Aufgabe 3.1 (4 Punkte)
Wir betrachten die in der Vorlesung vorgestellte 1-universelle Klasse H^{SP} mit $m = 13$ und $k = 4$. Gegeben seien zudem die Einträge $e_1, \ldots, e_9$ mitsamt ihrer Schlüssel, die in der nachfolgenden Tabelle aufgeführt sind.

e_i	key(e_i)
e_1	(2,4,5,1)
e_2	(1,2,1,3)
e_3	(5,2,0,7)
e_4	(2,0,3,1)
e_5	(1,0,0,2)
e_6	(6,3,1,0)
e_7	(1,2,1,3)
e_8	(0,0,6,0)
e_9	(0,4,0,4)

a) Berechnen Sie den Parameter w und geben Sie die Mengen an, aus denen die Zahlentupel a und die Schlüssel x jeweils gewählt werden dürfen.

b) Betrachten Sie nun die Hashfunktion $h_{\boldsymbol{a}} \in H^{SP}$ mit $\boldsymbol{a} = (5, 2, 9, 1)$. Fügen Sie die Einträge $e_1, \ldots, e_9$ in der gegebenen Reihenfolge mittels Hashing mit Verkettung in eine Hashtabelle geeigneter Länge ein. Berechnen Sie alle dazu benötigten Hashwerte. Stellen Sie die entstehende Hashtabelle mitsamt ihrer verketteten Listen graphisch dar.

c) Nennen Sie unter den neun gegebenen Einträgen diejenigen, für die die Suche maximale Zeit benötigt. Begründen Sie kurz Ihre Antwort.

Aufgabe 3.2 (4 Punkte)
Für diese Aufgabe wird angenommen, dass die Schlüssel Bitstrings der Länge k sind, d.h. Key $= \{0,1\}^k$, und dass die Tabellengröße m eine Zweierpotenz ist, etwa $m = 2^w$. Jede $w \times k$-Matrix M mit Einträgen aus $\{0, 1\}$ definiert auf folgende Weise eine Hashfunktion h_M: Für $x \in \{0,1\}^k$ setze

$$h_M(x) = Mx \bmod 2,$$

d.h., $h_M(x)$ ist ein Matrix-Vektor-Produkt, berechnet modulo 2. Der resultierende w-Bit-Vektor wird als Zahl in $\{0, \ldots, m-1\}$ interpretiert. Wir definieren dann die folgende 1-universelle Klasse:

$$H^{\text{lin}} = \{h_M \mid M \in \{0,1\}^{w \times k}\}.$$

a) Betrachten Sie die Hashfunktion $h_{\boldsymbol{M}} \in H^{\text{lin}}$ mit

$$\boldsymbol{M} = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{pmatrix}.$$

Fügen Sie die Einträge $e_1, \ldots, e_7$ mit den in der nachfolgenden Tabelle aufgeführten Schlüsseln in eine Hashtabelle geeigneter Länge mittels Hashing mit linearem Sondieren ein. Berechnen Sie alle dazu benötigten Hashwerte. Stellen Sie die entstehende Hashtabelle graphisch dar.

e_i	$\text{key}(e_i)^T$
e_1	(0,0,0,0)
e_2	(0,0,1,0)
e_3	(0,1,1,1)
e_4	(1,0,1,0)
e_5	(1,0,0,1)
e_6	(0,1,1,0)
e_7	(1,0,0,1)

b) In der Vorlesung wurden für das Hashing mit linearem Sondieren zwei Alternativen zur Durchführung des Löschens von Einträgen besprochen (Markierung als gelöscht oder Verschieben von Einträgen). Geben Sie für beide Möglichkeiten jeweils die Hashtabelle an, die aus dem Löschen des Eintrags e_4 aus der aus Teilaufgabe a) resultierenden Hashtabelle entsteht.

Aufgabe 3.3 (4 Punkte)
In dieser Aufgabe betrachten wir perfektes Hashing. Sei H^{SP} die in der Vorlesung vorgestellte 1-universelle Klasse von Hash-Funktionen mit $m = 7$ und $k = 2$. Sei $S = \{(0,3), (2,2), (2,3)\}$ die Schlüsselmenge, die eingefügt werden soll. Betrachten Sie folgende Werte von $\boldsymbol{a}$:

$$A = \{(2,3), (1,2), (0,6), (4,5)\}.$$

Berechnen Sie $h_{\boldsymbol{a}}(x)$ für alle $\boldsymbol{a} \in A$ und $\boldsymbol{x} \in S$. Geben Sie für jedes $\boldsymbol{a} \in A$ an, ob $h_{\boldsymbol{a}}$ injektiv ist auf S.

Aufgabe 3.4 (4 Punkte)
Zwei Hashfunktionen $h, h' : K \to \{1, \ldots, m\}$ heißen *unabhängig*, falls für zwei zufällig gewählte Schlüssel $x, y \in K$ gilt:

$$\Pr\left(h(x) = h(y)\right) = \frac{1}{m}, \quad \Pr\left(h'(x) = h'(y)\right) = \frac{1}{m}$$

$$\Pr\left(h(x) = h(y) \wedge h'(x) = h'(y)\right) = \frac{1}{m^2}$$

Geben Sie ein K und m mit $1 < m < |K|$ sowie ein Paar h, h' von unabhängigen Hashing-Funktionen an. Beachten Sie, dass m beliebig groß sein darf. Beweisen Sie die Unabhängigkeit.
Tipp: Verwenden Sie ein K der Größe m^2.