Aufgabe 1

$H = \{h_1, h_2, h_3, h_4, h_5, h_6\}$ ist eine universelle Familie von Hashfunktionen, wenn jedes Paar von unterschiedlichen Elementen aus U bei maximal 2 Hashfunktionen kollidiert (da $|H| = 6$ und $|T| = 3$).

Die Hashfunktionen h_1 bis h_5 haben die folgenden kollidierenden Paare:

- h_1: $(0,3), (1,4), (2,5)$
- h_2: $(0,3), (1,4), (2,5)$
- h_3: $(0,2), (0,4), (2,4), (1,3), (1,5), (3,5)$
- h_4: $(0,5), (1,2), (3,4)$
- h_5: $(0,1), (2,3), (4,5)$

Die einzigen kollidierenden Paare, die bereits zweimal auftauchen sind $(0,3), (1,4), (2,5)$. Solange h_6 diese kollidierenden Paare nicht enthält taucht jedes Paar maximal 2 mal auf. Man kann also z.B. $h_6 = h_5$ wählen.

Aufgabe 2

In dieser Aufgabe beschäftigt man sich mit $pseudo-universellen$ Familien von Hashfunktionen.

Eine Menge $\mathcal{H}$ von Funktionen $h : U \to \{0, 1, \ldots, t-1\}$ heißt $pseudo-universell$, wenn für jedes $x \in U$ und jedes $i \in \{0, 1, \ldots, t-1\}$ gilt

$$\frac{|\ \{h \in \mathcal{H} \mid h(x) = i\}\ |}{|\mathcal{H}|} \leq \frac{1}{t}.$$

Es sei jetzt $S \subset U$ mit $|S| = n$.

a) Zu zeigen ist, daß bei zufälliger Wahl einer Hashfunktion h aus einer pseudo-universellen Familie $\mathcal{H}$ für jedes $i \in \{0, 1, \ldots, t-1\}$ die *erwartete* Anzahl von Elementen von S, die durch h auf i abgebildet werden, höchstens $\frac{n}{t}$ ist.

   Dazu definiert man sich zunächst analog zur Vorlesung eine Funktion C_i, die die Stücke von S, die durch h auf i abgebildet werden, zurückliefert:

$$C_i = \{x \in S \mid h(x) = i\}.$$

Der für uns interessante Wert ist also der Erwartungswert von $|C_i|$. Um diesen leichter berechnen zu können, benötigt man noch die passende Indikatorvariable δ_{ix}:

$$\delta_{ix} = \begin{cases} 1, & h(x) = i \\ 0, & \text{sonst} \end{cases}$$

Somit läßt sich der Erwartungswert jetzt analog zur Vorlesung berechnen:

$$E[|C_i|] = \sum_{x \in S} E[\delta_{ix}] = \sum_{x \in S} Pr(\delta_{ix} = 1) = \sum_{x \in S} \underbrace{\frac{|\ \{h \in \mathcal{H} \mid h(x) = i\}\ |}{|\mathcal{H}|}}_{\leq \frac{1}{t}, da\ pseudo-univ.} \leq \frac{n}{t}.$$

b) Gesucht ist ein Beispiel für eine Hashfamilie $\mathcal{H}$, die zwar pseudo-universell ist, aber trotzdem für die Operationen `SEARCH` und `DELETE` wesentlich mehr als $\Omega(\frac{n}{t})$ Operationen benötigt.

Sei $\mathcal{H} = \{h_k \mid h_k(x) = k\ , 0 \leq k < t,\ \forall x \in U\}$. $\mathcal{H}$ ist pseudo-universell, denn es gilt:

- $|\mathcal{H}| = t$, da $0 \leq k < t$.
- $|\{h \in \mathcal{H} | h(x) = i\}| \leq 1$, da jeweils genau eine Funktion h_i alle Elemente auf i abbildet.

Daraus folgt

$$\frac{|\ \{h \in \mathcal{H} \mid h(x) = i\}\ |}{|\mathcal{H}|} \leq \frac{1}{t}.$$

Der Nachteil dieser Familie von Hashfunktionen ist leicht zu erkennen. Da jede Hashfunktion konstant ist, werden die Elemente bei Wahl einer Hashfunktion h_k alle auf k abgebildet. An Stelle k werden alle n Elemente also als Liste verwaltet. Die Anzahl von Operationen, die zum Suchen, bzw. Löschen eines Elementes benötigt werden ist also $\Omega(n)$ und somit deutlich höher als $\Omega(\frac{n}{t})$.

Aufgabe 3

Für alle $a, b \in U, a \neq b$ gilt:

$$
\begin{aligned}
&|\{h \in \mathcal{H} | h(a) = h(b)\}| = \\
&|\{h \in \mathcal{H}_0 \times \mathcal{H}_1 | h(a) = h(b)\}| = \\
&|\{(h_0, h_1) \in \mathcal{H}_0 \times \mathcal{H}_1 | h_0(a) = h_0(b) \wedge h_1(a) = h_1(b)\}| = \\
&\sum_{h_0 \in \mathcal{H}_0} \sum_{h_1 \in \mathcal{H}_1} \langle h_0(a) = h_0(b) \wedge h_1(a) = h_1(b) \rangle = \\
&\sum_{h_0 \in \mathcal{H}_0} \langle h_0(a) = h_0(b) \rangle \cdot \sum_{h_1 \in \mathcal{H}_1} \langle h_1(a) = h_1(b) \rangle = \\
&|\{h_0 \in \mathcal{H}_0 | h_0(a) = h_0(b)\}| \cdot |\{h_1 \in \mathcal{H}_1 | h_1(a) = h_1(b)\}| \\
&\text{Da } \mathcal{H}_0 \text{ und } \mathcal{H}_1 \text{ universell sind} \\
&= \frac{|\mathcal{H}_0|}{t_0} \cdot \frac{|\mathcal{H}_1|}{t_1} \\
&= \frac{|\mathcal{H}_0| \cdot |\mathcal{H}_1|}{t_0 \cdot t_1} \\
&= \frac{|\mathcal{H}_0 \times \mathcal{H}_1|}{t} \\
&= \frac{|\mathcal{H}|}{t}
\end{aligned}
$$

Aufgabe 4

(a) Für alle $x, y \in \{0,1\}^d, x \neq y$ bestimmen wir die Anzahl der $a \in \{0,1\}^d$ für die $h_a(x) = h_a(y)$. Da $x \neq y$ existiert eine Stelle $i \in [d] = \{1, 2, \ldots d\}$ für die $x_i \neq y_i$. Für jedes a bestimmen wir

$$\bar{x} = \bigoplus_{j \in [d] \setminus \{i\}} a_j \wedge x_j, \bar{y} = \bigoplus_{j \in [d] \setminus \{i\}} a_j \wedge y_j$$

Es gilt $h_a(x) = \bar{x} \oplus (a_i \wedge x_i)$ und $h_a(y) = \bar{y} \oplus (a_i \wedge y_i)$.

Fallunterscheidung:

Fall $\bar{x} = \bar{y}$: $h_a(x) = h_a(y)$ dann und genau dann wenn $a_i = 0$.

Fall $\bar{x} \neq \bar{y}$: $h_a(x) = h_a(y)$ dann und genau dann wenn $a_i = 1$

In beiden Fällen entscheidet also das Bit a_i und für exakt die Hälfte aller $a \in \{0,1\}^d$ gilt $h_a(x) = h_a(y)$, womit $\mathcal{H}$ universell ist.

(b) Wir verwenden die Konstruktion aus Aufgabe 3. und nutzen das s-fache kartesische Produkt von $\mathcal{H}$ mit sich selbst.