Ausgabedatum: 19.12.2013
Version: 2014-01-06 22:24

10. Übungsblatt für Track ε zur Vorlesung Entwurf und Analyse von Algorithmen, WS 13/14

Abgabe: *Bis* ***Freitag****, 10.01.2014, 12:00 Uhr, Kasten im Treppenhaus 48-6.*

Der Fehler der Woche

> `ENQUEUE` und `DEQUEUE` arbeiten korrekt, weil alle Teiloperationen korrekt arbeiten.

Was ist falsch?

Basisaufgaben

B1: Universelles Hashing — 2 Punkte

Zeigen Sie, dass für eine geeignete Konstante c und für alle $N, m \in \mathbb{N}$ die Menge aller Hashfunktionen $h : [1 \ldots N] \to [0 \ldots m]$ c-universell ist. Bestimmen Sie den minimalen Wert von c.

B2: Indirect Chaining mit Sortierung — 3 Punkte

Wir ändern Hashing mit Indirect Chaining so ab, dass die Listen *sortiert* vorgehalten werden.

Diskutieren Sie, inwiefern dies eine Verbesserung ist! Geben Sie dazu an, wie sich die erwarteten Such- und Einfügekosten qualitativ ändern und begründen Sie Ihre Behauptungen. Präzise Analysen sind nicht notwendig.

B3: Open Addressing zerlegen 3 Punkte

Wir haben bei der Diskussion von Hashing mit Open Addressing gesehen, dass darauf zu achten ist, dass die Kollisionsbeseitigung beginnend bei jeder Adresse jede andere erreichen kann, um in jedem Szenario maximale Auslastung zu ermöglichen.

Geben Sie für eine Kollisionsbeseitigungstrategie aus der Vorlesung Parameter an, die dies *nicht* leisten. Idealerweise sollte es schon für $\alpha \ll 1$ viele Hashwerte geben, für die keine weiteren Elemente eingefügt werden können.

B4: Optimale Suchbäume 4 Punkte

Sei eine Schlüsselmenge $A = \{a_1, \ldots, a_n\}$ mit Zugriffswahrscheinlichkeiten $P = \{p_1, \ldots, p_n\}$ mit $\sum_{i=1}^{n} p_i = 1$ gegeben. Wir nennen einen Suchbaum T (genau) dann optimal für A und P, wenn er die erwarteten Suchkosten $\sum_{i=1}^{n} p_i t_i$ minimiert. Hier ist t_i das Niveau des Knotens in T, der a_i speichert.

Zeigen Sie: es gibt unendlich viele Schlüsselfolgen A mit Wahrscheinlichkeitsgewichten P, für die alle optimalen binären Suchbäume lineare Listen sind, also Höhe $|A|$ haben.

Aufbauaufgaben

29. Aufgabe

3 Punkte

Wir betrachten Hashing mit Indirect Chaining und nehmen zur Vereinfachung an, $\mathsf{max} = m + 1$ sei eine gerade Zahl. Entgegen der bisherigen Annahmen seien nicht alle Hash-Folgen $a_1, a_2, \ldots, a_n$ gleichwahrscheinlich, sondern eine gerade Adresse (und die Null) werde doppelt so häufig durch unsere Hash-Funktion adressiert wie eine ungerade. Alle geraden Adressen (und die Null) untereinander seien gleichwahrscheinlich, ebenso alle ungeraden.

Bestimmen Sie unter dieser Annahme die erwarteten Kosten für eine erfolglose Suche in einer Hash-Tabelle mit n Schlüsseln. Dabei werde auch für die Suche eine gerade Adresse (und die Null) doppelt so häufig aufgesucht wie eine ungerade.

Hinweis: Zur Lösung dieser Aufgabe bietet es sich an, zuerst die Analyse für die uniforme Verteilung aller Hash-Folgen zu erarbeiten, um dann Veränderungen an den passenden Stellen vorzunehmen.

30. Aufgabe

3 Punkte

Wir betrachten Hashing mit Indirect Chaining und ersetzen die linearen Listen durch binäre Suchbäume. Wie in der Vorlesung sei $\alpha = \frac{N}{\mathsf{max}}$ und wir nehmen an, dass alle max^N Hashfolgen gleichwahrscheinlich sind. Zeigen Sie, dass für die Zahl an Vergleichen bei der erfolglosen Suche CU_N gilt, dass

$$\mathbb{E}[\mathrm{CU}_N] \sim 2\log(\alpha)$$

für $\alpha \to \infty$.

Hinweis:

(i) Für $P_{N,k}$ die Wahrscheinlichkeit, dass ein Baum genau k der N Schlüssel enthält, gilt:

$$\sum_{k=0}^{\infty}\sum_{i=1}^{k}\frac{P_{N,k}}{i} = \log(\alpha) + \mathcal{O}(1)$$

für $\alpha \to \infty$.

(ii) Aus der Gleichwahrscheinlichkeit der Hashfolgen ergibt sich, dass alle Endpositionen der erfolglosen Suche in einem Kollisionsbaum in Erwartung (über alle isomorphen Suchbäume und alle möglichen Suchschlüssel) gleich wahrscheinlich sind[1].

[1] Können Sie das zeigen?

31. Aufgabe

1 + 5 Punkte

Ein *Superstar* ist eine Person, die von allen anderen Personen gekannt wird, die selbst aber keine andere Person kennt.

a) Wenn Sie eine Menge von n Personen und die Relation „a kennt b" als Digraph modellieren, also „a kennt b" $\iff$ $(a, b) \in E$, wie können Sie dann einen Superstar charakterisieren?

b) Sei der Digraph aus Teilaufgabe a) als $n \times n$ Adjazenzmatrix A gegeben. Entwerfen Sie einen Algorithmus, der bei Eingabe A in Zeit $o(n^2)$ entscheidet, ob die durch A modellierte Population einen Superstar besitzt.

   Begründen Sie die Korrektheit Ihres Algorithmus und weisen Sie eine dem Zwecke der Aufgabenstellung dienliche Laufzeitschranke nach.

Wir wünschen frohe und besinnliche Feiertage und
einen guten Rutsch ins neue Jahr!

Graphik von Alain Matthes, siehe `http://tex.stackexchange.com/a/39211`