

Algorithmen und Datenstrukturen

Blatt 4

Uni Hamburg, Wintersemester 2019/20

Präsentation am 11.–13. Dezember 2019

Jede Teilaufgabe zählt als ein einzelnes Kreuzchen.

Übung 1.

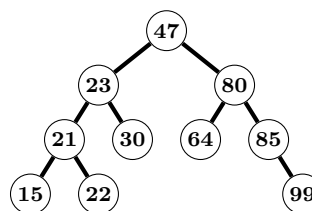
Gegeben ist eine Hashtabelle T der Länge $m = 11$ mit Hashfunktion $h(k; i) = (k + i^2) \bmod m$. Tragen Sie die Schlüssel 56, 46, 31, 45, 42, 65, 29, 44, 23 in der angegebenen Reihenfolge in die Hashtabelle ein. Verwalten Sie Kollisionen:

- (a) durch Verkettung ($i = 0$),
- (b) durch offene Adressierung.
- (c)
 - (i) Nehmen Sie an, dass die Hash-Funktion $h(k)$ Schlüssel zufällig gleichverteilt auf einen Wert in $\{0, 1\}^b$ abbildet, wobei b die Anzahl der Bits ist. Berechnen Sie die Wahrscheinlichkeit, dass keiner von m Schlüsseln einen beliebigen, festen Hash-Wert besitzt.
 - (ii) Ein System hat m Benutzer, die sich durch Eingabe eines Passwort authentifizieren. Nehmen Sie an, dass die m Hash-Werte der von den Benutzern gewählten Passwörtern zufällig und unabhängig voneinander sind. Nutzen Sie das vorherige Ergebnis, um zu ermitteln, wieviele Bits notwendig sind, damit ein Angreifer durch Eingabe eines zufälligen Passworts mit höchstens Wahrscheinlichkeit p Erfolg hat. *Hinweis: Nutzen Sie, dass $(1+x)^n \approx e^{nx}$ für $n > 1, |x| \leq 1$.*

Übung 2.

Betrachten Sie den nebenstehenden binären Suchbaum. Geben Sie an, in welcher Reihenfolge bei Traversieren mit

- (a)
 - (i) Preorder
 - (ii) Inorder
 - (iii) Postorder



die Knoten evaluiert werden.

- (b) Nehmen Sie an, die Suche nach einem Schlüssel k in einem binären Suchbaum endet in einem Blatt. Dann gibt es drei Mengen: A , die Schlüssel links des Suchpfads; B , die Schlüssel auf dem Suchpfad; sowie C , die Schlüssel rechts des Suchpfads. Beweisen oder widerlegen Sie die folgende Behauptung: Für beliebige drei Schlüssel $a \in A$, $b \in B$ und $c \in C$ gilt $a \leq b \leq c$.

Übung 3.

Erweitern Sie einen Baum um eine Funktion `getDepth`, die für einen beliebigen Knoten die Höhe des Teilsbaums (mit diesem Knoten als Wurzel) zurückliefert.

- (a) Beschreiben Sie zwei Möglichkeiten zur Umsetzung der `getDepth` Methode:
 - (i) ohne zusätzlichen Speicher im Baum,
 - (ii) mit zusätzlichem Speicher im Baum.Begründen Sie kurz deren Korrektheit.
- (b) Welche Laufzeit haben die beiden `getDepth` Methoden und welcher zusätzliche Speicherbedarf wird jeweils benötigt?
- (c) Welche Auswirkungen haben beide Alternativen auf die Laufzeiten von Einfügen und Löschen?

Übung 4.

Betrachten Sie den folgenden Rot-Schwarz-Baum.

- (a) Fügen Sie die Werte 13 und 42 ein. Löschen Sie anschließend den Wert 8. Geben Sie den Inhalt des Baums nach jeder Einfüge/Lösch-Aktion an.
- (b) Zeichnen Sie einen validen Rot-Schwarz-Baum mit Schwarz-Höhe 4, der so unbalanciert wie möglich ist.
- (c) Entwickeln Sie basierend auf der vorigen Aufgabe eine Formel für die minimale Anzahl der Knoten in einem maximal unbalancierten Rot-Schwarz-Baum der Höhe h .

