

# Übungsblatt 9

Abgabe bis 28.06.2018  
Besprechung: 02.07.2018 – 05.07.2018

## Aufgabe 1: Dynamisches Programmieren (6 + 2 Punkte)

Seien  $u, v, w \in \Sigma^*$  Wörter über einem Alphabet  $\Sigma$ . Wir nennen  $w$  *einen Shuffle* aus  $u$  und  $v$ , wenn es eine Zerlegung  $u = u_1 \dots u_k$  und  $v = v_1 \dots v_l$  gibt mit  $u_i, v_j \in \Sigma^*$  für alle  $i, j$ , so dass  $w = u_1 v_1 u_2 v_2 \dots$  ist.

Zum Beispiel ist  $w = \text{INaFOlgoRMrithAmTlenK}$  ein Shuffle aus den Wörtern  $u = \text{INFORMATIK}$  und  $v = \text{algorithmen}$ .

- (a) Geben Sie einen Algorithmus an, der für drei gegebene Wörter  $u, v, w$  unter der Verwendung von dynamischer Programmierung entscheidet, ob  $w$  ein Shuffle aus  $u$  und  $v$  ist.
- (b) Geben Sie die Laufzeit in Abhängigkeit der Längen der Wörter an.

## Aufgabe 2: Programmieraufgabe (6 + (opt. 6) Punkte)

Laden Sie die Java-Vorlage aus dem Moodle herunter und implementieren Sie die folgenden Methoden:

- (a) `isShuffled(String w, String u, String v)`, die entscheidet ob das Wort  $w$  ein Shuffle aus den zwei Wörtern  $u$  und  $v$  ist.
- (b) `isShuffled(String w, String u, String v, String z)`, die entscheidet ob das Wort  $w$  ein Shuffle aus den drei Wörtern  $u, v$  und  $z$  ist.

Verwenden Sie bei Ihrer Implementierung sinnvolle Variablennamen und kommentieren Sie Ihren Code! Laden Sie Ihre Lösung ins Moodle. Nicht kompilierende Abgaben werden **mit 0 Punkten** bewertet.

## Aufgabe 3: Anagramme (2 + 2 + 2 + 2 Punkte)

Seien  $u$  und  $v$  zwei Wörter über einem endlichen Alphabet  $\Sigma$ . Ein Anagramm eines Wortes  $u$  ist ein Wort  $v$ , das aus einer Permutation der Buchstaben von  $u$  gebildet werden kann.

- (a) Welche Eigenschaften muss eine Hashfunktion erfüllen, um Anagramme schnell identifizieren zu können?
- (b) Die Funktion `getBytes` weist jedem Buchstaben aus  $\Sigma$  eine eindeutige natürliche Zahl zu. Betrachte folgende Hashfunktion  $h: \Sigma^* \rightarrow \mathbb{N}$

$$h(w_1 \dots w_n) = \sum_{i=1}^n \text{getBytes}(w_i)$$

Zeigen Sie, dass die Hashfunktion  $h$  zwei Anagramme auf denselben Hashwert abbildet.

- (c) Warum ist die Funktion dennoch ungeeignet, um Anagramme zu filtern? Welche Strings werden noch auf dieselben Hashwerte abgebildet?
- (d) Modifizieren Sie die Hashfunktion so, dass sie sich besser dazu eignet Anagramme zu filtern. Begründen Sie, warum sich Ihre neue Hashfunktion besser verhalten sollte und zeigen Sie, dass sie immer noch zwei Anagramme auf denselben Wert abbildet.

#### Aufgabe 4: Hashing (2 + 2 + 2 + 2 Punkte)

Betrachten Sie die folgende Hashfunktion

$$\begin{aligned} h_{a,b,m}: & \rightarrow \{0, \dots, m-1\} \\ x & \mapsto (ax + b) \bmod m \end{aligned}$$

und die Eingabewerte

$$S = \{13, 45, 64, 78, 116\} .$$

- (a) Geben Sie Werte für  $a$ ,  $b$  und  $m$  an, sodass
- alle Werte aus  $S$  auf den selben Hashwert abgebildet werden.
  - alle Werte aus  $S$  auf unterschiedliche Hashwerte abgebildet werden.
- (b) Sei  $g_i(x) := (h_{1,0,5}(x) + i) \bmod 5$ . Geben Sie jeweils die Hashtabelle an, die entsteht, wenn  $S$  in aufsteigender Reihenfolge in eine zuvor leere Hashtabelle eingefügt wird. Verwenden Sie dabei die folgenden Kollisionsauflösungsstrategien:
- Verkettetes Hashing mit der Hashfunktion  $g_0$
  - Hashing mit offener Adressierung mit  $(g_i)_{i \geq 0}$
- (c) Zeigen Sie, dass es für alle Werte von  $a$  und  $b$  bei  $m = 5$  zu Kollisionen kommt.
- (d) Geben Sie Werte für  $a$ ,  $b$  und  $m$  an, sodass  $m$  minimal ist und alle Werte aus  $S$  auf unterschiedliche Hashwerte abgebildet werden.

---

**Am 29.06. findet auf dem Sand ab 10:00 Uhr der Alumnitag statt. Dort gibt es kostenloses Essen und spannende Vorträge.**

**Im Anschluss geht es direkt weiter: Ab 17:00 Uhr beginnt das Sommerfest der Fachschaft Informatik. Es gibt wie immer Pizza, Volleyball, eine brandheiße Feuershow und gute Laune.**