

Übungen zur Vorlesung Effiziente Algorithmen

Blatt 7

Hinweis zur Bearbeitung des Blattes: Für die Analyse von offen adressierten Hashtabellen dürfen Sie immer *Uniform Hashing* voraussetzen. Außerdem finden Sie unter

http://www.geocities.com/primes_r_us/small/index.html
<http://www.rsok.com/~jrm/printprimes.html>

Werkzeuge zum Auffinden von Primzahlen.

Aufgabe H-24: Fügen Sie die Schlüssel 10,22,31,4,15,28,17,88,59 der Reihe nach in eine offen adressierte Hashtabelle der Größe $m = 11$ ein, unter Verwendung der Hashfunktion $h(k) = k \bmod 11$ und Kollisionsauflösung mittels

- a) Linearem Sondieren
- b) Quadratischem Sondieren mit $c_1 = 1, c_2 = 3$
- c) Double Hashing mit $h_1(k) = h(k)$ und $h_2(k) = (k \bmod 10) + 1$.

Dokumentieren Sie die Schritte geeignet. (6 Punkte)

Aufgabe H-25: (6 Punkte)

- a) In einer Hashtabelle mit Kollisionsauflösung durch verkettete Listen und 100000 Elementen werden häufig Elemente gesucht. Für das Hashing wird die *Divisionsmethode* verwendet. Die mittlere Suchdauer soll nicht mehr als durchschnittlich 3 betragen. Bestimmen Sie für diese Anforderungen einen möglichst kleinen Wert für die Größe m der Hashtabelle, wenn (in den meisten Fällen) die Suche nicht erfolgreich ist.
- b) Bestimmen Sie einen möglichst guten Wert für die Größe m im folgenden Szenario: Es wird offene Adressierung eingesetzt, und (in den meisten Fällen) ist die Suche nicht erfolgreich—sonst wie in Teilaufgabe a).

- c) Überlegen Sie, wie groß die mittlere Suchdauer mit Kollisionsauflösung durch verkettete Listen bei *erfolgreicher* Suche in Abhängigkeit von Größe der Tabelle n , Anzahl der Schlüssel $|U|$, und/oder Lastfaktor α ist.

Aufgabe H-26:

- a) In einer Anwendung einer offen adressierten Hashtabelle kann eine erwartete Zahl von 15 Versuchen pro Einfügung noch toleriert werden. Wie hoch darf man die Tabelle auslasten?
- b) Durch Messungen wurde festgestellt, dass die mittlere Suchdauer bei erfolgreicher Suche in einer gewissen offen adressierten Hashtabelle 4,3 beträgt. Geben Sie eine vernünftige Schätzung für den Lastfaktor α dieser Tabelle.

(4 Punkte)

Aufgabe H-27: Ich möchte mit dem Auto von Lissabon nach Tallinn fahren. Mein Tank fasst genügend Benzin, um n Kilometer zu fahren, und ich habe ein Navi, das alle Tankstellen auf der Strecke mit den Entfernungen dazwischen anzeigt. Ich möchte so selten wie möglich zum Tanken anhalten.

Entwerfen Sie einen effizienten Greedy-Algorithmus, der mir angibt, welche Tankstellen ich anfahren sollte, und zeigen Sie, dass Ihr Algorithmus eine minimale Lösung liefert, d.h. mit so wenigen Tankstopps wie möglich.

(6 Punkte)

Abgabe bis Dienstag, 19. Juni, 09.00 Uhr im dafür vorgesehenen Briefkasten in der Resi-39.