Technische Universität München
Fakultät für Informatik
Lehrstuhl für Effiziente Algorithmen
Prof. Dr. Christian Scheideler
Johannes Nowak

Wintersemester 2006/2007
Übungsblatt 5
23. Mai 2007

Netzwerk-Algorithmen

Aufgabe 1

Zeigen Sie, dass die cut-and-paste Strategie mit n Knoten nur $O(\log n)$ Zeit benötigt, um die Position eines beliebigen Datenelements zu berechnen.
(Hinweis: Es reicht aus zu zeigen, dass jedes Datenelement nur $O(\log n)$-mal umplatziert wird, wenn das System von einen auf n Knoten schrittweise anwächst. Da die cut-and-paste Strategie diese Umplatzierungen in ihrer Positionsberechnung nachspielt, ergibt sich die Zeitschranke. Um zu zeigen, dass die Umplatzierungssprünge in etwa exponentiell anwachsen, betrachte zwei aufeinanderfolgende Umplatzierungen eines beliebigen Datenelements ausgehend von einer beliebigen Knotenposition.)

Zeigen Sie weiterhin, dass für jede perfekt gewissenhafte Strategy gilt: Wenn die Anzahl der Knoten schrittweise von 1 bis n steigt, dann muss im Durchschnitt ein Datenelement $\Omega(\log n)$-mal bewegt werden.
(Hinweis: Berechnen Sie Gesamtzahl der Datenbewegungen, und teilen Sie diese durch die Anzahl der Datenelemente.)

Aufgabe 2

Implementieren Sie in der Subjects Umgebung die konsistente Hashingstrategie auf einem Ring von 8 Knoten. Knoten $i \in \{0, \ldots, 7\}$ soll dabei der Position $i/8$ zugewiesen werden. Folgende Operationen sind weiterhin zu implementieren:

- Insert(d): fügt Datenelement d ein.
- Delete(d): löscht Datenelement d.
- Lookup(d): sucht nach Datenelement d.

d sei der Einfachheit halber eine natürliche Zahl. Verwenden Sie eine lineare Hashfunktion, um jedes d auf einen Wert in $[0, 1)$ abzubilden. Zum Beispiel:

$$h(x) = ((51 \cdot d + 23) \bmod 73)/73$$

Die Operationen sollen in einem beliebigen Knoten gestart werden können. Die Anfrage soll dann entlang des Ringes geschickt werden, bis der Knoten, der gemäß der konsistenten Hashingstrategie verantwortlich für das Datenelement ist, gefunden wurde. Daraufhin wird eine Antwort zurückgeschickt, die einfach nur "wahr" oder "falsch" zurückgibt. "Wahr" bedeutet dabei, dass die Insert/Delete/Lookup Anfrage erfolgreich war (d.h. das Element wurde eingefügt/gelöscht/ gefunden).