Technische Universität München
Fakultät für Informatik
Lehrstuhl für Effiziente Algorithmen
Prof. Dr. Ernst W. Mayr
Matthias Baumgart

Wintersemester 2006/2007
Übungsblatt 8
8. Dezember 2006

Effiziente Algorithmen und Datenstrukturen I

Abgabetermin: 15.12.2006 **vor** *der Vorlesung*

Aufgabe 1 (10 Punkte)

Geben Sie eine graphische Darstellung (d.h. eine Darstellung des Arrays b mit seinen Listen sowie der Werte aus dem Array u) des Radix-Heaps an, der entsteht, wenn die Werte

`100, 29, 50, 17, 72, 49, 18, 42, 31, 24, 117`

in einen leeren Heap eingefügt werden. Führen Sie anschließend auf diesem Heap eine ExtractMin-Operation aus und geben Sie den resultierenden Heap an.

Aufgabe 2 (10 Punkte)

In der Vorlesung wurde die Laufzeit $\mathcal{O}(k \log^* n)$ für eine Union-Find-Struktur mit *weighted union* und *Pfadkompression* bewiesen. Hierbei wurde für jeden Knoten der folgende Wert definiert:

$$\mathit{rank}(x) = \text{Höhe des Unterbaums mit Wurzel } x \text{ in der Datenstruktur } \textit{ohne} \text{ Pfadkompression.}$$

a) Zeigen Sie, dass für alle Knoten in der Datenstruktur mit Pfadkompression gilt, dass die Werte von *rank* auf allen Pfaden von den Blättern zur Wurzel strikt steigen.

b) Sei $\mathit{rank}_p(x)$ der Wert von *rank* des aktuellen Vaters vom Knoten x. Zeigen Sie, dass $\mathit{rank}_p(x)$ bei einem Wechsel des Vaterknotens von x nicht kleiner wird.

Aufgabe 3 (10 Punkte)

Wir untersuchen eine alternative Implementierung von Union-Find-Strukturen: Jede der verwalteten Mengen wird als einfach verkettete Liste dargestellt, wobei zusätzlich jedes Element einen *Rückwärts-Zeiger* auf das Element am Listenanfang besitzt, welches als kanonisches Element der Menge dient. Dies ermöglicht Find-Operationen mit Kosten $O(1)$. Für Union(S_1, S_2) wird die kürzere der beiden Listen an das Ende der längeren Liste angehängt, und die Rückwärtszeiger der Elemente der kürzeren Liste werden aktualisiert. Jede Aktualisierung eines Zeigers kostet $O(1)$ Zeit.

(a) Beginnend mit n einelementigen Mengen führen wir $n-1$ Union-Operationen aus, bis wir bei einer einzigen n-elementigen Menge angelangt sind. Geben Sie eine Laufzeitabschätzung hierfür an.

(b) Geben Sie eine Implementierung an, die ein n-elementiges Feld statt einer linearen Liste verwendet.

Aufgabe 4 (10 Punkte)

Eine Union-Find-Struktur soll mittels Färbung der Elemente realisiert werden, d.h. zwei Elementen ist genau dann die gleiche Farbe zugeordnet, wenn sie in der gleichen Menge liegen. Werden zwei Mengen vereinigt, bekommen alle Elemente aus der Vereinigung die gleiche Farbe zugewiesen.

Zeigen Sie, dass es ein Verfahren gibt, bei dem, wenn man mit n 1-elementigen Mengen beginnt, für eine beliebige Folge von UNION-Operationen an deren Ende eine einzige Menge verbleibt (in der alle n Elemente enthalten sind), insgesamt nur $\mathcal{O}(n \log n)$ Umfärbungen nötig sind.