**Fachbereich Mathematik**
Prof. Dr. M. Joswig
Katja Kulas

TECHNISCHE UNIVERSITÄT DARMSTADT

Sommersemester 2007
10.07.2007

# Diskrete Optimierung II

## 13. Übung

### Gruppenübungen

**Aufgabe G1** Berechnen Sie mithilfe des Algorithmus von Conti und Traverso das Minimum $\min\{c^T x | Ax = b, x \in \mathbb{N}^n\}$ für

(i)
$$A = \begin{pmatrix} 4 & 5 & 1 & 0 \\ 2 & 3 & 0 & 1 \end{pmatrix}, \quad b = \begin{pmatrix} 37 \\ 20 \end{pmatrix} \quad \text{und} \quad c = \begin{pmatrix} 11 \\ 15 \\ 0 \\ 0 \end{pmatrix}$$

(ii)
$$A = \begin{pmatrix} 5 & 1 & 6 \\ 1 & 3 & 4 \\ 3 & 2 & 1 \end{pmatrix}, \quad b = \begin{pmatrix} 35 \\ 21 \\ 12 \end{pmatrix} \quad \text{und} \quad c = \begin{pmatrix} 100 \\ 1 \\ 5 \end{pmatrix}.$$

**Aufgabe G2** Im Dijkstra-Algorithmus zum Bestimmen eines kürzesten $s$-$t$-Pfades in einem Graphen $G = (V, E)$ mit Kantengewichten $c : E \to \mathbb{R}_+$ müssen die noch zu untersuchenden Knoten in bestimmten Datenstrukturen verwaltet werden. Eine effektive Datenstruktur sind die *$d$-Heaps*. Dies sind Wurzelbäume mit Grad $d > 0$, d.h. es gibt einen ausgezeichneten Wurzelknoten und jeder Knoten des Baumes hat maximal $d$ Nachfolgerknoten, sogenannte *Kinder*. Jeder Knoten $i$ des $d$-Heaps hat einen bestimmten Wert $key(i)$ und die Werte seiner Kinder sind nicht kleiner als $key(i)$. Die *Tiefe* eines Knotens $i$ ist die Anzahl der Kanten von der Wurzel zu $i$ im Baum. Die Kanten des Baumes repräsentieren die Vorgänger-Nachfolger-Beziehung der entsprechenden Knoten. Dabei werden Knoten in aufsteigender Tiefe und innerhalb gleicher Tiefe von links nach rechts hinzugefügt, d.h. jede Tiefe muss aufgefüllt sein.

(i) Geben Sie die Laufzeiten folgender Operationen in einem $d$-Heap $H$ an:
   - *find_min($i, H$)*: Finde in $H$ einen Knoten $i$ mit minimalem Wert.
   - *insert($i, H$)*: Füge einen neuen Knoten $i$ in $H$ ein.
   - *decrease_key($value, i, H$)*: Reduziere den Wert von $i$ auf $value$.
   - *delete_min($i, H$)*: Lösche den Knoten $i$ mit minimalem Wert aus $H$.

   Betrachten Sie dafür die maximale Anzahl von Knoten in Tiefe $k$ und die maximale Tiefe eines $d$-Heaps bei $n$ Knoten.

(ii) Folgern Sie, dass der Dijkstra-Algorithmus mit von $d$-Heaps das Kürzeste-Wege-Problem in $O(nd \log_d n + m \log_d n)$ löst.

(iii) Zeigen Sie, dass der Dijkstra mit $d$-Heaps für dünne Graphen, d.h. $m = O(n)$, eine Laufzeit von $O(n \log n)$ hat.

(iv) Wie muss man $d$ wählen, dass der Dijkstra mit $d$-Heaps bestmögliche Laufzeit hat?

**Bemerkung:** Eine noch effizientere Datenstruktur stellen die *Fibonacci-Heaps* dar. Damit benötigt der Dijkstra-Algorithmus nur noch $O(m + n \log n)$.