Technische Universität München
Institut für Informatik
Prof. Dr. J. Esparza/Dr. S. Gerke

WS 2000/01
Übungsblatt 7
5. Dezember 2000

Effiziente Algorithmen und Datenstrukturen

2. Kontrollblatt

(Abgabetermin: Dienstag, 19. Dezember in der Vorlesung)

Aufgabe 1

Gegeben sei eine Implementierung einer Union-Find-Struktur nach dem Schema Union-by-Rank. Das heißt, es soll an jedem Knoten die Höhe seines Unterbaums gespeichert und bei UNION stets der Baum mit dem kleineren Wurzelrang Unterbaum des anderen werden.

(i) Zeigen Sie, daß hierdurch die Höhe des Union-Find-Baums bei n Elementen durch $\mathrm{O}(\log n)$ beschränkt ist.

(ii) Warum gilt diese Schranke auch dann, wenn man den Rang eines Knotens als die Anzahl seiner Nachkommen im Union-Find-Baum definiert?

Aufgabe 2

Wir betrachten Union-Find mit statischer Vereinigung (d.h., die Wurzel des ersten der beiden Bäume wird auch die neue Wurzel) und mit Pfadkompression (d.h., alle Knoten auf dem Find-Pfad werden Kinder der Wurzel). Geben Sie, für $n > 0$, eine Folge von n Find- und $O(n)$ Union-Operationen (startend auf einem genügend großen Universum) an, so daß jede der Find-Operationen reelle Kosten $\Omega(\log n)$ hat.

Hinweis: Betrachten Sie die Vereinigung eines Binomialbaums B_k mit einem B_0 und überlegen Sie, was bei einer Find-Operation auf den tiefsten Knoten im B_k passiert.

Aufgabe 3

Entwickeln Sie eine Union-Find-Datenstruktur, bei der FIND(x) konstante Kosten und UNION(S_1, S_2) lineare Kosten verusacht. Geben Sie ein Laufzeitabschätzung an, wenn Sie beginnend mit n einelementigen Mengen $n-1$ UNION-Operationen ausführen, bis Sie bei einer einzigen n-elementigen Menge angelangt sind.

Aufgabe 4

Betrachten Sie eine Menge von Bäumen mit den Knoten $0, 1, \dots, n-1$. Die Operation LINK(r, v) macht die Wurzel r eines Baumes zum Kind des zu einem anderen Baum gehörenden Knoten v. Die Operation DEPTH(v) gibt die Tiefe des Knoten v aus. Zeigen Sie, wie man eine Union-Find-Datenstruktur zum Lösen dieser Aufgaben benutzen kann.

Tip: Man kann die Knoten eines Baumes T als eine der Mengen der Union-Find-Struktur auffassen, da diese in Bäumen gespeichert werden. Man kann nun jedem Knoten einen Wert $h(v)$ zuordnen, so daß gilt: DEPTH(v) $= \sum h(u)$ wobei über alle Vorfahren von v in der summiert wird.