Algorithmen und Programmierung 3, WS 2003/2004 — 5. Übungsblatt

Abgabe bis Donnerstag, 27. November 2003

29. (a) (0 Punkte) In einem *vollen* binären Baum hat jeder innere Knoten zwei Kinder. Man kann einen beliebigen Binärbaum als vollen Binärbaum darstellen, indem man für jedes fehlende Kind eines Knotens ein neues Blatt (einen *externen* Knoten) einsetzt. Alle ursprünglichen Knoten werden zu inneren Knoten, und die Blätter repräsentieren die `null`-Zeiger im ursprünglichen Baum.
    Beweisen Sie, dass in ein voller binärer Baum mit n Blättern $n-1$ innere Knoten hat.

    (b) (4 Punkte) Beweisen Sie, dass in einem vollen binären Baum mit n Blättern auf Tiefe $l_1, \ldots, l_n$ die folgende Gleichung gilt:

    $$\sum_{i=1}^{n} 2^{-l_i} = 1$$

    (c) (0 Punkte) Für jede Folge $l_1, \ldots, l_n$ ganzer Zahlen, die obige Gleichung erfüllt, gibt es einen vollen binären Baum mit n Blättern auf Tiefe $l_1, \ldots, l_n$.

30. (0 Punkte) In einem a-b-Baum speichert ein Knoten P mit k Kindern $k-1$ Schlüssel $v_1, \ldots, v_{k-1}$, wobei v_i der kleinste Schlüssel im $(i+1)$-ten Teilbaum ist. Beweisen Sie, dass jeder Schlüssel fast genau einmal gespeichert wird. (Mit welcher Ausnahme?)

31. (a) (0 Punkte) Nehmen wir an, dass jeder innere Knoten genau 2 Kinder hat. Was können Sie aus der vorigen Aufgabe über die Beziehung zwischen der Anzahl der inneren Knoten und der Anzahl der Blätter schließen?

    (b) (0 Punkte) Welche Beziehung besteht im allgemeinen Fall (bei beliebigen Knotengraden) zwischen der Anzahl der inneren Knoten, der Anzahl der Blätter, und der Summe der Grade (Anzahlen der Kinder) der inneren Knoten? Gilt diese Beziehung auch für Bäume, die keine a-b-Bäume sind, wo also die Blätter auf verschiedenen Ebenen sein können?
    Man kann die Aussage auf mehrere verschiedene Arten beweisen.

32. (0 Punkte) Konstruieren Sie für jede Höhe h einen 2-3-Baum mit Höhe h, bei dem man einen neues Element einfügen kann, sodass man den gesamten Weg zur Wurzel durchlaufen muss und sich die Höhe um 1 erhöht, und beim anschließenden Entfernen dieses Elementes der ursprünglich Zustand wiederhergestellt wird. (Bei 2-4-Bäumen kann das nicht passieren.)

33. (4 Punkte) Geben Sie ein Beispiel eines 2-3-Baums mit Höhe $h = 6$ an, bei dem man ein Element entfernen kann, sodass man mit dem Umbau des Baumes in Tiefe 4 aufhören kann, aber dennoch den gesamten Weg zur Wurzel durchlaufen muss, um die Schlüssel richtigzustellen. (Sie müssen nicht den gesamten Baum aufzeichnen, sondern nur die betroffenen Teile.)

    (Zusatzfrage, 0 Punkte.) Warum kann so etwas beim Einfügen nicht passieren?

34. (0 Punkte) Um die Schwierigkeit, die in der vorigen Aufgabe behandelt wurde, zu umgehen, kann man die vierte Bedingung bei der Definition von a-b-Bäumen abschwächen:

    > In einem a-b-Baum speichert ein innerer Knoten P mit k Kindern $k-1$ Schlüssel $v_1, \ldots, v_{k-1}$, wobei v_i größer als alle Schlüssel im i-ten Teilbaum und kleiner oder gleich allen Schlüsseln im $(i+1)$-ten Teilbaum ist.

    Wie muss man das Suchen, Einfügen, und Entfernen an diese veränderten Definition anpassen?

35. (12 Punkte) Implementieren Sie 2-3-Bäume zum Speichern von Zahlen in Java oder Haskell.