Otto-von-Guericke-Universität Magdeburg
Institut für Mathematische Optimierung
Prof. Dr. Robert Weismantel
Dr. Matthias Köppe

Sommersemester 2003

2. Übungsblatt

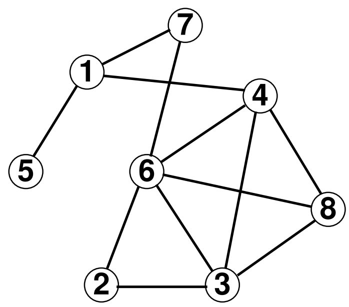
Computerorientierte Mathematik

`http://www.math.uni-magdeburg.de/~mkoeppe/lehre/coma-2003`

Abgabe: Donnerstag, 17. April, zu Beginn der Übung

6. Aufgabe (Depth-First-Search) **3 Punkte**

Gegeben sei der folgende Graph $G = (V, E)$, der durch die nebenstehende Adjazenzliste beschrieben ist.



1	$\mapsto$	4, 5, 7
2	$\mapsto$	3, 6
3	$\mapsto$	2, 4, 6, 8
4	$\mapsto$	1, 3, 6, 8
5	$\mapsto$	1
6	$\mapsto$	2, 3, 4, 7, 8
7	$\mapsto$	1, 6
8	$\mapsto$	3, 4, 6

In G soll ein aufspannender Baum T bestimmt werden. Dazu verwenden wir den in der Vorlesung vorgestellten Depth-First-Search-Algorithmus. Wir nehmen zusätzlich an, daß die Auswahl des Knotens, der als nächstes zu bearbeiten ist, streng nach der durch die Adjazenzliste vorgegebenen Ordnung erfolgt.

Entscheiden Sie, welchen aufspannenden Baum T der Algorithmus unter dieser Annahme konstruiert? Überlegen Sie sich dazu einen Ablaufplan, in dem der Algorithmus den Graphen G durchsucht (Vorsicht Rekursion!) und der die markierten und unmarkierten Knoten berücksichtigt. Begründen Sie (kurz) ihre Auswahl.

7. Aufgabe **3 Punkte**

Modifizieren Sie den Depth-First-Search-Algorithmus, so daß zusätzlich die Anzahl der Komponenten bestimmt wird, aus denen der gegebene Graph besteht und geben Sie den modifizierten Algorithmus ebenfalls in Pseudocode an. Begründen Sie kurz, warum ihr Algorithmus das Richtige berechnet.

8. Aufgabe (Greedy-Max) **4 Punkte**

Wir betrachten den folgenden Algorithmus, der als Greedy-Algorithmus bezeichnet wird („greedy“ heißt „gierig“).

Input: Ein einfacher Graph $G = (V, E)$ mit positiven Kantengewichten $c(e) > 0$ für alle Kanten $e \in E$.

Output: Ein Wald $W \subseteq E$ mit maximalen Gewicht $c(W) := \sum_{e \in W} c(e)$.

(1) Sei m die Anzahl der Kanten in G. Numeriere diese m Kanten, so daß gilt $c(e_1) \geq c(e_2) \geq \cdots \geq c(e_m) > 0$.

(2) **Setze** $W := \emptyset$.

(3) **For** $i = 1$ **to** m **do**

Falls $W \cup \{e_i\}$ keinen Kreis enthält, dann setze $W := W \cup \{e_i\}$.

(4) Gib die Menge W aus.

Beweisen Sie, daß der Algorithmus Greedy-Max immer einen Wald mit maximalem Gewicht konstruiert.

(**Hinweis:** Induktion über die Anzahl ausgewählter Kanten, d.h. die Kardinalität der Menge W. Dabei ist durch Induktion zu zeigen, daß, falls $e_{i_1}, \ldots, e_{i_k}$ die ersten k Kanten sind, die der Algorithmus Greedy-Max ausgewählt hat, immer ein maximaler Wald in G existiert, der diese k Kanten enthält).

9. Aufgabe **2.5+2.5 Punkte**

Wir betrachten noch einmal den vorgestellten Depth-First-Search-Algorithmus und nehmen zusätzlich an, daß der Graph $G = (V, E)$, den der Algorithmus als Input erhält, zusammenhängend ist. Zeigen Sie

(a) Die im Algorithmus konstruierte Menge T ist ein aufspannender Baum für G.

(b) Der Graph G enthält genau dann einen Kreis, wenn die konstruierte Menge $B \neq \emptyset$ ist.

10. Aufgabe **5 Punkte**

Gegeben sei ein (einfacher) Graph $G = (V, E)$ mit n Knoten und m Kanten, wobei $m > \binom{n-1}{2}$ sein soll. Zeigen Sie, daß unter diesen Bedingungen der Graph G zusammenhängend sein muß.