TU Berlin
Institut für Mathematik
Sekretariat MA 3-2
Dr. Brigitte Lutz-Westphal
Andreas Fest
Andrea Hoffkamp

www.math.tu-berlin.de/Vorlesungen/SS07/DiskrMaLS/

7. Übung zur Vorlesung
Diskrete Mathematik und ihre Anwendungen
Sommersemester 2007

Am Do, den 7. Juni findet zu Beginn der Vorlesung der zweite Kurztest statt.

Aufgabe 1 (4 Punkte)
(False coin problem)

Angenommen wir haben eine echte Münze, die wir mit 0 bezeichnen und $1, \dots, r$ andere Münzen, die bis auf ihre Bezeichnungen $1, \dots, r$ ununterscheidbar sind. Wir vermuten, dass eine der Münzen $1, \dots r$ falsch ist und zwar entweder zu leicht oder zu schwer im Vergleich zu den anderen Münzen.

a) Zeigen Sie, dass mindestens $\lceil \log_3(2r+1) \rceil$ Wägungen auf einer Balkenwaage nötig sind, um zu entscheiden, welche Münze (wenn es denn eine falsche gibt) falsch ist und um zu entscheiden, ob diese Münze zu schwer oder zu leicht ist.

b) Bescheiben Sie die Wägungsprozedur, die genau $\lceil \log_3(2r+1) \rceil$ Wägungen im Falle von $r = 4$ benutzt.

Aufgabe 2 (6 Punkte)
(Algorithmus von Borůvka)

In dieser Aufgabe lernen Sie einen weiteren Algorithmus zur Bestimmung eines minimal aufspannenden Baumes kennen.

Die Eingabe ist ein Graph $G = (V, E)$ mit Kantengewichtsfunktion w. Wir gehen davon aus, dass verschiedene Kanten verschiedene Gewichte erhalten, dass die Gewichtsfunktion also injektiv ist.

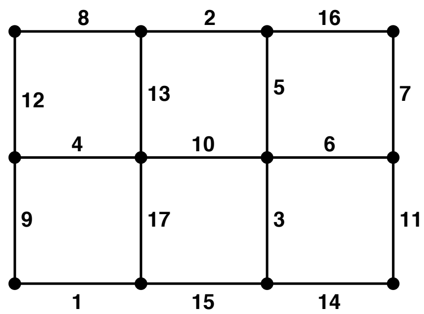
Der Algorithmus konstruiert Schritt für Schritt Kantenmengen $E_0, E_1, \ldots \subseteq E$, er beginnt mit $E_0 = \emptyset$. Angenommen, die Menge E_{i-1} ist schon berechnet; $(V_1, \ldots, V_t)$ sei die Partition der Knotenmenge, die durch die Komponenten des Graphen (V, E_{i-1}) gegeben ist. (Um ganz genau zu sein, müsste diese Partition auch noch einen Index i haben, weil sie in jedem Schritt eine andere ist. Wir lassen diesen Index hier um der Übersicht willen weg.)

Für jede Menge V_j von dieser Partition bestimmen wir die Kante $e_j = \{x_j, y_j\}$ (mit $x_j \in V_j$, $y_j \notin V_j$), deren Gewicht minimal ist unter allen Kanten der Form $\{x, y\}$, $x \in V_j$, $y \notin V_j$ (es kann passieren, dass $e_j = e_{j'}$ ist für $j \neq j'$). Wir setzen $E_i = E_{i-1} \cup \{e_1, \dots, e_t\}$. Der Algorithmus endet, wenn der Graph (V, E_i) eine Komponente hat.

a) Wenden Sie den oben beschriebenen Algorithmus auf den unten abgebildeten Graphen an. Zeichnen Sie für jeden Schritt des Algorithmus den Graphen nochmals neu und tragen Sie jeweils alle bis dahin ausgewählten Kanten ein (Wie bei einem Daumenkino).

b) Zeigen Sie, dass der Algorithmus von Borůvka keine Kreise erzeugt. (Tipp: Beachten Sie, dass die Gewichtsfunktion injektiv ist!)

c) Wenden Sie den Algorithmus von Prim auf den unten abgebildeten Graphen an. Lassen Sie den Algorithmus im linken oberen Knoten starten. Zeichnen Sie wiederum für jeden Schritt des Algorithmus den Graphen nochmals neu.



Abgabetermin: Am Do, den 7. Juni **vor Beginn** der Vorlesung.