

Prof. Dr. Bernhard Seeger  
Jochen van den Bercken

Übungen zur Vorlesung  
**Implementierung von Datenbanksystemen**

Sie erhalten in der Übung vom 19.5. nähere Informationen zu den Klassen der XXL-Bibliothek!  
Abgabe am 2. 6. in der Übung!

**Aufgabe 3.1:**  $B^+$ -Bäume (2+2)

- Bestimmen Sie  $b$  und  $b^*$  für einen  $B^+$ -Baum bei gegebener Seitengröße  $p$  und Schlüsselgröße  $s$ . Verweise innerhalb des Baumes haben die Größe  $v$ , die TIDs die Größe  $d$ . Berechnen Sie  $b$  und  $b^*$  für den Fall  $p=4096$ ,  $s=4$ ,  $v=6$  und  $d=8$ .
- Geben Sie für einen  $B^+$ -Baum je eine Formel an, mit der man die obere und untere Schranke für die Höhe des Baumes bei gegebenem  $b$ ,  $b^*$  und  $n$  (der Anzahl der eingetragenen TIDs) bestimmen kann.

**Aufgabe 3.2:** Parallele  $B^+$ -Bäume (3+2+2+3+2)

Betrachten Sie in dieser Aufgabe nur Einfügeoperationen!

- Fügen Sie die folgenden Zahlen in dieser Reihenfolge in einen anfangs leeren parallelen  $B^+$ -Baum, der auf 5 Festplatten aufgeteilt werden soll. Die Seitenkapazität des  $B^+$ -Baumes sei 3.

(99,156,55,153,192,81,130,98,101,124,172,100,12,124,15,183,18,51,63,59,181,72)

Geben Sie jeweils nach jedem Seitensplit die Zuordnung der Seiten zu den Platten an.

- Wieviele Seiten pro Level sind einer Festplatte mindestens zugeordnet? Begründen Sie!
- Wieviele Seiten pro Level sind einer Festplatte maximal zugeordnet? Geben Sie eine detailliertere Formel als im Skript an. Begründen Sie!
- Bei welcher Einfügereihenfolge der Zahlen aus a) wird der Worst-Case erreicht?
- Beschreiben Sie, wie der Löschalgorithmus für parallele  $B^+$ -Bäume zu modifizieren ist. Gehen Sie davon aus, daß bei einem Seitenunterlauf eine Verschmelzung mit einer der beiden Nachbarseiten stattfindet.

**Aufgabe 3.3:** Implementierung von parallelen  $B^+$ -Bäumen (15)

Ein wesentlicher Vorteil bei der Implementierung von Bäumen unter XXL besteht in der Abstraktion vom Speichermedium. Dazu nutzen Bäume die Klasse Container, welche einen abstrakten Zugriff auf Objektmengen leistet, wobei in diesem Fall die Objekte die Knoten der Bäume sind. Bei einem Split eines Knotens (oder dem Anlegen einer neuen Wurzel) wird vom Baum eine vom Benutzer spezifizierte Funktion aufgerufen (determineContainer), die für den neuen Knoten bestimmt, in welchem Container dieser zu speichern ist.

In XXL steht bereits eine Implementierung konventioneller  $B^+$ -Bäume zur Verfügung. Um die Klasse BTree auf parallele  $B^+$ -Bäume umzustellen, muß einerseits diese Funktion determineContainer geeignet implementiert werden, andererseits müssen die Verweise auf Knoten (Klasse IndexEntry) um ein Attribut erweitert werden, welches den Container des Knotens spezifiziert, auf den verwiesen wird.

Sie erhalten in der Übung vom 19.5. detailliertere Informationen über die entspr. Klassen.

