

Übungsblatt 5

A1: Kruskal und Prim

Im mehrfach angesprochenen *Mathematik-Didaktik-Buch Kombinatorische Optimierung erleben*¹ [Hus07, S.29/30]. werden u.a. die Algorithmen von *Kruskal* und *Prim* beschrieben. Dabei wird nicht nur ein Problem modelliert und anschliessend eine Idee für einen Algorithmus vorgestellt, sondern auch eine Pseudo-Programmiersprachen-Version der Algorithmen vorgestellt:

Der erste Algorithmus „malt“ immer die nächstbilligste Kante an, sofern dabei keine Kreise aus markierten Kanten entstehen. Im Verlauf des Algorithmus können dabei unzusammenhängende Teilgraphen entstehen. Hier bietet es sich wieder an, Daumenkinos zu basteln, um den Ablauf des Algorithmus zu visualisieren.

Der Algorithmus von Kruskal

- Eingabe: ein gewichteter zusammenhängender Graph mit n Knoten
- Ausgabe: ein minimaler aufspannender Baum des Graphen
- 1. Wähle eine *billigste* noch unmarkierte Kante.
 2. Markiere sie, falls sie keinen Kreis mit anderen markierten Kanten schließt.
 3. Stopp, falls $n-1$ Kanten markiert sind, anderenfalls gehe zu 1.

Der andere Algorithmus lässt einen Baum möglichst kostengünstig wachsen. Das „Produkt“ des Algorithmus ist zu jedem Zeitpunkt zusammenhängend.

Der Algorithmus von Prim

- Eingabe: ein gewichteter zusammenhängender Graph
- Ausgabe: ein minimaler aufspannender Baum des Graphen
- 1. Wähle einen Startknoten.
 2. Markiere die *billigste* vom bereits konstruierten Baum ausgehende Kante, falls sie keinen Kreis schließt.
 3. Stopp, falls $n-1$ Kanten markiert sind, anderenfalls gehe zu 2.

Es ist nicht ganz einfach, diese Algorithmen in so wenigen Schritten darzustellen. Ein erster Formulierungsversuch wird meist sehr viel mehr Schritte brauchen als die Endfassung. Der Trick bei der obigen Formulierung des Algorithmus von Prim besteht darin, den Startknoten bereits als Baum zu betrachten. Somit muss man nicht mehr zwischen der Wahl der ersten Kante und der Wahl der weiteren Kanten unterscheiden.

- Diese so einfach dargestellten Algorithmen sollen sich nicht in ein schülergerechtes Programm übertragen lassen?
Man versuche sich doch noch einmal an einem dieser Algorithmen!
(Vorschlag: Aus- und Eingabe als Matrizen)
- Es wurde zu Recht bemängelt, dass ohne grafische Veranschaulichung der Graphen bzw. Spannbäume diese Programme für Schüler doch recht unverständlich bleiben können. Es wäre also im Sinne eines *Werkzeugkastens* sinnvoll, eine derartige Matrix auf dem Bildschirm graphisch darzustellen.
Alternativ zur Realisierung eines der obigen Algorithmen in einem Programm kann auch ein derartiges *Tool* erzeugt werden.
- Wir haben teilweise über die schulische Zuordnung dieses Gebietes gesprochen. Wo gehört dieses Thema nun hin? In die Mathematik oder in die Informatik? Oder in beide Gebiete?
Wer hat denn dann welche Teilaspekte zu behandeln.

¹http://ls11-www.cs.uni-dortmund.de/people/paster/lehre/ddi/buecher/hussmann_Lutz-westphal_kombinatorische_optimierung_erleben/kapitel2.pdf

Man sehe sich dazu ergänzend auch das Inhaltsverzeichnis des Buches von Barth² [Bar03] aus derselben Buchserie für *Mathematik-Didaktik* an! Spannend wird es, wenn wir diese *Mathematik*-Bücher mit dem *Informatik*-Buch von Gallenbacher³ [Gal08] vergleichen.

- Kennen Sie Studenten, die im Studium des Lehramtes für Mathematik sich mit diesen Themen beschäftigt haben? Fragen Sie diese, was sie von dieser Thematik halten und ob sie beabsichtigen, diese Thematik in den Schulunterricht zu integrieren.

A2: Unterrichtsmethoden

Völlig unstrittig ist, dass das Thema *Hard- und Softwaresysteme* ein Thema der Informatik ist (Stichwort: von-Neumann-Architektur). Auch wenn das Zentralabitur dieses Thema derzeit völlig ignoriert, ist es (doch hoffentlich) nicht (endgültig) aus dem Schulkanon verschwunden. Überlegen sie sich eine Unterrichtssequenz (also nicht nur eine Stunde!), mit der sie den Schülern diese Thematik nahebringen wollen.

Ob sie dafür ein *Lernprogramm*, ein *Leitprogramm*, ein *Gruppenpuzzle* oder andere kooperative Lernformen oder ein *Projekt* verwenden wollen, ist Ihnen überlassen. Stellen Sie Ihr Konzept skizzenhaft dar.

A3: Projektunterricht

Es ist nur wenige Jahre her, da war *Projektunterricht* in der pädagogischen Diskussion das, was heute *kooperatives Lernen* ist.

Meine These (oder besser nur Hypothese):

Mit der Einführung des Zentralabiturs ist der Projektunterricht in der Schule tot!

- Nehmen Sie auf den Vorlesungsinhalten aufbauend aufbauend zur obig formulierten These Stellung!
- Lassen sich evtl. einzelne Elemente des Projektunterrichtes mit evtl. einzelnen Elementen des kooperativen Unterrichtes retten?
- Versuchen Sie, ihre Position an einem Beispiel zu konkretisieren!

Literatur

- [Bar03] BARTH, Armin P.: *Algorithmik für Einsteiger : für Studierende, Lehrer und Schüler in den Fächern Mathematik und Informatik*. 1. Aufl. Braunschweig [u.a.] : Vieweg, 2003. – ISBN 3-528-03196-4
- [Gal08] GALLENBACHER, Jens: *Abenteuer Informatik : IT zum Anfassen - von Routenplaner bis Online-Banking*. 2. Aufl. Heidelberg : Spektrum Akad. Verl., 2008 (Sachbuch). – Enth. S.107-119: Über 10 000 000 Jahre Informatik. – ISBN 978-3-8274-1926-2
- [Hus07] HUSSMANN, Stephan [. (Hrsg.): *Kombinatorische Optimierung erleben : in Studium und Unterricht*. 1. Aufl. Wiesbaden : Vieweg, 2007 (Mathematik erleben Mathematik für das Lehramt). – ISBN 3-528-03216-2 ; 978-3-528-03216-6

Achtung: Abgabe bis: 2. Juni 2010, 13:00 Uhr

Die nächste Übung am 26. Mai findet wie angekündigt im Schullabor statt!

²http://ls11-www.cs.uni-dortmund.de/people/paster/lehre/ddi/buecher/barth_algorithmik_fuer_einsteiger.pdf

³http://ls11-www.cs.uni-dortmund.de/people/paster/lehre/ddi/buecher/abenteuer_informatik_inhalt.pdf