Albert-Ludwigs-Universität
Institut für Informatik
Prof. Dr. F. Kuhn
M. Fuchs, G. Schmid

Algorithmen und Datenstrukturen
Sommersemester 2024
Korrekturanweisung Übungsblatt 4

Abgabe: Dienstag, 14. Mai, 2024, 10:00 Uhr

Aufgabe 1: Hashing mit offener Adressierung *(5 Punkte)*

Sei $\mathcal{H}$ eine Hashtabelle der Größe $m = 13$ und seinen $h_1, h_2, h_3 : \mathbb{N}_0 \mapsto \{0, ..., m-1\}$ Hashfunktionen definiert wie folgt[1]:

- $h_1(k) := \overline{k} \mod m$
- $h_2(k) := 3 \cdot x \mod m$
- $h_3(k) := x + 1 \mod m$

Fügen Sie die Schlüssel $23, 12, 75, 945, 30, 99, 345$ (in dieser Reihenfolge) in die initial leere Hashtabelle $\mathcal{H}$ ein. Lösen Sie Konflikte wie folgt:

a) Lineares Sondieren unter der Benutzung von h_1. *(2 Punkte)*

b) Doppel-Hashing unter Benutzung von h_2 und h_3.[2] *(3 Punkte)*

Geben Sie den Zustand der Hashtabelle in jedem Schritt an!

Aufgabe 2: Hashing mit Chaining *(5 Punkte)*

Gegeben sei eine Hash-Table der Größe m und eine beliebige Hashfunktion $h : S \mapsto \{0, ..., m-1\}$. Die Menge S habe mindestens $y \cdot m$ Elemente, also $|S| \geq y \cdot m$.

a) Zeigen Sie, dass S mindestens eine Teilmenge Y, bestehend aus mindestens y Elementen (also $|Y| \geq y$), besitzt, so dass $h(x_1) = h(x_2)$ für alle $x_1, x_2 \in Y$. *(4 Punkte)*

b) Was sagt uns das Resultat in *a)* über die Worst-Case Laufzeit von *"find"* in einer Hashtabelle mit Chaining aus (wenn unsere Hashtabelle genau die Elemente aus S speichert bevor *"find"* aufgerufen wird)? *(1 Punkt)*

Aufgabe 3: Anwendung von Hashtabellen *(10 Punkte)*

Gegeben ist folgender Algorithmus:

Algorithm 1 `algorithm` ▷ Input: Array A of length n with integer entries

```
for i = 1 to n − 1 do
    for j = 0 to i − 1 do
        for k = 0 to n − 1 do
            if |A[i] − A[j]| = A[k] then
                return true
return false
```

[1] Wir definieren $\overline{k}$ als die Quersumme von k.

[2] Es soll also $(h_2(k) + i \cdot h_3(k)) \mod m$ als Hashfunktion verwendet werden.

(a) Beschreiben Sie, was `algorithm` berechnet und analysieren Sie die asymptotische Laufzeit. *(3 Punkte)*

Hinweis: Die Differenz $|A[i] - A[j]|$ *kann beliebig große Werte annehmen.*

(b) Beschreiben Sie einen auf *hashing* basierenden alternativen Algorithmus $\mathcal{B}$ für dieses Problem (d.h. $\mathcal{B}(A) =$ `algorithm`(A) für jede Eingabe A) mit einer Laufzeit von $\mathcal{O}(n^2)$ (mit Begründung). *(3 Punkte)*

Hinweis: Sie dürfen annehmen, dass das Einfügen und Finden von Schlüsseln in einer Hashtabelle $\mathcal{O}(1)$ *Zeitschritte benötigt, wenn* $\alpha = \mathcal{O}(1)$ *(*α *ist der Load der Hashtabelle).*

(c) Beschreiben Sie einen weiteren Algorithmus für dieses Problem ohne Verwendung von Hashing mit einer Laufzeit von $\mathcal{O}(n^2 \log n)$ (mit Begründung). *(4 Punkte)*