1. Prove that if $H$ is $(k, c)$-independent for $k > 1$, then it is also $(k - 1, c)$-independent

\[ P[h(x_1) = a_1 \land h(x_2) = a_2 \land \cdots \land h(x_{k-1}) = a_{k-1}] = \]
\[ = \sum_{i=1}^{m} P[h(x_1) = a_1 \land h(x_2) = a_2 \land \cdots \land h(x_{k-1}) = a_{k-1} \land h(x_k) = i] \leq \]
\[ \leq m \frac{c}{m^k} = \frac{c}{m^{k-1}} \]

2. Prove that if $H$ is $(2, c)$-independent, then it is $c$-universal.

\[ \forall x_1, x_2, a_1, a_2 : P[h(x_1) = a_1 \land h(x_2) = a_2] \leq \frac{c}{m^2} \]

\[ P[h(x_1) = h(x_2)] = \sum_{i=1}^{m} P[h(x_1) = i \land h(x_2) = i] \leq m \frac{c}{m^2} = \frac{c}{m} \]

3. Prove that tabulation hashing is 3-independent but not 4-independent.

If tabulation uses only one table, it is perfectly random, so it is $k$-independent for each $k$ not exceeding the size of the universe.

Otherwise, we have $t \geq 2$ tables, each indexed by bits. Now consider $x_1, x_2, x_3$ and $a_1, a_2, a_3$ from the definition of 3-independence. At the same time, we imagine $x_i$ as ordered $k$-tuples of $b$-bit values (parts indexing individual tables).

(a) If there is a position $i$ in which all $x_1, x_2, x_3$ differ, then regardless of the values in the other positions, we can always fill in the table corresponding to this position so that the XORs come out $a_1, a_2, a_3$. Since $T_i$ is perfectly random, the probability of generating $a_1, a_2, a_3$ is $1/m^3$.

(b) Or there exists (WLOG) a position $i$ and position $j$, for which:

$x_1[i], x_2[i], x_1[i] = x_3[i], \quad \text{(call } x_1[i] = A \text{ and } x_2[i] = B)\]

$x_1[j] = x_2[j], x_1[j], x_3[j], \quad \text{(call } x_1[j] = C \text{ and } x_3[j] = D)\]
We get a set of equations (\(^\) is XOR)

\[
\begin{align*}
T_i[x_1[i]] \ ^\lor T_j[x_1[j]] \ ^\lor v_1 &= a_1 \\
T_i[x_2[i]] \ ^\lor T_j[x_2[j]] \ ^\lor v_2 &= a_2 \\
T_i[x_3[i]] \ ^\lor T_j[x_3[j]] \ ^\lor v_3 &= a_3
\end{align*}
\]

where \(v_1, v_2, \) and \(v_3\) are XORed values from the other tables.

\[
\begin{align*}
T_i[A] \ ^\lor T_j[C] &= a_1 \ ^\lor v_1 \\
T_i[B] \ ^\lor T_j[C] &= a_2 \ ^\lor v_2 \\
T_i[A] \ ^\lor T_j[D] &= a_3 \ ^\lor v_3
\end{align*}
\]

For each \(T_j[C]\) and each \(v_1, v_2, v_3,\) this set of equations has just one solution. So the probability that filling of the tables leads to a solution is exactly \(1/m^3\), as we need for 3-independence.

Counter-example for 4-independence:
Imagine these four hash keys.
Values A, B, C, D are corresponding values in the hash tables.

A B E F
A C E F
D B E F
D C E F

Then for each \(a_1, a_2, a_3, a_4:\)

\[
\begin{align*}
A \ ^\lor B \ ^\lor E \ ^\lor F &= a_1 \\
A \ ^\lor C \ ^\lor E \ ^\lor F &= a_2 \\
D \ ^\lor B \ ^\lor E \ ^\lor F &= a_3 \\
D \ ^\lor C \ ^\lor E \ ^\lor F &= a_4
\end{align*}
\]

\(\Rightarrow a_1 \ ^\lor a_2 \ ^\lor a_3 \ ^\lor a_4 = 0\)

If \(a_1 \ ^\lor a_2 \ ^\lor a_3 \ ^\lor a_4 = 0,\) probability is \(1/m^3\)
Otherwise, probability is 0.