Technische Universität München
Fakultät für Informatik
Lehrstuhl für Effiziente Algorithmen
Dr. Riko Jacob
Matthias Baumgart

Sommersemester 2009
Übungsblatt 9
14. Juli 2009

Effiziente Algorithmen und Datenstrukturen II

Abgabetermin: 23.07.2009 **vor** *der Vorlesung*

Aufgabe 1 (3 Punkte)

Wir betrachten den van-Emde-Boas-Baum mit den Operationen `insert` und `findsucc`. Zeigen Sie, dass es damit möglich ist n Elemente aus einem Universum U, $U > n$, in Zeit $O(n \log\log U)$ zu sortieren. Mehrfach auftretende Elemente sollen ebenfalls berücksichtigt werden.

Aufgabe 2 (3 Punkte)

Wir betrachten die erste Implementierung der Operation `insert` in der Zusammenfassung von Gudmund (siehe Literaturverzeichnis auf der Webseite). Zeigen Sie, dass diese Implementierung im worst-case die Laufzeit $O(\log U)$ hat.

Aufgabe 3 (4 Punkte)

In einem Algorithmus soll ein großes Array verwendet werden. Der Algorithmus verlangt, dass zu Beginn alle Arraypositionen initialisiert werden (zum Beispiel mit NULL). Einerseits dauert uns das Initialisieren des Arrays aber zu lang, andererseits kann in einem nicht-initialisierten Speicher natürlich beliebiger Unsinn stehen. Beschreiben Sie eine Möglichkeit, wie man ohne die aufwendige Initialisierung auskommen kann. Genauer gesagt soll beim lesenden Zugriff auf eine Array-Position das Folgende passieren:

- Falls der Algorithmus vorher einen Eintrag in diese Position geschrieben hat, wird dieser Eintrag zurückgegeben.
- Falls in diese Position noch nicht geschrieben wurde (diese also „Bitmüll“ enthält), wird NULL zurückgegeben.

Wie kann man ein solches Array so implementieren, dass ein (lesender oder schreibender) Zugriff auf ein Arrayelement Zeit $O(1)$ kostet und dass wir beim lesenden Zugriff immer die richtige Antwort zurückgeliefert bekommen? Das Zurücksetzen des Arrays (jeder lesende Zugriff liefert wieder NULL) soll ebenfalls in Zeit $O(1)$ möglich sein.

Hinweis: Benutzen Sie eine Extradatenstruktur, in der steht, welche Positionen des Arrays schon *echt* beschrieben wurden. (Dort werden sozusagen *Zeugen* der Echtheit verwaltet.) Benutzen Sie außerdem eine Extrainformation in jeder Arrayposition, die gegebenenfalls auf den Zeugen verweist.