
Algorithmen für die Speicherhierarchie

Abgabetermin: 8.7.2009 vor der Vorlesung

Buffer Trees

Aufgabe 1

Der in der Vorlesung vorgestellte Buffer Tree hat einen Verzweigungsgrad $q = \Theta\left(\frac{M}{B}\right)$ für alle inneren Knoten außer der Wurzel. Wie verändern sich die amortisierten Kosten pro Operation, wenn ein Grad $q = o\left(\frac{M}{B}\right)$ bzw. $q = \omega\left(\frac{M}{B}\right)$ verwendet wird.

Aufgabe 2

Wir wollen nun Buffer Trees um eine Lookup Operation erweitern, welche es ermöglicht zu einem gegebenen Schlüssel k den Inhalt des Elements mit Schlüssel k zu erhalten. Es soll genügen diese Lookup Operation als *batched* Operation anzubieten, d.h. es wird zusätzlich ein Befehl angeboten, nach dessen Aufruf keine Update-Operationen des Buffer Trees mehr erfolgen und alle gesammelten Lookup-Operationen ausgeführt werden.

Zeigen Sie, dass diese Lookup Operationen mit amortisierten Kosten $\mathcal{O}\left(\frac{1}{B} \log_{M/B}\left(\frac{N}{B}\right)\right)$ realisierbar sind.

Hinweis: Die Datenstruktur des Buffer Tree ist nicht unbedingt nötig!

Aufgabe 3

Auf dem Übungsblatt 4 wurde eine alternative Balancierungsmethode für B-Bäume vorgestellt. Dabei wurde der Grad eines inneren Knotens v nicht durch die Parameter a und b eingeschränkt, sondern die Anzahl der Blätter im Unterbaum mit Wurzel v (das Gewicht von v) beschränkt. Das Gewicht eines beliebigen Knotens mit Distanz i zu den Blättern beträgt dann höchstens t^i und für alle Knoten außer der Wurzel mindestens s^i .

Gibt es passende Werte für s und t , so dass ein so definierter Buffer Tree amortisierte Kosten von $\mathcal{O}\left(\frac{1}{B} \log_{M/B}\left(\frac{N}{B}\right)\right)$ hat? Falls ja geben Sie diese Werte an und zeigen Sie, dass die amortisierten Kosten erhalten bleiben.