Universität Leipzig
Institut für Informatik
Abteilung Datenbanken
Prof. Dr. E. Rahm
T. Böhme, Dr. P. Richter

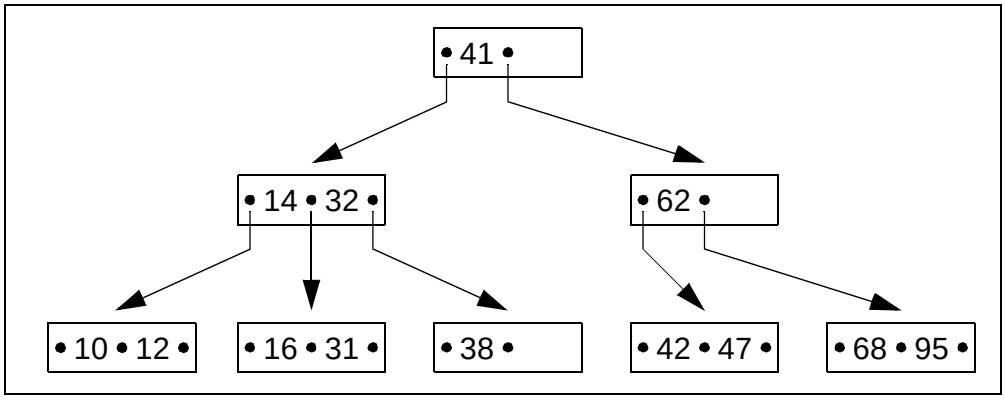
Ausgabe: 06.05.2002
Abgabe: 27.05.2002

Algorithmen und Datenstrukturen 2
SS 2002 - Übungsblatt 3

Hinweis: Alle *Listings* müssen in der ersten Zeile nach der Klassendefinition den *Namen des Programmautors* als Kommentar enthalten. Ohne diesen Namen werden die Listings **nicht gewertet**. Ebenfalls führt das Fehlen eines geforderten Ausdrucks der Programmausführung zur Nichtbewertung der Aufgabe.

1. Aufgabe (Verallgemeinerte Überlaufbehandlung) (6 Punkte)

Gegeben sei der folgende 2-3-Baum:



Fügen Sie in diesen Baum die folgenden Werte ein: 11, 51 und 64. Verwenden Sie für den Einfügealgorithmus die verallgemeinerte Überlaufbehandlung (Folie 1-38). Fügen Sie die Werte einmal für m=1 und einmal für m=3 ein.

Hinweis: Die verallgemeinerte Überlaufbehandlung wurde für B*-Bäume eingeführt, läßt sich jedoch analog auch für B-Bäume anwenden.

2. Aufgabe (Digitale Suchbäume) (6 Punkte)

In dieser Aufgabe sollen Sie verschiedene digitale Suchbäume aus der folgenden Schlüsselmenge erstellen:

UNO, UNHCR, UNHCRHR, UNICEF, UNICRI, UNDP, UNDOF, UNESCO, UNESCOHQ

a) Erstellen Sie aus der Schlüsselmenge einen Digitalbaum (analog zu Folie 1-45). Verwenden Sie einzelne Buchstaben als Schlüsselteile.

b) Erstellen Sie aus der Schlüsselmenge einen Trie. Führen Sie in den Knoten nur die besetzten Verweise auf (Knotenkompression - die Angabe des zugehörigen Schlüsselteils ist erforderlich)

c) Erstellen Sie aus der Schlüsselmenge einen Radix-Baum (analog zu Folie 1-51).

3. Aufgabe (Hash-Tabelle) (4 Punkte)

In eine leere Hash-Tabelle werden *n* Schlüssel eingefügt. Wieviele Vergleiche werden dafür im schlechtesten Fall benötigt, wenn zur Überlaufbehandlung die Methode der Verkettung der Überläufer mit unsortierten bzw. sortierten Listen verwendet wird? Wieviele Schritte benötigt man in beiden Fällen, um nach jedem der *n* eingefügten Schlüssel einmal zu suchen?

4. Aufgabe (Löschen in Hash-Tabellen) (8 Punkte)

Implementieren Sie in Java das Löschen von Einträgen in einer Hash-Tabelle mit verketteter Überlaufbehandlung. Auf der Web-Übungsseite haben wir Ihnen dafür die Klasse *LinkedHashTable* bereitgestellt, die eine Hash-Tabelle teilweise implementiert. Die darin als Rumpf vorgegebene Methode *public void remove (Object key)* ist von Ihnen zu implementieren. Sie soll den Eintrag, der zu dem gegebenen Schlüsselwert (key) gehört, aus der Hash-Tabelle entfernen. Beachten Sie, dass auch nicht existente Schlüsselwerte übergeben werden können.

Überprüfen Sie die Implementierung mit dem ebenfalls bereitgestellten Programm *Ueb3Test*, welches mittels vorgegebenen Werten eine Hash-Tabelle füllt und aus dieser mittels der von Ihnen implementierten Loeschmethode einige Werte entfernt. Zur Kontrolle wird der Inhalt der Hash-Tabelle vor und nach dem Löschen ausgegeben. Für das Übersetzen und Ausführen des Programms benötigen Sie die auf der Web-Übungsseite bereitgestellten Dateien *HTEntry.java*, *HTLinkedEntry.java*, *HashTable.java*, *LinkedHashTable.java* und *Ueb3Test.java*.

Abgabe:

1. dokumentiertes Listing des Quellprogramms (*LinkedHashTable.java*)
   Hinweis: Methoden, die Sie nicht geändert haben, können weggelassen werden.
2. Ausgabe eines Programmlaufs von *Ueb3Test*