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

Wintersemester 2001/02
Übungsblatt 2
30. Oktober 2001

Effiziente Algorithmen und Datenstrukturen I

Abgabetermin: Dienstag, den 6. November 2001 (in der Übung)

Hausaufgabe 1

Berechnen Sie das Potential eines Splay-Trees mit n Knoten, der

1. zu einer Liste entartet ist.
2. aus einem vollständigen binären Baum besteht.

Hausaufgabe 2

Sowohl bei Splay-Trees als auch bei Fibonacci-Heaps wissen wir, daß die amortisierten Kosten „klein" sind. Beide Datenstrukturen aber können durch eine ungünstige Eingabesequenz entarten, so daß eine einzelne Operation viel Zeit benötigen kann.
Geben Sie für beide Datenstrukturen jeweils eine Konstruktion einer Eingabesequenz mit $O(n)$ Operationen an, bei der die Baumhöhe am Ende $\Theta(n)$ ist.

Hausaufgabe 3

Beschreiben Sie eine Implementierung für eine *Delete* Operation bei Fibonacci-Heaps, die nicht mehr als $O(\log n)$ amortisierte Kosten hat, wobei n die Anzahl der Elemente im Fibonacci-Heap ist.

Hausaufgabe 4

Die Kosten für eine *DeleteMin* Operation beträgt bei Fibonacci-Heaps $O(\log n)$, wobei n die Anzahl der in der Datenstruktur gespeicherten Elemente ist.
Angenommen die Schlüssel der Elemente können nur Werte aus $\{1, \ldots, k\}$ annehmen und $k \ll n$. Überlegen Sie sich, wie man einen Fibonacci-Heap modifizieren und erweitern muß, damit die *DeleteMin* Operation nur noch $O(\log k)$ Kosten hat und die amortisierten Kosten für *DecreaseKey* und *Insert* nach wie vor konstant bleiben.
Hinweis: Benutzen Sie einen Fibonacci-Heap in dem nur Elemente mit verschiedener Priorität sind und verwalten Sie die restlichen Elemente in zusätzlichen Datenstrukturen.