Technische Universität Berlin **SS 2013**
Fakultät II, Institut für Mathematik
Sekretariat MA 5–2, Frau Kiefer

Prof. Dr. Martin Skutella
Torsten Gellert
Martin Groß
Dr. Max Klimm

Aylin Acikel, Katharina Bütow, Christian Döblin,
Alexander Hopp, Daniel Kuske, Olivia Röhrig, Robert Rudow
Daniel Schmand, Hendrik Schrezenmaier, Mona Setje-Eilers
Judith Simon, Sebastian Spies, Fabian Wegscheider, Jan Zur

3. Übungsblatt Computerorientierte Mathematik II

Abgabe: 30.04.2013 (bis 14:15 Uhr in MA 004)

1. Aufgabe (1 + 2 Punkte)

(a) In Anwendungen kommt es häufig vor, dass ein Heap aus n Daten aufgebaut werden muss und diese Daten anschließend mit `pop()` extrahiert werden. Wie hoch ist der asymptotische Aufwand für den Aufbau und die n Extraktionen, wenn ihr den Aufbau und die Extraktionen wie in der Vorlesung beschrieben durchführt? Könnt ihr eine untere Komplexitäts-Schranke dafür angeben?

(b) Ihr habt ein Array der Länge $2^n - 1$ gegeben, von dem ihr wisst, dass es einen Max-Heap darstellt.

(i) Wieviele Elemente müsst ihr betrachten, um das minimale Element des Heaps zu finden?

(ii) Wieviele Elemente müsst ihr betrachten, um das zweitgrößte Element zu finden?

Begründet jeweils kurz eure Antwort.

2. Aufgabe (3 + 1 + 3 Punkte)

(a) Sei T ein beliebiger nicht-leerer Binärbaum mit n Knoten. Für $i = 0, 1, 2$ sei n_i die Anzahl der Knoten von T, die genau i Kinder haben. Beweist, dass für alle binären Bäume gilt: $2n_0 + n_1 = n + 1$.

(b) Sei T ein vollständiger, nicht-leerer Binärbaum mit $n \geq 2$ Knoten, d. h. alle Knoten sind entweder Blätter oder haben zwei Kinder. Zeigt, dass in T mindestens die Hälfte der Knoten Blätter sind.

(c) Bei einem nicht-leeren Binärbaum mit Wurzel r und n Knoten seien diese von 1 bis n so durchnummeriert, dass entlang jedes Pfades von der Wurzel zu einem Blatt diese Nummern streng monoton aufsteigen. Wir nennen einen solchen Baum auch *aufsteigend*. Beweist folgende Eigenschaft: Es gibt genau $n!$ verschiedene solcher Bäume auf n Knoten.

3. Aufgabe (1 + 4 + 2 Punkte)

Sei T ein binärer Baum mit paarweise verschiedenen Werten in den Knoten.

(a) Ihr habt folgende Preorder- und Inorder-Darstellungen eines Binärbaums gegeben:
Preorder: `D A E C H F B G I`
Inorder: `E A H C D F G B I`

Rekonstruiert daraus den ursprünglichen Baum.

(b) Beschreibt einen Algorithmus basierend auf eurem Vorgehen aus a) zur Rekonstruktion eines Baums aus Pre- und Inorder-Darstellung. Ist der rekonstruierte Baum eindeutig? Begründet kurz.

(c) Kann man den Baum auch dann eindeutig rekonstruieren, wenn

- die In- und Postorder-Darstellungen bekannt sind?
- die Pre- und Postorder-Darstellungen bekannt sind?

Gebt ein Gegenbeispiel an, wenn der Baum nicht eindeutig rekonstruierbar ist.

4. Aufgabe (3 Punkte)

Beweist, dass die Blätter eines Baumes in einem Inorder-Durchlauf, in der gleichen Reihenfolge besucht werden wie bei einem Preorder-Durchlauf.