Universität Stuttgart
Fakultät 5, Institut FMI
Einf.i.d.Informatik II

Sommersemester 2003
Lewandowski, Claus
11. Juli 2003

Aufgabenblatt 11 **Abgabe: 18. Juli, 20.00 Uhr**

1. **Einfügen und Löschen in AVL-Bäumen** (leicht-mittel) (votieren, 3 Punkte)
Fügen Sie die Elemente „WOLF“, „GANG“, „SCH“, „MID“, „AN“, „DREAS“, „CHIA“, „REL“, „LO“ in einen anfangs leeren AVL-Baum in dieser Reihenfolge ein und löschen Sie danach die Elemente in der gleichen Reihenfolge aus dem AVL-Baum (betrachten Sie beim Löschen beide Varianten (sowohl Inorder-Vorgänger als auch Inorder-Nachfolger). Geben Sie jeweils die nötigen Rotationen an.

2. **Löschen in AVL-Bäumen** (mittel) (votieren, 2 Punkte)
Konstruieren Sie einen AVL-Baum, und geben Sie einen Knoten in Ihrem Baum an, der mehr als 1 Rotation beim Löschen dieses Knotens benötigt. Beachten Sie, dass eine LR- oder RL-Rotation als 1(!) Rotation gilt!

3. **Sortieren mit Such- und AVL-Bäumen** (mittel) (votieren, 3+2 Punkte)
Gegeben ist eine Folge $a_1, a_2, ..., a_n$ von paarweise verschiedenen natürlichen Zahlen. Um diese Zahlen zu sortieren, kann man wie folgt vorgehen:

   - man füge die n Zahlen nacheinander in einen Suchbaum ein
   - man gebe den Inhalt der Knoten in Inorder-Reihenfolge aus

   Wie viele Vergleiche zwischen Elementen der Folge benötigt dieses Sortierverfahren im schlechtesten, im besten Fall und im Mittel, wenn man

   a) einen binären Suchbaum verwendet.
   b) einen AVL-Baum verwendet. (Aufwand im Mittel hier nicht nötig)

   Geben Sie jeweils eine Beispiel-Folge für den besten und schlechtesten Fall an.

4. **Zusatzaufg.: Sort. mit Such- und AVL-Bäumen II** (mittel) (schr., 3+5 P.)
Programmieren Sie das Verfahren aus Aufgabe 3 in Ada 95 und messen Sie die Zahl der Vergleiche für unterschiedliche Werte von n (auf jeden Fall für $n = 1000$, 5000, 10000 und 20000, wobei Sie für jedes solche n eine n-elementige Folge von Zahlen zufällig fünfzig Mal erzeugen, die Sortierung durchführen, und den Mittelwert über die Vergleiche in diesen fünfzig Experimenten bilden)

   a) mit binären Suchbäumen.
   b) mit AVL-Bäumen.

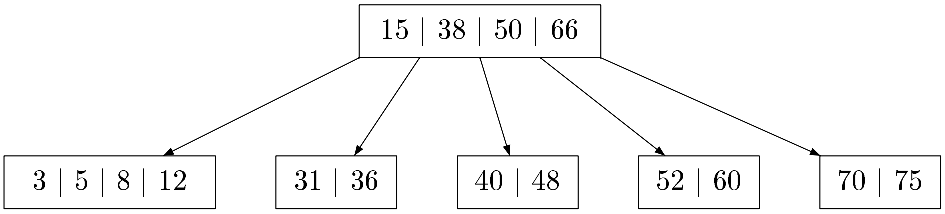
   Vergleichen Sie Ihre experimentellen Ergebnisse mit den theoretischen aus Aufgabe 3.

5. B-Bäume (leicht-mittel) (votieren, 2+2 Punkte)

Einfügen und Löschen in B-Bäumen:

a) Fügen Sie in einen zunächst leeren B-Baum der Ordnung 1 die Elemente 10, 7, 4, 13, 17, 1, 2, 3 in dieser Reihenfolge ein und geben Sie den resultierenden B-Baum an.

b) Gegeben sei folgender B-Baum der Ordnung 2:



Löschen Sie mehrere Knoten aus diesem B-Baum, so dass dabei alle Fälle (Knoten ist nicht Blatt, Löschen im Blatt, Borgen vom Bruderknoten, Verschmelzen mit Bruderknoten) auftreten.

6. Hashing (leicht) (votieren, 2 Punkte)

In eine Hashtabelle mit 11 Einträgen (Indizes 0 bis 10) sollen die Wörter CHIARELLO, CLAUS, HOLUB, JEHLICKA, KAHL, LEWANDOWSKI, SCHILLER, SCHMID, ZIMMER in der angegebenen Reihenfolge eingefügt werden. Die verwendete Hashfunktion ist nach der Teilfolgenmethode (3.3.2.4) zu wählen, wobei den Buchstaben A bis Z die Werte 1 bis 26 zugeordnet werden (vgl. 3.3.1 Folie 8), als Teilfolge wird der 2. Buchstabe (mit Gewicht 3) und der 4. Buchstabe (mit Gewicht 2) gewählt. Kollisionen sollen mit dem Verfahren der linearen Sondierung aufgelöst werden. Die Schrittweite ist 1.

Geben Sie die ausgefüllte Tabelle, sowie für jeden Eintrag alle die im Zuge der Kollisionsauflösung berechneten Indizes an.

7. Zusatzaufgabe: Perfektes Hashing (mittel-schwer) (schriftlich, 4 Punkte)

Finden Sie eine perfekte Hashfunktion (nach der Teilfolgenmethode) für die obigen 9 Elemente für eine Hashtabelle mit 11 Einträgen. Finden Sie auch eine perfekte Hashfunktion, wenn die Hashtabelle nur 9 Einträge hat?

8. Löschen in Hashtabellen mit Linearer Sondierung (mittel) (vot., 4 Pkt.)

Überlegen Sie sich, welche Probleme auftreten, wenn man einzelne Namen aus der Tabelle aus Aufgabe 6 herausstreicht (löschen Sie jeweils einen einzelnen Namen, und überlegen Sie sich wie andere Tabelleneinträge verschoben werden müssen, damit das Auffinden der anderen Namen (und auch das Suchen nach Namen, die nicht in der Tabelle vorkommen) wie gewünscht funktioniert). Schreiben Sie eine Ada 95 Prozedur, die das Löschen in Hashtabellen mit Linearer Sondierung durchführt.

Es werden maximal 20 (der erreichbaren 32) Punkte des Aufgabenblattes gewertet.
Alles weitere unter

http://www.informatik.uni-stuttgart.de/fmi/fk/lehre/ss03/info_II/default.htm