Technische Universität München
Fakultät für Informatik
Lehrstuhl für Effiziente Algorithmen
Dr. Hanjo Täubig
Jeremias Weihmann

Sommersemester 2013
Übungsblatt 11
1. Juli 2013

Grundlagen: Algorithmen und Datenstrukturen

Abgabetermin: Jeweilige Tutorübung in der Woche vom 8. bis 12. Juli

Tutoraufgabe 1

Diese Tutoraufgabe hat eine Eigenschaft von $(2,3)$-Bäumen zum Gegenstand, die für die zweite Hausaufgabe ausgenutzt werden kann. In der zweiten Hausaufgabe wiederum wird eine Aussage aus der Vorlesung bewiesen.

Wir betrachten einen $(2,3)$-Baum, der nur das Dummy-Element ∞ enthält, und fügen in den Baum der Reihe nach die Zahlen $1, 2, \ldots, n$ ein. Zeigen Sie, dass der resultierende Baum ein echter Binärbaum ist, wenn $n+1$ eine Zweierpotenz ist.

Ein Binärbaum ist echt, wenn jeder innere Knoten genau zwei Kinder hat.

Hinweis: Betrachten Sie den Pfad von der Wurzel bis zum rechtesten/größten Blatt des Baumes (das Dummy-Element ∞). Sehen Sie einen Zusammenhang zwischen den Knotengraden der inneren Knoten, die auf diesem Pfad liegen, und der Zahl der Blätter im Baum? Führen Sie drei/vier Einfügeoperationen durch, um ein Gefühl für den Zusammenhang zu entwickeln.

Tutoraufgabe 2

Führen Sie auf einem anfangs leeren $(2,4)$-Baum folgende Operationen aus und zeichnen Sie die Zwischenergebnisse: `insert(23)`, `insert(30)`, `insert(13)`, `insert(6)`, `insert(40)`, `insert(80)`, `insert(62)`, `insert(75)`, `insert(28)`, `insert(21)`, `insert(29)`, `remove(62)`, `remove(75)`, `remove(13)`.

Anmerkung: Zum leichteren Verständnis der Operationen beobachten wir, dass ein Split-Schlüssel stets dem größten (d.h. rechtesten) Blatt entspricht, das sich in dem zu dem Split-Schlüssel korrespondierenden Unterbaum befindet.

Zusatzaufgabe 1

Um zu zeigen, dass jeder AVL-Baum logarithmische Tiefe in der Zahl der Elemente besitzt, wurde in der Vorlesung wurde wie folgt argumentiert:

(i) Die Baumtiefe bei Fibonacci-Bäumen ist logarithmisch in der Zahl seiner Knoten.

(ii) Ein Fibonacci-Baum der Tiefe t besitzt von allen AVL-Bäumen mit Tiefe t die geringste Zahl an Knoten.

(iii) Daher hat jeder AVL-Baum logarithmische Tiefe in der Knotenzahl.

Führen Sie die Argumentation, insbesondere Teil (ii), detailliert aus.

Zur Erinnerung: Fibonacci-Bäume sind rekursiv definiert:

- Ein Fibonacci-Baum der Stufe 0 ist der leere Baum.
- Ein Fibonacci-Baum der Stufe 1 besteht nur aus der Wurzel.
- Ein Fibonacci-Baum der Stufe $h \geq 2$ besteht aus der Wurzel, deren Kinder Fibonacci-Bäume der Stufen $h-1$ und $h-2$ sind.

Hausaufgabe 1

Implementieren Sie in der Klasse `UIbiHeap` einen Binomial-Heap. Hierzu muss auch ein BinomialBaum in der Klasse `binomialTree` implementiert werden.

Verwenden Sie für Ihre Implementierung die auf der Übungswebseite bereitgestellten Klassen und verändern Sie für Ihre Implementierung *ausschließlich* die Klassen `UIbiHeap` und `binomialTree`.

Achten Sie bei der Abgabe Ihrer Aufgabe darauf, dass Ihre Klassen `UIbiHeap` und `binomialTree` heißen und auf den Rechnern der Linuxhalle (`lxhalle.informatik.tu-muenchen.de`) mit der bereitgestellten Datei `main_bi` kompiliert werden können. Andernfalls kann eine Korrektur nicht garantiert werden. Achten Sie darauf, dass Ihr Quelltext ausreichend kommentiert ist.

Schicken Sie die Lösung per Email mit dem Betreff `[GAD] Gruppe <Gruppennummer>` an Ihren Tutor.

Hausaufgabe 2

Beweisen Sie folgende Aussage: Für einen $(2,3)$-Baum gibt es eine Folge von n insert bzw. remove-Operationen, sodass die Anzahl der nötigen Aufspaltungen und Vereinigungen in $\Omega(n \log n)$ ist.

Hinweis: Verwenden Sie in Abhängigkeit von n einen Teil der Operationen dafür, einen großen echten Binärbaum zu konstruieren, und den Rest der Operationen auf solche Weise, möglichst viele Aufspaltungen und Vereinigungen zu erhalten. Beachten Sie auch die erste Tutoraufgabe.

Hausaufgabe 3

Führen Sie auf einem anfangs leeren $(2,3)$-Baum folgende Operationen aus und zeichnen Sie die Zwischenergebnisse: `insert(11)`, `insert(66)`, `insert(22)`, `insert(55)`, `insert(33)`, `insert(44)`, `insert(99)`, `insert(7)`, `insert(6)`, `insert(5)`, `insert(4)`, `remove(22)`, `remove(11)`, `remove(7)`, `remove(44)`, `remove(66)`, `remove(55)`, `remove(4)`,

Hausaufgabe 4

Für die Definition von Fibonacci-Baum siehe Vorlesung oder Zusatzaufgabe 1.

(a) Zeigen Sie mit (starker) vollständiger Induktion, dass jeder Fibonacci-Baum der Stufe $h > 0$ Baumtiefe $t_h = h - 1$ besitzt.

(b) Seien $a > b > 1$ Konstanten. Zeigen Sie, dass eine Konstante $c > 0$ existiert, sodass $a^n - b^n \geq c \cdot a^n$ für alle $n \geq 1$ gilt.