

**Aufgabe 1** Hashing im Selbstversuch II

10 Punkte

- (a) Fügen Sie nacheinander die Schlüssel 10, 22, 31, 4, 15, 28, 17, 88, 59 in eine Hashtabelle der Größe 11 ein. Die Hashfunktion sei  $h(k) = k \bmod 11$ . Die Konflikte werden durch offene Adressierung mit linearem Sondieren gelöst.
- (b) Fügen Sie nacheinander die Schlüssel 10, 22, 31, 4, 15, 29, 17, 88, 59 in eine Hashtabelle der Größe 11 ein. Die Konflikte werden durch Kuckuck gelöst, mit  $h_1(k) = k \bmod 11$  und  $h_2(k) = (k \bmod 13) \bmod 11$ .

Illustrieren Sie jeweils die einzelnen Schritte.

*Hinweis:* Pseudocode für die Hashoperation findet sich im Skript.

**Aufgabe 2** Implementierung einer Hashtabelle

10 Punkte

- (a) Implementieren Sie eine Hashtabelle mit Verkettung in Scala. Benutzen Sie dazu die Funktion `hashCode`, die von allen Objekten in Scala zur Verfügung gestellt wird.  
Gestalten Sie Ihre Implementierung so, dass sich die Größe der Hashtabelle wählen lässt, und implementieren Sie mit mindestens zwei verschiedenen Kompressionsfunktionen.
- (b) Erweitern Sie Ihre Implementierung so, dass die Größe der Hashtabelle dynamisch angepasst wird, um einen Ladefaktor zwischen 1 und 3 zu garantieren (sobald mindestens 20 Einträge in der Hashtabelle vorhanden sind). Welche Strategie wählen Sie, um Ihre Hashtabelle anzupassen?

**Aufgabe 3** Lineares Sondieren und Löschen

10 Punkte

In der Vorlesung haben Sie eine Strategie gesehen, mit der das Löschen in einer Hashtabelle mit linearem Sondieren umgesetzt werden kann: Gelöschte Elemente werden durch einen eigenen Eintrag markiert und in den Einfüge- und Lookup-Routinen speziell behandelt.

Beschreiben Sie eine alternative Methode, bei welcher der gelöschte Eintrag ggf. durch einen geeigneten anderen Eintrag ersetzt wird. Beschreiben Sie Ihre Methode verbal und geben Sie Pseudocode. Geben Sie auch zwei interessante Beispiele, die zeigen, wie Ihre Methode funktioniert.