



Universität Freiburg
Institut für Informatik
PD Dr. A. Heinz
PD Dr. S. Schuierer

Georges-Köhler-Allee, Geb. 051
D-79110 Freiburg i. Br.
Tel. (0761) 203-8164
Tel. (0761) 203-8165
Freiburg, 15. November 2000

Algorithmentheorie Übungsblatt 3

Abgabe bis *Montag*, 27.11.2000, (in der Vorlesung)
Besprechung am *Mittwoch*, 29.11.2000, 11-13 Uhr

Aufgabe 1: (1 + 3 Punkte) Amortisierung

Ein Bit-Zähler erlaubt die Operationen *Increment* und *Reset* zum Erhöhen einer k -Bit Zahl um 1 bzw. zum Zurücksetzen der Zahl auf 0.

- Beschreiben Sie eine Implementierung eines solchen Bit-Zählers, die auch für große k effizient ist. Geben Sie die tatsächlichen Kosten für beide Operationen an.
- Wählen Sie eine geeignete Potentialfunktion Φ und geben Sie die amortisierten Kosten der beiden Operationen an. Zu Beginn steht der Bit-Zähler auf Null. (Hinweis: Beziehen Sie die Position der führenden 1 in das Potential mit ein.)

Aufgabe 2: (4 Punkte) Dynamische Tabellen

Betrachten Sie die folgende Strategie zur Kontraktion einer dynamischen Tabelle: Sobald der Belegungsfaktor der Tabelle T unter $1/3$ fällt, wird die Tabelle T ersetzt durch eine neue Tabelle T' , deren Größe $2/3$ der Größe von T ist. Es sei Φ wie folgt definiert:

$$\Phi(T) = |2T.num - T.size|.$$

Zeigen Sie, daß die amortisierten Kosten einer Table-Delete-Operation durch eine Konstante nach oben beschränkt sind.

Aufgabe 3: (3 Punkte) Binomial Queues

Zeigen Sie, daß für eine geeignet gewählte Potentialfunktion die Operation `delete-min` auf Binomial Queues amortisierte Zeit $O(1)$ benötigt und die Operation `insert` amortisierte Zeit $\Theta(\log n)$. Wie interpretieren Sie dieses Ergebnis?

Aufgabe 4: (5 Punkte) Fibonacci Heaps

Führen Sie auf einem anfangs leeren Fibonacci Heap die folgenden Operationen aus:

```
insert(45), insert(33), insert(28), insert(21), insert(17), insert(14),  
insert(9), insert(6), insert(5), insert(1), delete-min,
```

`decrease-key(33,11)`, `delete(21)`, `decrease-key(28,3)` und `delete-min`.

Geben Sie jeweils Struktur und Schlüsselwerte (samt Markierungen) des Fibonacci Heaps für alle Zwischenergebnisse an. Das Einfügen soll jeweils vor dem aktuellen minimalen Element erfolgen.

Zusatzaufgabe 5: (2 + 2 Punkte) Fibonacci-Heaps

a) Nehmen Sie an, wir führen folgende Änderungen bei Fibonacci-Heaps durch:

1. Das Löschen von Markierungen soll nur in den Methoden *decrease-key* und *delete* erfolgen. Und zwar werden immer die Markierungen derjenigen Knoten gelöscht, die von ihrem Vater abgetrennt werden, d.h. wir fügen in der Methode *cut* die Zuweisung

$$v.mark = false$$

ein und die Methode *link* setzt keine Markierungen zurück.

2. m_Q wird als Gesamtanzahl *aller* markierten Knoten von Q definiert. Die Potentialfunktion bleibt ansonsten unverändert.

Berechnen Sie nun die amortisierten Kosten einer *delete-min* und einer *decrease-key*-Operation.

- b) In dem Buch von Ottmann/Widmayer wird vorgeschlagen, in einer Link-Operation anstelle der Markierung desjenigen Knotens, der zum Sohn wird, die Markierung desjenigen Knotens, der zum Vater wird, zu löschen. Bei welcher Operation kann dieses Vorgehen zu nicht-konstanten amortisierten Kosten führen? Begründen Sie Ihre Antwort.