Matthias Krause
Alexander Moch

Mannheim, den 26.10.2020

CS 307 Algorithmen und Datenstrukturen, Herbstsemester 2020

Übungsblatt 5

AUFGABE 5.1:

Das Schlüsseluniversum $\mathcal{U}$ bestehe aus Zahlen 720·i, wobei $1 \leq i \leq 10^9$. Die Schlüssel werden mit Gleichverteilung gezogen.
Man gebe Zahlen m_1, m_2 aus dem Bereich $1000 \dots 2000$ an, so dass

a) die Hashfunktion $h_1(k) = k \mod m_1$ die Bedingung des Simple Uniform Hashing (fast) erfüllt.

b) die Hashfunktion $h_2(k) = k \mod m_2$ die Bedingung des Simple Uniform Hashing nicht erfüllt.

AUFGABE 5.2$^{\mathbf{K}}$:

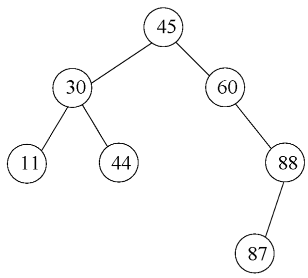
Folgende Schlüssel sollen in eine Hash-Tabelle eingefügt werden: 12, 23, 13, 56, 26, 45, 24, 94, 42.

a) Benutzen Sie zunächst eine Hash-Tabelle mit 13 Behältern und *chained hashing*. Als Hashfunktion soll $h_1(x) := (3x + 1) \mod 13$ benutzt werden. Wie sieht die Hash-Tabelle nach den Einfügeoperationen aus?

b) Wie sähe die Hash-Tabelle aus, wenn offene Adressierung mit Doppelhashing und der zusätzlichen Hashfunktion $h_2(x) := 1 + (x \mod 12)$ benutzt würde?

c) Wieviele Schlüsselvergleiche werden in jedem der beiden Fälle benötigt, wenn der Schlüssel 42 gesucht wird? (Zählen Sie nur die Vergleiche mit 42.)

AUFGABE 5.3: Betrachten Sie den abgebildeten binären Suchbaum.



a) Fügen Sie in den Baum die Elemente mit den Schlüsseln 42, 12, 49, 14, 99, 23 (in dieser Reihenfolge) ein. Wie sieht der Baum nun aus?

b) Löschen Sie aus dem Baum aus Aufgabenteil (a) die Elemente mit den Schlüsseln 44, 42, 45. Wie sieht der Baum nun aus?

AUFGABE 5.4: Angenommen, die Suche nach einem Schlüssel in einem binären Suchbaum endet in einem Blatt. Betrachten Sie die folgenden drei Mengen: P, die Schlüssel auf dem Suchpfad; L, die Schlüssel links von dem Suchpfad; und R, die Schlüssel rechts von dem Suchpfad. Beweisen Sie oder finden Sie ein Gegenbeispiel: Für alle $p \in P$, $l \in L$ und $r \in R$ gilt: $l \leq p \leq r$.

AUFGABE 5.5: Beantworten Sie die folgenden Teilaufgaben, in denen es um die Tiefe von Suchbäumen geht:

a) Man gebe für alle n eine Folge π von n untereinander verschiedener Schlüssel an, die einen binären Suchbaum $T(\Pi)$ der maximal möglichen Tiefe $n-1$ erzeugt. $T(\Pi)$ bezeichnet hier den eindeutig bestimmten Suchbaum, der dadurch entsteht, dass die Schlüssel in der Reihenfolge π eingegeben werden.

b) Für $n = 15$ und $n = 31$ gebe man jeweils eine Folge von n Schlüsseln an, die binäre Suchbäume der jeweils minimal möglichen Tiefe 3 bzw. 4 erzeugen.

c) Für alle $r \geq 1$ gebe man eine (ggf. rekursiv definierte) Folge von $n = 2^r - 1$ Schlüsseln an, die einen binären Suchbaum der minimal möglichen Tiefe $r - 1$ erzeugen.

AUFGABE 5.6$^{\mathbf{K}}$: Geben Sie eine möglichst genaue untere Schranke (in der O-Notation) für die worst-case Laufzeit eines beliebigen vergleichsbasierten Algorithmus an, der eine beliebige Menge von n Zahlen bekommt und einen binären Suchbaum mit diesen Zahlen als Schlüsseln aufbaut. Begründen Sie Ihre Antwort.