel	In Abhängigkeit von den Vorleistungen der Schüler ist an dieser Stelle ein geleitetes UG oder aber auch ein SV notwendig. Aufgrund seiner hohen Komplexität wird nicht erwartet, dass die S. den Algorithmus vollständig selbst entwickeln. Daher können ggf. in dieser Phase die Ideen und Beobachtungen der S. zusammengeführt und mit Hilfe des L zu einem Algorithmus formuliert werden. Ggf. ist an dieser Stelle ein stärker vom L geleitetes UG notwendig, um im Hinblick auf die folgende Implementierung den Algorithmus in Pseudocode zu formulieren.
Sicherung in Form einer Übung	– S. vollziehen den Algorithmus von Dijkstra an einem weiteren Graphen nach	EA/PA	AB	Idealerweise findet die Sicherung noch in der Stunde statt, dies gibt den S. die Gelegenheit, Fragen und Unsicherheiten im Gespräch mit S. und L zu klären. Eventuell kann dies aus zeitlichen Gründen nicht mehr erfolgen. Dann darf diese Phase nicht erst in der nächsten Woche erfolgen, sondern entfällt in die Hausaufgabe.
HAUSAUFGABE ZUR STUNDE: keine				
HAUSAUFGABE ZUR NÄCHSTEN STUNDE: Ggf.: Führen Sie den Algorithmus von Dijkstra an einem weiteren Graphen (siehe AB) durch.	Falls die HA gestellt wird: L. gibt Hinweis, dass diese HA nicht einer Prüfung sondern eher einer Festigung des Gelernten dient. D.h. die S sollen wissen: es nicht schlimm, wenn die HA nicht erfolgreich gelöst werden kann, allerdings sollen die S. dann ihre Fragen und Probleme benennen können.			

Graphen
Viele Wege führen nach Rom – aber welcher ist der kürzeste? (AB I)

Aufgabe 1

a) Finden Sie *gemeinsam* in der Gruppe in dem gegebenen Graphen den kürzesten Weg von Imstadt nach Oppenheim.
b) Wenn Sie den kürzesten Weg gefunden haben, dann öffnen Sie den unteren Teil dieses Arbeitsblattes. Dort finden Sie den zweiten Arbeitsauftrag!

Aufgabe 2

Der kürzeste Weg ist 123km lang. Haben Sie ihn gefunden?

a) Notieren Sie stichpunktartig, wie Sie vorgegangen sind, um den kürzesten Weg zu ermitteln!

_ _

_ _

_ _

_ _

_ _

_ _

_ _

b) Diskutieren Sie in der Gruppe folgende Fragen und notieren Sie eine *kurze* Antwort:

   a. Inwiefern gab es bei der Lösung der Aufgabe Schwierigkeiten?

_ _

_ _

_ _

   b. Warum waren Sie sicher, dass Ihr Ergebnis wirklich der kürzeste Weg war?

_ _

_ _

_ _

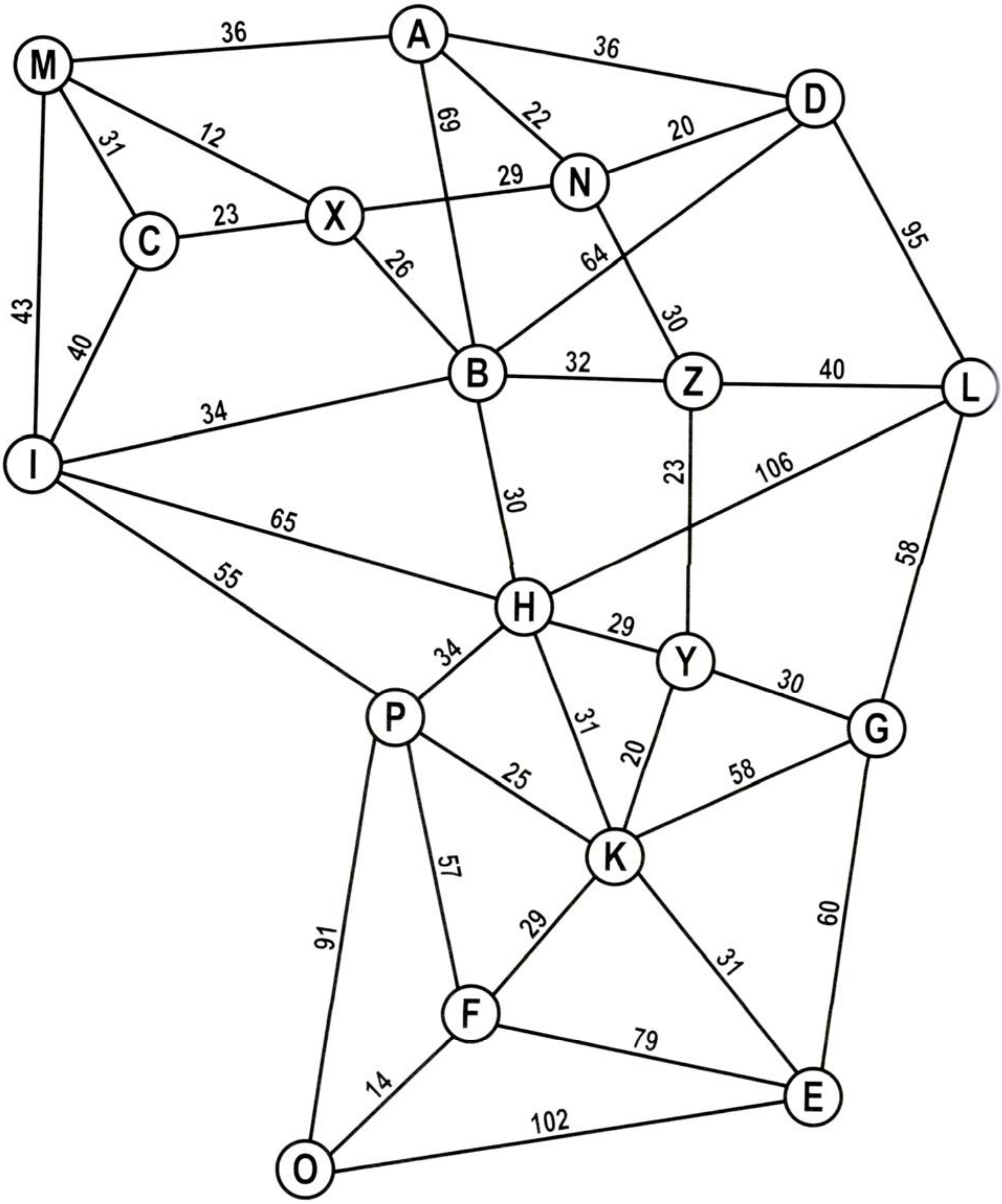
   c. Kann Ihr Vorgehen auf einen größeren Graphen (z.B. das deutsche Autobahnnetz) angewendet werden? (Begründen Sie Ihre Antwort!)

_ _

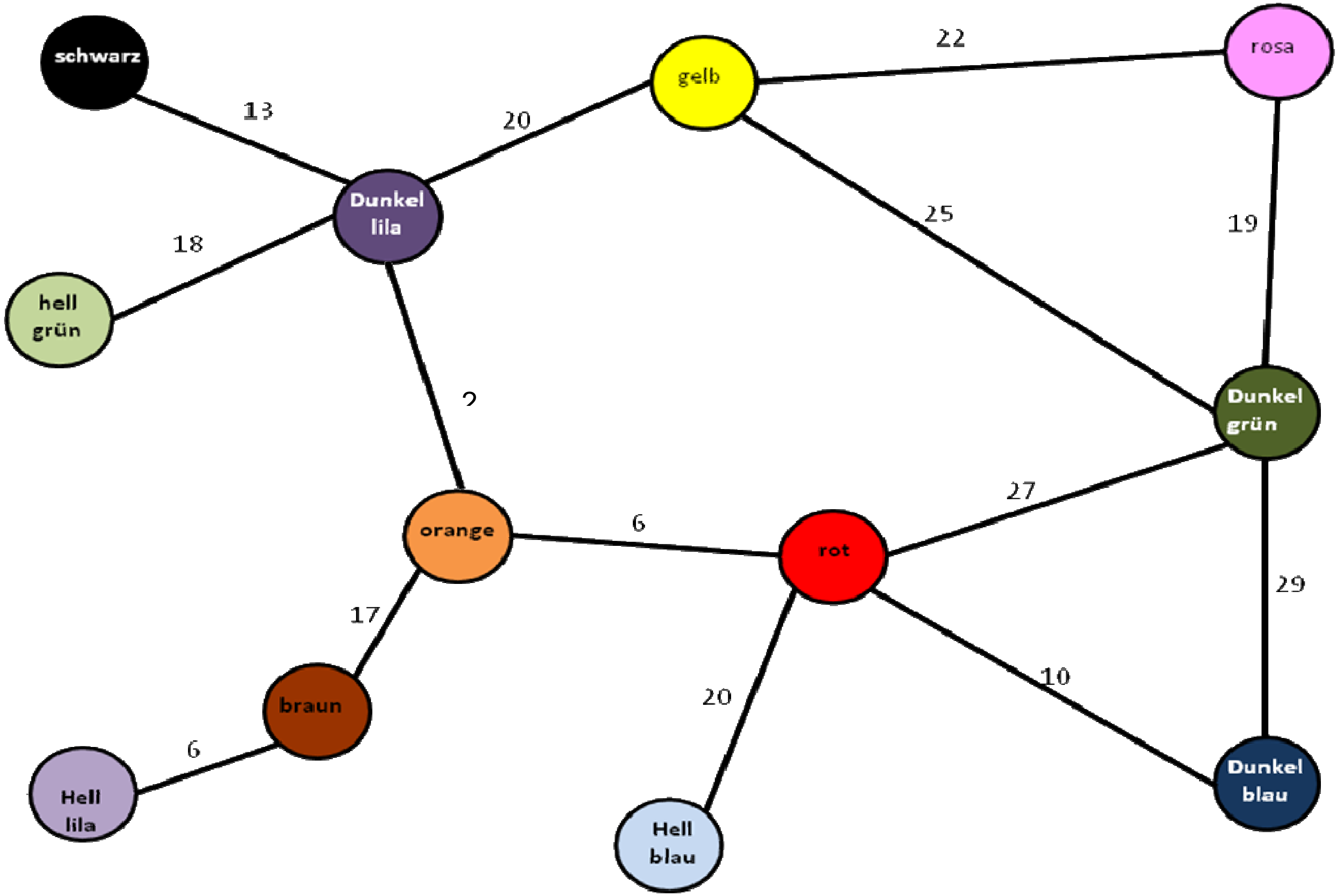
_ _

_ _

Graphen
Anlage zu AB I



Graphen
Anlage zu AB II-IV



Achtung: Diese Darstellung ist *nicht* maßstabsgetreu!!

Graphen
Viele Wege führen nach Rom – aber welcher ist der kürzeste? Auf der Suche nach dem Algorithmus von Dijkstra (AB II)

Hinweis:

Damit Ihre Gruppenarbeit erfolgreich verläuft und Sie mit Hilfe des folgenden Arbeitsauftrags erfolgreich den Algorithmus von Dijkstra „entdecken" können, ist es absolut notwendig, dass Sie die folgenden Arbeitsanweisungen <u>genau lesen und präzise befolgen</u>!

Gruppenarbeitsauftrag

1. Bilden Sie den Graphen mit Hilfe der Schnur und der Knetmasse *maßstabsgetreu* nach!
    a. Knoten werden durch Knete-Kugeln dargestellt.
       Jeder Knoten wird durch eine farbige Kugel dargestellt.
       Damit wir die Gruppenarbeit nachher besser auswerten können, ist es sinnvoll, wenn Sie sich an die Farben halten, so wie Sie auf dem Papier dargestellt sind.
    b. Kanten werden durch Schnüre dargestellt.
    c. Arbeiten Sie die Schnüre sorgfältig in die Knete-Kugeln ein!
2. Breiten Sie den fertigen Graphen auf dem Tisch aus.
3. Achten Sie darauf, dass die Knete *nicht* am Tisch haftet.

Wenn Sie fertig gebastelt haben, beginnen Sie mit Aufgabe 1!

Aufgabe 1:

Wie können Sie nun mit Hilfe des „Netzes" den kürzesten Weg von zum Beispiel **dunkel Grün** nach **dunkel Lila** bestimmen?

Notieren Sie die Antwort stichpunktartig. Sie dient als Gedächtnisstütze für das folgende Unterrichtsgespräch!

_ _

_ _

_ _

Tipp 1: Ziehen Sie den Startknoten (dk. grün) langsam (Zeitlupe) senkrecht hoch!

Tipp 2: Über welche Schnüre gelangen Sie auf dem kürzesten Weg vom Startknoten (dk. Grün) zum Zielknoten (dk. Lila)?

Fertig? Dann geht es weiter mit Aufgabe 2

Graphen
Viele Wege führen nach Rom – aber welcher ist der kürzeste? Auf der Suche nach dem Algorithmus von Dijkstra (AB III)

Aufgabe 2

1. Bringen Sie den Graphen wieder in die Ausgangsposition (ausgebreitet auf den Tisch legen). Achten Sie erneut darauf, dass die Knoten nicht am Tisch festkleben.
2. Ziehen Sie den Startknoten *in Zeitlupe* senkrecht hoch und…
3. … beobachten Sie dabei genau, in welcher Reihenfolge sich die Knoten heben!

   (Für Ihre Beobachtungen ist es nur wichtig, wann ein Knoten von der Tischplatte abhebt. Es *nicht* wichtig, wann er sich bewegt.)

Wiederholen Sie die Punkte 1-3 ruhig einige Male.

Erkennen Sie ein Kriterium nach dem man entscheiden (voraussagen) kann, welcher Knoten sich als nächstes hebt?

_ _

_ _

_ _

_ _

Tipp 1: Betrachten Sie die Entfernung der liegenden Knoten zum Startknoten.

Tipp 2: Wie groß ist die Entfernung des nächsten „abhebenden“ Knoten zum Startknoten? Vergleichen Sie diese Entfernung mit denen der anderen liegenden Knoten zum Startknoten.

Graphen
Viele Wege führen nach Rom – aber welcher ist der kürzeste? Auf der Suche nach dem Algorithmus von Dijkstra (AB IV)

Aufgabe 3

1. Bringen Sie den Graphen wieder in die Ausgangsposition (ausgebreitet auf den Tisch legen). Achten Sie erneut darauf, dass die Knoten nicht am Tisch festkleben.
2. Ziehen Sie den Startknoten *in Zeitlupe* senkrecht hoch und…
3. … füllen Sie gleichzeitig die Entfernungstabelle aus.

   Lesen Sie die Ausfüllanleitung sorgfältig durch und beachten Sie diese unbedingt!!

Tabelle über die kürzeste bekannte Entfernung zum Startknoten

Anzahl hängender Knoten	*Rot*	*Gelb*	*Dk. Blau*	*Hell Blau*	*Dk. Grün*	*Hell Grün*	*Rosa*	*Braun*	*Dk. Lila*	*Hell Lila*	*Schwarz*	*Orange*
0	∞	∞	∞	∞	*0*	∞	∞	∞	∞	∞	∞	∞
1												
2												
3												
4												
5												
6												
7												
8												
9												
10												
11												
12												

Ausfüll-Anleitung

Stellen Sie sich nun vor, Sie reisen durch den Graphen. Die Tabelle nutzen Sie, um sich alle wichtigen Informationen zu notieren:

Für jeden Knoten jeweils die kürzeste bekannte Strecke zum Startknoten.

Allerdings können Sie nicht *direkt* alle Entfernungen zu den anderen Knoten sehen.

- Sie beginnen Ihre Reise im „Startknoten“. Zu Beginn wissen Sie dann, dass Sie die Entfernung Null zum Startknoten haben (sie sind ja schon dort). Die Wege zu den anderen Knoten kennen Sie noch nicht. Sie gehen vom schlimmsten Fall aus: sie sind nicht erreichbar, die Entfernung ist also unendlich. (schon eingetragen: ∞)

- *Jedesmal, wenn sich ein neuer Knoten hebt, erreichen Sie diesen Knoten.*
- *Sobald Sie einen neuen Knoten erreichen, aktualisieren Sie ihre Tabelle mit den neu gewonnen Informationen:*
- In jedem Knoten steht ein Wegweiser, der Sie über die Richtungen und *Entfernungen* (Kanten + Gewichte) der Nachbarknoten informiert.
- Haben Sie einen Knoten erreicht, so kreisen Sie dessen letzte eingetragene Entfernung. ein und füllen diese Spalte nicht mehr weiter aus.

Graphen
Viele Wege führen nach Rom – aber welcher ist der kürzeste? Auf der Suche nach dem Algorithmus von Dijkstra (AB V)

Für besonders schnelle Gruppen:

Können Sie umgangssprachlich einen Algorithmus entwickeln, der ihr Vorgehen beschreibt?

Wenn Sie sich noch an die Struktogramme erinnern, so verwenden Sie diese…

Vielleicht hilft es Ihnen, wenn Sie die folgenden Fragen beantworten:

- Wie kann man aus der Tabelle erkennen, welcher Knoten sich als nächstes heben wird?

  Betrachten Sie die Tabelle zeilenweise.
- Wie kann man die Abstände zum Startknoten geschickt berechnen?

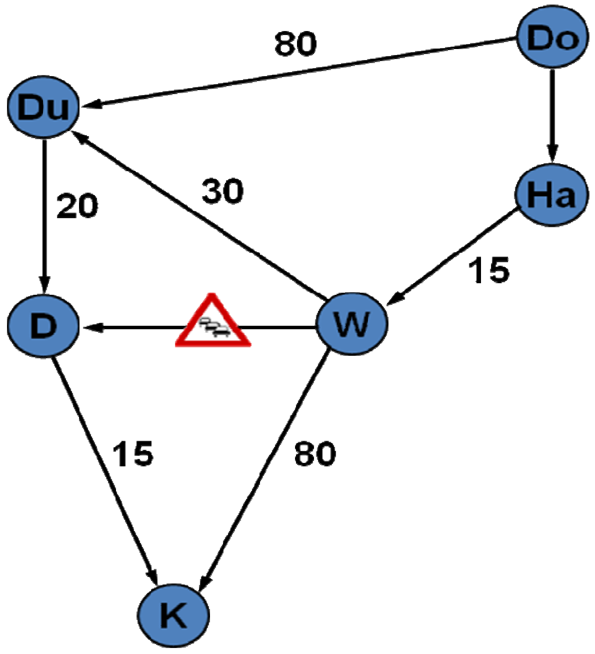
Das Grundgerüst des Pseudocodes (MUSTERLÖSUNG – nicht auf Schüler-Arbeitsblatt!):

```
Funktion kürzesteWege(Graph)
    Alle Knoten liegen,
    alle Entf() sind unendl,
    nur die Entf(Startknoten) = 0
    While es liegende Knoten gibt do
        Nenne den liegenden Knoten mit kleinster Entf() v
        Mache v hängend
        For all Fäden von v zu allen Nachbarn u Länge l do
            If    Entf(v)+L < Entf(u)
            then  Entf(u) = Entf(v)+L
        EndFor
    EndWhile
End
```

Entfernung vom Start zu u über Knoten v ist kürzer, als die bisher bekannte, dann: speichere die neue (kürzere) Entfernung!

Graphen
Viele Wege führen nach Rom – aber welcher ist der kürzeste?
Der Algorithmus von Dijkstra
(AB VI)

Unten sehen Sie den Graph der die Autobahnverbindungen zwischen 6 Städten darstellen soll. Die Entfernungen sind jeweils in Kilometer angegeben.



Legende:

= Stau = 150km

Do = Dortmund
Du = Duisburg
Ha = Hagen
W = Wuppertal
D = Düsseldorf
K = Köln

Aufgabe:
Führen Sie den Algorithmus von Dijkstra für den gegebenen Graphen aus und bestimmen Sie so den kürzesten Weg von Dortmund nach Köln

(Lehrerexemplar)

1. *Am Anfang: Entfernung vom Startknoten zu sich selbst: Null, alle anderen Knoten: unbekannte Entfernung: daher: Entfernung unendlich.*
2. *[Man hebt den Startknoten hoch und misst die Entfernung zu allen Nachbarn (und notiert diese in der Tabelle)].*
3. *Man hebt den liegenden Knoten mit der kürzesten Entfernung zum Startknoten hoch.*
4. *Dann betrachtet man alle liegenden Nachbarn von hängenden Knoten (und notiert deren Entfernung zum Startknoten in der Tabelle)…*
5. *… und nimmt den Nachbarn hoch, der die kürzeste Entfernung zum Startknoten hat.*
6. *Wenn noch Knoten liegen: Beginne wieder bei Schritt 3*

(FOLIE)

Anzahl hängender Knoten	*Rot*	*Gelb*	*Dk. Blau*	*Hell Blau*	*Dk. Grün*	*Hell Grün*	*Rosa*	*Braun*	*Dk. Lila*	*Hell Lila*	*Schwarz*	*Orange*
0	∞	∞	∞	∞	*0*	∞	∞	∞	∞	∞	∞	∞
1												
2												
3												
4												
5												
6												
7												
8												
9												
10												
11												
12												

