Universität Konstanz
Lehrstuhl für Praktische Informatik
Dr. S. Cornelsen / C. Pich

Entwurf und Analyse von Algorithmen
WS 2006/2007

4. Übungsblatt

Ausgabe: 17.11.2006 **Abgabe:** 24.11.2006, 12 Uhr
Die Bearbeitung in Zweiergruppen ist ausdrücklich erwünscht.

Aufgabe 12: Graphen **6 Punkte**

Zeigen Sie, dass für einen Graphen $G = (V, E)$ folgende Aussagen äquivalent sind:

(a) G ist zusammenhängend und kreisfrei.

(b) Zwischen je zwei Knoten in G gibt es genau einen einfachen Weg.

(c) G ist zusammenhängend und hat $n - 1$ Kanten.

(d) G ist minimal zusammenhängend, d.h. G ist zusammenhängend und für alle $e \in E$ ist $G - e$ unzusammenhängend.

(e) G ist maximal kreisfrei (azyklisch), d.h. G ist kreisfrei und für alle $e \in \binom{V}{2} \setminus E$ enthält $G + e$ einen Kreis.

Aufgabe 13: Integer-Datenstruktur **6 Punkte**

Entwerfen Sie eine Datenstruktur, die die folgenden zwei Operationen für eine dynamische Menge S von Ganzzahlen unterstützt:

- $\textsc{Insert}(S, x)$ fügt x in S ein.
- $\textsc{DeleteLargerHalf}(S)$ löscht die größten $\lceil |S|/2 \rceil$ Elemente aus S.

Zeigen Sie, dass man diese Datenstruktur so impleme¡ntieren kann, dass die Laufzeit einer beliebigen Folge von n Operationen in $\mathcal{O}(n)$ ist. Sie können annehmen, dass die Menge zu Beginn leer ist.

[Bitte wenden]

Aufgabe 14: Heaps **8 Punkte**

Ein *Heap* ist ein Binärbaum, der, soweit es geht, vollständig aufgefüllt ist. Die unterste Ebene ist möglicherweise nicht vollständig, aber zumindest „linksbündig“aufgefüllt. Die Einträge in jedem Knoten erfüllen die *Heap-Bedingung*, dass für jeden Knoten der Eintrag kleiner als die Einträge seiner Kindknoten ist. Zwischen den Kindknoten braucht keine Relation eingehalten zu werden. Im Ergebnis ist stets der Eintrag mit dem kleinsten Wert an der Wurzel.

Ein Knoten wird eingefügt (INSERT), indem er zunächst in der untersten Ebene an die linkeste freie Stelle eingefügt und dann solange immer mit seinem Elternknoten vertauscht wird, wie dieser einen größeren Eintrag hat (ggf. bis hinauf zur Wurzel), so dass danach die Heap-Bedingung wiederhergestellt ist.

Die Wurzel wird gelöscht (DELETEMIN), indem sie entfernt und der rechteste Knoten der untersten Ebene an ihre Stelle gesetzt wird. Da dies i.A. die Heap-Bedingung verletzt, muss dieser Knoten nun, ähnlich dem Einfügen, nach unten „durchgetauscht“werden, bis die Heap-Bedingung wiederhergestellt ist, d.h. der Knoten wird immer mit einem Kindknoten vertauscht, dessen Eintrag kleiner ist (ggf. bis hinunter zu untersten Ebene).

(a) Fügen Sie in eine anfänglich leere Heap-Datenstruktur die folgenden Werte ein bzw. führen Sie die Operation aus: 6, 23, 14, 17, 13, 25, 2, DELETEMIN, 20, 21, 26, DELETE-MIN. Geben Sie den Heap nach jeder Operation an.

(b) Ein Heap wird meistens mit einem Array $a[1..n]$ realisiert, wobei $a[1]$ die Wurzel des Baums ist, $a[2]$ das linke Kind der Wurzel, $a[3]$ das rechte Kind der Wurzel, $a[4]$ das linke Kind von $a[2]$ usw. Wie lautet die Heap-Bedingung für ein Array?

(c) Wieviele Elemente hat ein Heap der Höhe h höchstens/mindestens?

(d) Zeigen Sie, dass ein n-elementiger Heap Höhe $\lfloor \log_2 n \rfloor$ hat.

(e) Zeigen Sie, dass es in einem n-elementigen Heap höchstens $\lceil \frac{n}{2^{h+1}} \rceil$ Elemente der Höhe h geben kann. Gebrauchen Sie dies, um zu zeigen, dass man einen Heap von n Elementen in Linearzeit aufbauen kann.