Technische Universität München | WS 1994/95
Institut für Informatik | Übungsblatt 3
Prof. Dr. Ernst W. Mayr | 22. Dez. 1994

Algorithmen und Datenstrukturen – Effiziente Algorithmen I

(Hausaufgaben-Abgabetermin: 30. November 1994 (!!), 10:00Uhr)

Aufgabe 1

(i) Zeigen Sie detailliert, daß die Merge-Operation zweier Binomial-Vorrangwarteschlangen (mit zusammen n Elementen) in Zeit $O(\log n)$ durchgeführt werden kann.
(ii) Eine Binomial-Vorrangwarteschlange werde aufgebaut, indem n einzelne Elemente der Reihe nach in die zunächst leere Priority Queue eingefügt werden. Bestimmen Sie den Gesamtzeitbedarf für die n aufeinanderfolgenden INSERT-Operationen.

Aufgabe 2

Sei T ein Baum in einer Fibonacci-Heap-Datenstruktur mit n Knoten. Zeigen Sie, daß die Tiefe von T $O(\log n)$ ist.

Aufgabe 3

Wir betrachten folgende Modifikation der Implementierung von Fibonacci-Heaps ("verzögertes Löschen"): Bei einer DELETE-Operation wird der entsprechende Knoten nicht entfernt, sondern lediglich als gelöscht markiert. Bei der FIND_MIN-Operation werden alle maximalen Unterbäume mit nicht als gelöscht markierter Wurzel in einer Liste zusammengefaßt und miteinander verbunden, wie bei DELETE_MIN in der Vorlesung besprochen. Seien $r_1, \ldots, r_d$ die Ränge der Wurzeln dieser Unterbäume. Zeigen Sie:

$$\sum_{i=1}^{d} r_i = O(d \cdot \log(n/d)) \ .$$

Aufgabe 4

Entwickeln Sie in C++ einen Datentyp, der Mengen von Zahlen vom Typ `unsigned` repräsentiert und auf (2,4)-Bäumen basiert. Der Datentyp soll über die Operationen INSERT, DELETE, FIND_MIN sowie DELETE_MIN verfügen. Für den Test Ihrer Implementierung stehen auf den hphalle-Maschinen unter

`/usr/wiss/stadther/ue3`

die Programme `ue3gen` und `ue3ref` sowie der Testrahmen `ue3test.C` zur Verfügung. Mit `ue3gen` lassen sich Codierungen von Operationsfolgen erzeugen. Der Aufruf `ue3gen 100000 100` z.B., erzeugt eine zufällige Folge von 100000 Operationen mit Schlüsselwerten im Bereich 0 bis 99. Der Testrahmen liest solche Folgen ein, ruft die entsprechenden Operationen Ihrer Datenstruktur auf und gibt die Folge der Ergebnisse von FIND_MIN aus. Die Ergebnisfolge kann mit der Ausgabe des Programms `ue3ref` verglichen werden, welches den gleichen Testrahmen und eine Referenzimplementierung von (2,4)-Bäumen enthält. Messen Sie die Laufzeit Ihrer Implementierung für verschieden lange Operationsfolgen.