Lehrstuhl für Informatik I
Algorithmen und Komplexität
Universität Würzburg

Würzburg, den 27. November 2025

Prof. Dr. Alexander Wolff
Samuel Wolf, M. Sc.

6. Übungsblatt zur Vorlesung Algorithmen und Datenstrukturen (Winter 2025/26)

Aufgabe 1 – Hashing

Die Schlüssel $44, 12, 23, 88, 71, 11, 94, 39, 20, 5$ und 16 sollen in dieser Reihenfolge nacheinander in eine Hashtabelle $T[0..15]$ eingefügt werden. Dabei können verschiedene Verfahren eingesetzt werden, um Kollisionen aufzulösen.

a) Zeichnen Sie für jedes der folgenden Verfahren die resultierende Hashtabelle.

   1. Kollisionen werden durch Verkettung aufgelöst;
      die Hashfunktion ist $h(k) = (3k + 7) \bmod 16$.
   2. Kollisionen werden durch lineares Sondieren aufgelöst;
      die Hashfunktion ist $h(k, i) = (h_0(k) + i) \bmod 16$
      mit $h_0(k) = (3k + 7) \bmod 16$.
   3. Kollisionen werden durch quadratisches Sondieren aufgelöst;
      die Hashfunktion ist $h(k, i) = (h_0(k) + c_1 i + c_2 i^2) \bmod 16$
      mit $c_1 = \frac{1}{2}$, $c_2 = \frac{1}{2}$ und $h_0(k) = (3k + 7) \bmod 16$.
   4. Kollisionen werden durch doppeltes Hashing aufgelöst;
      die Hashfunktionen ist $h(k, i) = (h_1(k) + i h_2(k)) \bmod 16$
      mit $h_1(k) = (3k + 7) \bmod 16$ und $h_2(k) = 7 - 2(k \bmod 4)$.

   Bei den Verfahren 2. bis 4. durchläuft i die Werte $0, \ldots, 15$.

   Geben Sie bei jedem Verfahren (außer bei 1.) und für jeden Schlüssel an, wie viele Zellen Sie testen mussten, bevor Sie eine freie Zelle gefunden haben. Geben Sie für jedes Verfahren auch die Gesamtzahl der erfolglosen Tests an. **6 Punkte**

b) Welches Problem tritt beim doppelten Hashing auf, wenn die Hashfunktion
   $h(k, i) = (h_1(k) + i h_2(k)) \bmod 16$
   mit $h_1(k) = (3k + 7) \bmod 16$ und $h_2(k) = 8 - (k \bmod 8)$ verwendet wird?
   1 Punkt

c) Beim quadratischen Sondieren mit $h(k, i) = (h_0(k) + c_1 \cdot i + c_2 \cdot i^2) \bmod m$ kann es durch ungeschickte Wahl von c_1 und c_2 zu dem Problem kommen, dass $\{h(k, 0), \ldots, h(k, m-1)\} \subsetneq \{0, \ldots, m-1\}$. Geben Sie ein Beispiel mit $c_1, c_2 \in \mathbb{N} \setminus \{0\}$ an. **1 Punkt**

Aufgabe 2 – BinarySearch

Im Folgenden sei A stets ein aufsteigend sortiertes Feld der Länge n.

a) Betrachten Sie folgenden Algorithmus für binäre Suche.

```
boolean BinarySearch(key[] A, key k, int ℓ = 1, int r = A.length)
if ℓ > r then
    return false
m = ⌊(ℓ + r)/2⌋
if A[m] == k then
    return true
if A[m] > k then
    return BinarySearch(A, k, ℓ, m − 1)
else
    return BinarySearch(A, k, m + 1, r)
```

Geben Sie die genaue Anzahl der Vergleiche mit Elementen des Eingabefeldes an, die BinarySearch im Worst-Case benötigt. Sie dürfen dabei annehmen, dass die Länge des Feldes eine Zweierpotenz ist (also $n = 2^i$ und $i \in \mathbb{N}$). **2 Punkte**

b) Geben Sie einen Algorithmus BinarySearch2(key[] A, key k) an, der das gleiche Problem löst, jedoch höchstens $\lceil \log_2 n \rceil + 1$ Vergleiche mit Elementen des Eingabefeldes benötigt. Sie dürfen nur Hilfsvariablen vom Typ int verwenden (keine vom Type key). Begründen Sie, warum Ihr Algorithmus tatsächlich nur die geforderte Anzahl von Vergleichen ausführt. **5 Punkte**

Aufgabe 3 – Spezialsuche

Gegeben sei ein Feld $A[1..k]$ mit ganzen Zahlen, für die $A[1] < A[2] < \cdots < A[k]$ gilt. Geben Sie in Worten und im Pseudocode einen Algorithmus an, der ermittelt, ob es eine Zahl $j \in \{1, \ldots, k\}$ mit $A[j] = j$ gibt. Die Worst-Case Laufzeit Ihres Algorithmus soll $\Theta(\log k)$ sein. **5 Punkte**

Hinweis: Finden Sie ein geeignetes, leicht zu berechnendes Kriterium, um damit den gesuchten Index zu suchen. Bedenken Sie auch die besondere Struktur von A.

Bitte geben Sie Ihre Lösungen bis **Donnerstag, 4. Dezember 2025, 14:00 Uhr** einmal pro Gruppe über Wuecampus als pdf-Datei ab. Vermerken Sie dabei stets die Namen und Übungsgruppen aller BearbeiterInnen auf der Abgabe.

Grundsätzlich sind stets alle Ihrer Aussagen zu begründen und Ihr Pseudocode ist stets zu kommentieren.

Die Lösungen zu den mit PABS gekennzeichneten Aufgaben, geben Sie bitte nur über das PABS-System ab. Vermerken Sie auf Ihrem Übungsblatt, in welchem Repository

(*sXXXXXX*-Nummer) die Abgabe zu finden ist. Geben Sie Ihre Namen hier als Kommentare in den Quelltextdateien an.