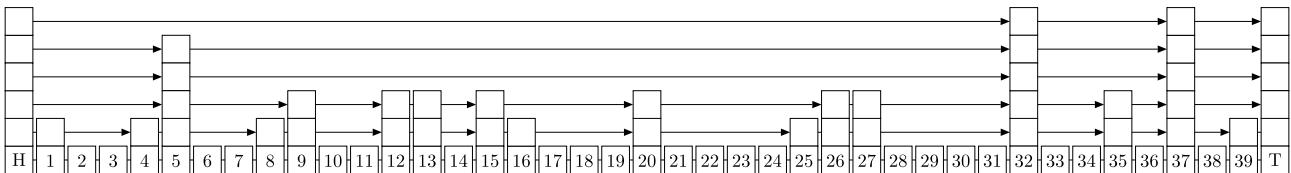


## Übungsblatt 04

### Aufgabe T11

Geben sie jeden Schritt im Detail an, wenn wir in folgender Skiplist nach 22, 40 und 0 suchen.



### Aufgabe T12

Wir verwenden jetzt folgende universelle Familie von Hashfunktionen:

$$\mathcal{H}_{T12} = \{ h_{a,b} \mid 1 \leq a < 5, 0 \leq b < 5 \}$$

mit

$$h_{a,b}(x) = ((ax + b) \bmod 5) \bmod 4.$$

Berechnen Sie die Wahrscheinlichkeit, dass  $h(2) = h(3)$  ist, falls wir  $h$  zufällig aus obiger Familie von Funktionen wählen. Was müsste herauskommen, wenn man bedenkt, dass es sich um eine universelle Familie von Hashfunktionen handelt?

Da es sich um viele Funktionen handelt, sollte die Arbeit unter vielen Personen verteilt werden, damit es schneller geht. Wie lange brauchen Sie gemeinsam, um diese Wahrscheinlichkeit ohne Taschenrechner oder ähnlichem zu berechnen?

### Aufgabe T13

Wir betrachten die Hashfunktion  $h: \Sigma^* \rightarrow \{0, \dots, m-1\}$ ,  $c_1 c_2 \dots c_k \mapsto \left( \sum_{i=1}^k c_i \right) \bmod m$ , welche die Menge  $\Sigma^*$  aller Wörter (im ASCII-Alphabet) auf eine Zahl zwischen 0 und  $m - 1$  abbildet.

- Ist  $h$  eine gute Hashfunktion? Überlegen Sie sich realistische Anwendungsbeispiele, für welche  $h$  sehr viele Kollisionen erzeugt.
- Wie könnte eine vernünftige Hashfunktion für Zeichenketten aussehen, für welche Sie unter normalen Umständen ein gutes Verhalten erwarten würden? Wie gehen Sie mit dem Fall um, dass  $m$  sehr groß ist?

### Aufgabe T14

Ist die folgende Behauptung wahr? Wenn nein, dann finden Sie ein Gegenbeispiel und den Fehler im angeblichen Beweis.

*Behauptung:* Für alle  $x \in U$ , alle universelle Familien  $\mathcal{H}$  von Hashfunktionen  $U \rightarrow \{1, \dots, m\}$  und alle  $k \in \{1, \dots, m\}$  gilt:  $\Pr[h(x) = k] = 1/m$ , falls  $h$  zufällig aus  $\mathcal{H}$  gewählt wurde.

*Beweis:* Nehmen wir an, es gibt ein  $y$  mit  $h(y) = k$ . Dann setze  $S = \{y\}$  und wende das letzte Theorem an:

$$E(|\{y \in S \mid h(x) = h(y)\}|) \leq \frac{|S|}{m}.$$

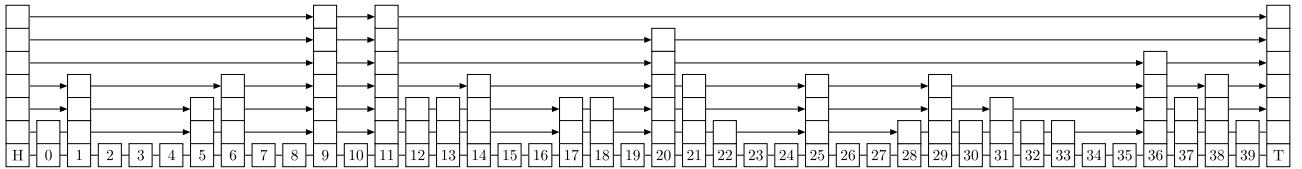
Hier folgt daraus  $\Pr[h(x) = k] \leq 1/m$ .

Wenn es kein  $y$  mit  $h(y) = k$  gäbe, dann wäre  $\Pr[h(x) = k] = 0$ .

Das geht aber nicht, denn  $\sum_{k=1}^m \Pr[h(x) = k] = 1$ .

### Aufgabe H12 (6 Punkte)

Geben sie jeden Schritt im Detail an, wenn wir in folgender SkipList nach 10, 19 und 93 suchen.



### Aufgabe H13 (4+1+3 Punkte)

- Erstellen Sie mithilfe eines Programms eine Tabelle, welche die Wahrscheinlichkeiten von  $h(x) = h(y)$  für alle  $0 \leq x, y < 5$  enthält, falls  $h$  wieder zufällig aus der Familie  $\mathcal{H}_{T12}$  von T12 gezogen wird.
- Ist das Ergebnis das, was Sie erwarten? Warum?
- Wiederholen Sie das Experiment für  $0 \leq x, y < 6$ ,  $1 \leq a < 6$ ,  $0 \leq b < 6$  und  $h_{a,b}(x) = ((ax + b) \bmod 6) \bmod 4$ . Kommentieren Sie das Ergebnis. Nehmen Sie insbesondere dazu Stellung, ob es sich auch jetzt um eine universelle Familie von Hashfunktionen handelt.

### Aufgabe H14 (7+7 Punkte)

Im April 2020 mussten das Studium plötzlich komplett von zuhause aus gemacht werden. So auch FoSAP, welches wir damals gehalten haben; Prof. Rossmannith hielt die Vorlesung über Zoom und Daniel hat den Chat betreut. Dies war auch zwingend nötig, da sich einige Studis in ihrer Pseudonymität schnell zu wohl gefühlt haben. Da die Moderationsmöglichkeiten in Zoom ziemlich beschränkt waren, schlägt Daniel vor, so einen Chatraum selber zu realisieren.

In einem Chatraum gibt es eine feste Menge von Studis  $S$ , die alle durch eine eindeutige IntegerID  $s$  identifiziert sind. Jeder Studi  $s$  kann Nachrichten in den Chatraum schicken. Jeder Studi kann die letzten  $k$  Nachrichten lesen, wobei  $k$  ein selbstgewählter Wert ist. Sollte Daniel das Verhalten eines Studis nicht gefallen, so kann er diesen bannen; d.h. es werden alle Nachrichten von diesem Studi gelöscht und er kann keine neuen senden.<sup>1</sup> Genauer gesagt, gibt es folgende vier Operationen, die es umzusetzen gilt:

- $erstellen(S)$ : Initialisieren des Chatraumes mit  $n = |S|$  Studis.
- $senden(s, m)$ : Sende Nachricht  $m$  zum Chat von Studi  $s$  (außer er ist gebannt).
- $letzte(k)$ : Gib die letzten  $k$  (oder alle, falls es weniger als  $k$  Nachrichten gibt) nicht-gelöschten Nachrichten aus.
- $ban(s)$ : Banne Studi  $s$  und lösche alle seine Nachrichten.

Random-Access ist hier nicht nötig; der Chat ist lediglich ein Strom von affektiven Reaktionen ohne Gedächtnis.

Geben Sie für die folgenden Teilaufgaben die Datenstrukturen an und wie die einzelnen Operationen mittels der Datenstrukturen realisiert werden. Erläutern Sie, wie die Laufzeiten zustande kommen.

- Entwickeln Sie Datenstrukturen, die die obigen Operationen unterstützt, mit folgenden worst-case Laufzeiten:

<sup>1</sup>Inspiriert durch eine Aufgabe aus der Vorlesung 6.006 „Intro to Algorithms“ vom MIT.

- $\text{erstellen}(S)$  in  $O(n \log n)$  Zeit,
- $\text{senden}(s, m)$  in  $O(\log n)$  Zeit,
- $\text{letzte}(k)$  in  $O(k)$  Zeit,
- $\text{ban}(s)$  in  $O(n_s + \log n)$  Zeit, wobei  $n_s$  die Anzahl der Nachrichten von Studi  $s$  ist.

(b) Ihnen fällt auf, dass die Vorlesungen FoSAP bzw. DSAL so beliebt sind, dass die obigen Laufzeiten zu langsam sind. Entwerfen Sie Datenstrukturen mit folgenden *amortisierten worst-case Laufzeiten im Erwartungswert*:

- $\text{erstellen}(S)$  in  $O(1)$  Zeit,
- $\text{senden}(s, m)$  in  $O(1)$  Zeit,
- $\text{letzte}(k)$  in  $O(k)$  Zeit,
- $\text{ban}(s)$  in  $O(1)$  Zeit.

*Hinweis:* Das „im Erwartungswert“ bezieht sich auf die zufällige Wahl der Funktion aus einer universellen Familie von Hashfunktionen; worst-case bezieht sich auf den Input.

*Hinweis:* Kombinieren Sie aus der Vorlesung bekannte Datenstrukturen auf geschickte Weise. Außerdem ist explizit nach keinem Programm oder Pseudocode gefragt, sondern nach einer prosaischen Beschreibung.

### Aufgabe H15 (3+6+3 Punkte)

Am Ende dieser Aufgabe und in Moodle finden Sie das Programm `WordCount13.java`, das eine Datei lesen kann, welche je Zeile ein Wort enthalten soll. Das Programm zählt, wie viele verschiedene Wörter es in der Datei gibt, wobei es eine Hashtabelle verwendet.

- a) Wenn Sie das Programm auf einer typischen Datei laufen lassen, wie lang ist die Laufzeit auf Ihrem Computer typischerweise für eine Datei mit  $n$  Bytes?
- b) Erzeugen Sie eine Eingabedatei, deren Größe 1MB nicht überschreitet und deren Bearbeitung länger als 5 Sekunden dauert. Laden Sie diese Datei mit in Moodle hoch. Erklären Sie, wie Sie die Datei erstellt haben.
- c) Könnten Sie solch eine furchtbare Datei auch dann *leicht* erstellen, wenn im Programm  $h$  nicht mit  $x$  sondern mit  $y$  multipliziert würde? (Eine sehr kurze Antwort reicht hier.)

```

import java.io.IOException;
import java.nio.file.Files;
import java.nio.file.Paths;
import java.util.*;

public class WordCount13 {
    static int x = 13;
    static int y = Math.abs((new Random()).nextInt());

    public static int hash(String w) {
        int h = 0;
        for(int i = 0; i < w.length(); i++) {
            h = h * x + w.charAt(i);
        }
        return Math.abs(h);
    }

    public static void main(String args[]) throws IOException {
        String filename = args[0];
        List<String> words = Files.readAllLines(Paths.get(filename));
        int m = 100000;
        long count = 0;
        ArrayList<LinkedList<String>> slot = new ArrayList<>();
        for(int i = 0; i < m; i++) {
            slot.add(new LinkedList<String>());
        }
        for(String w : words) {
            LinkedList<String> list = slot.get(hash(w)%m);
            if(!list.contains(w)) {
                count = count + 1;
                list.add(w);
            }
        }
        System.out.println(count + " different words in " + filename + ".");
    }
}

```