10. Aufgabenblatt zur Vorlesung

Höhere Algorithmik WiSe 2011/12

Wolfgang Mulzer, Claudia Dieckmann, Romain Grunert, Tillmann Miltzow, Lena Schlipf

Abgabe am 13. Januar 2012 vor der Vorlesung in die Mitarbeiterschachteln

Aufgabe 1 Hashing mit Verkettung *10 Punkte*

Angenommen, wir haben eine Hashtabelle mit n Plätzen, die eine Menge $S \subseteq K$ der Größe n speichert. Konflikte werden mit Hilfe von Verkettung gelöst. Wir nehmen an, dass sich die Hashfunktion wie eine zufällige Funktion verhält.

Für einen Platz i in der Hashtabelle, bezeichne mit Q_i die *Anzahl* der Schlüssel, die auf i gehasht werden. Unser Ziel ist es, $\mathbf{E}[\max_{i=0}^{n-1} Q_i]$ zu bestimmen, also den Erwartungswert für die maximale Anzahl an Elementen, die auf denselben Platz abgebildet werden.

Hinweis: Bei jedem Aufgabenteil, mit Ausnahme von (d), genügt eine Zeile als Antwort.

(a) Begründen Sie in einem Satz: Es gilt für $i = 0, \ldots, n-1$ und $r = 0, \ldots, n$,

$$\Pr[Q_i = r] = \left(\frac{1}{n}\right)^r \left(1 - \frac{1}{n}\right)^{n-r} \binom{n}{r}.$$

(b) Zeigen Sie,

$$\Pr\big[\max_{i=0}^{n-1} Q_i = r\big] \leq n \Pr[Q_0 = r].$$

(c) Aus der Stirling-Formel folgt die Abschätzung $\binom{n}{r} \leq (\frac{ne}{r})^r$. Benutzen Sie dies, um zu zeigen: $\Pr[Q_0 = r] \leq e^r/r^r$.

(d) Definiere $r_0 := c \log n / \log\log n$, für eine Konstante $c > 1$. Zeigen Sie, dass man c so wählen kann, dass $\Pr[Q_0 = r] < 1/n^3$ ist, für alle $r \geq r_0$.
Folgern Sie daraus:

$$\Pr\big[\max_{i=0}^{n-1} Q_i \geq r_0\big] \leq 1/n.$$

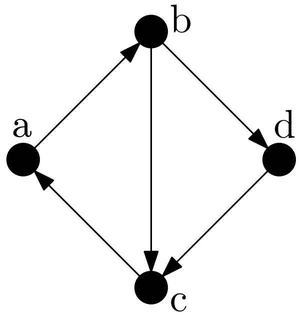
(e) Zeigen Sie

$$\mathbf{E}\big[\max_{i=0}^{n-1} Q_i\big] \leq r_0 \cdot \Pr\big[\max_{i=0}^{n-1} Q_i < r_0\big] + n \cdot \Pr\big[\max_{i=0}^{n-1} Q_i \geq r_0\big].$$

Folgern Sie daraus: $\mathbf{E}[\max_{i=0}^{n-1} Q_i] = O(\log n / \log\log n)$.
In der Vorlesung haben wir gezeigt, dass für alle i gilt: $\mathbf{E}[Q_i] = 1$. Ist das ein Widerspruch?

Aufgabe 2 Page-Rank *10 Punkte*

Betrachten Sie den folgenden Graphen.



(a) Geben Sie die modifizierte Adjazenzmatrix für den Graphen an.

(b) Bestimmen Sie den Page-Rank-Score für jeden Knoten algebraisch durch Lösen des Gleichungssystems (verwenden Sie den Dämpfungsfaktor 0.25).

(c) Führen Sie den iterativen Page-Rank-Algorithmus für das Beispiel durch (wieder mit Dämpfungsfaktor 0.25). Wie viele Iterationen sind notwendig, bis der absolute Fehler kleiner als 0.001 ist?

Hinweis: Sie können bei dieser Teilaufgabe gerne Octave oder Maple benutzen oder auch den Algorithmus selbst implementieren.

Aufgabe 3 Prioritätswarteschlangen *10 Punkte*

(a) Nennen Sie zwei Ihnen bekannte Implementierungen des abstrakten Datentyps *Prioritätswarteschlange*, und geben Sie die zugehörigen Laufzeiten an.

(b) Zeigen Sie, wie man mit Hilfe einer Prioritätswarteschlange eine Folge von n Elementen aus einem total geordneten Universum sortieren kann.

(c) Wie Sie wissen, benötigt jeder vergleichsbasierte Sortieralgorithmus mindestens $\Omega(n \log n)$ Operationen. In Anbetracht von (b), was besagt dies über die Laufzeit jeder vergleichsbasierten Implementierung einer Prioritätswarteschlange? Kann amortisierte Analyse hier helfen?