Technische Universität München
Institut für Informatik
Lehrstuhl für Effiziente Algorithmen
Prof. Dr. Angelika Steger

Wintersemester 2001/02
Übungsblatt 3
6. November 2001

Effiziente Algorithmen und Datenstrukturen I

Abgabetermin: Dienstag, den 13. November 2001 (in der Übung)

Hausaufgabe 1

Im Grundstudium wurden die Fibonacci-Zahlen F_n definiert als:

$$F_0 = 0, \qquad F_1 = 1, \qquad F_{n+2} = F_{n+1} + F_n \quad (n \geq 0)$$

1. Zeigen Sie das gilt:
$$\forall n \geq 0 \ : \quad F_{n+2} = 1 + \sum_{i=0}^{n} F_i$$

2. Zeigen Sie: Ist x ein beliebiger Knoten in einem Fibonacci-Heap und r der Rang von x, dann enthält der Baum mit Wurzel x mindestens F_{r+2} Knoten. (Diese Eigenschaft gab dem Fibonacci-Heap seinen Namen.)

Hausaufgabe 2

Angenommen Sie führen bei einem Fibonacci-Heap einen *Cascading-Cut* erst dann durch, wenn ein Knoten sein drittes Kind verliert, d.h. Sie erlauben einem Knoten zwei Kinder zu verlieren und erst beim dritten Kind schneiden Sie den Knoten von seinem Vater ab. Was gilt dann für die Anzahl der Knoten eines Baumes mit Rang r ? Was folgt daraus für die amortisierten Kosten von *DecreaseKey* und *DeleteMin* ?

Hausaufgabe 3

Verallgemeinern Sie die Idee der 2-Level-Buckets auf k-Level-Buckets. Die amortisierte Laufzeit von *Insert* und *DecreaseKey* soll wieder $\mathcal{O}(1)$ sein, die von *ExtractMin* nur noch $\mathcal{O}(C^{\frac{1}{k}})$, wobei C wieder die obere Schranke für die Differenz zwischen dem größten und dem kleinsten Schlüssel sein soll.