

## Übungsblatt 5

Abgabe bis Dienstag, den 28. Mai um 16:00 Uhr

### Aufgabe 1 (10 Punkte)

Bei dieser Aufgabe geht es darum, den Begriff der *Universalität* einer Klasse von Hash-Funktionen zu verstehen und diese empirisch nachzuprüfen. Einen Design-Vorschlag + Tipps für eine Klasse *UniversalityCheck* finden Sie auf dem Wiki.

Sei  $\mathcal{H}$  eine beliebige Klasse von Hash-Funktionen. Für ein Schlüsselpaar  $x, y$  wollen wir die Kollisions-Wahrscheinlichkeit  $\Pr(h(x) = h(y))$  abschätzen, wenn  $h$  zufällig aus  $\mathcal{H}$  gewählt wird. Für ein gegebenes  $N$  berechnen wir dazu  $\hat{p}(x, y, N) = |\{h_i : h_i(x) = h_i(y)\}|/N$ , wobei  $h_1, \dots, h_N$  zufällig und unabhängig voneinander aus  $\mathcal{H}$  gewählt sind. Dann ist  $\lim_{N \rightarrow \infty} \hat{p}(x, y, N) = \Pr(h(x) = h(y))$ .

Ziel der Aufgabe ist es nun, für jede der folgenden drei Klassen von Hash-Funktionen ein Histogramm (siehe dazu Erklärung in der Vorlesung) der  $u \cdot (u-1)$  geschätzten Kollisions-Wahrscheinlichkeiten  $\hat{p}(x, y, N)$  für alle  $x, y \in U$  mit  $x \neq y$ , zu erstellen. Wählen Sie dabei für alle drei Klassen  $m = 16$ ,  $u = 256$  und  $N = 1000$ .

- (1) Die Menge  $\mathcal{H}_1$  aller Funktionen  $h(x) = (a \cdot x + b) \bmod m$ , wobei  $a, b \in \{0, \dots, u-1\}$ .
- (2) Die Menge  $\mathcal{H}_2$  aller Funktionen  $h(x) = (a \cdot x + b) \bmod p \bmod m$ , wobei  $a, b \in \{0, \dots, p-1\}$  und  $p$  eine Primzahl  $\geq u$  ist.
- (3) Die Menge  $\mathcal{H}_3$  aller Funktionen  $h(x) = a \cdot x \bmod 2^k \operatorname{div} 2^{k-l}$ , wobei  $a \in \{1, 3, \dots, 2^k - 1\}$  (nur die *ungeraden* Zahlen aus diesem Bereich),  $k = \lceil \log_2 u \rceil$  und  $l = \lfloor \log_2 m \rfloor$ .

Diskutieren Sie die drei Histogramme kurz in Ihren *erfahrungen.txt*. Insbesondere: was sagen das jeweilige Histogramm über die Universalität der jeweiligen Klasse und warum?

### Aufgabe 2 (10 Punkte)

Implementieren Sie das in der Vorlesung am Mittwoch erklärte Cuckoo Hashing, mit Funktionen für *insert* und *lookup* und *rehash*. Einen Design-Vorschlag + Tipps für eine Klasse *CuckooHashing* finden Sie auf dem Wiki. Verwenden Sie für die beiden Hash-Funktionen eine der drei universellen Klassen aus der Vorlesung.

Committen Sie Ihre Lösung (Code + Histogramme) in unser SVN, in einen neuen Unterordner *uebungsblatt\_05*. Schreiben Sie für jede nicht-triviale Methode einen Unit Test, und stellen Sie sicher, dass auf Jenkins alles (insbesondere *checkstyle*) fehlerfrei durchläuft.

Committen Sie in diesem neuen Unterordner außerdem wie beim letzten Mal eine Textdatei *erfahrungen.txt*. Beschreiben Sie dort in ein paar Sätzen Ihre Erfahrungen mit diesem Übungsblatt und der Vorlesung dazu. Insbesondere: Wie lange haben Sie ungefähr gebraucht? An welchen Stellen gab es Probleme und wieviel Zeit hat Sie das gekostet? Wie wird das Wetter übernächsten Dienstag und warum?