Exercise 1: Bad Hash Functions

Let $m$ be the size of a hash table and $M \gg m$ the largest possible key of the elements we want to store in the table. The following “hash functions” are poorly chosen. Explain for each function why it is not a suitable hash function.

(a) $h : x \mapsto \lfloor \frac{x}{m} \rfloor \mod m$

(b) $h : x \mapsto (2x + 1) \mod m$ (m even).

(c) $h : x \mapsto (x \mod m) + \lfloor \frac{m}{x+1} \rfloor$

(d) For each calculation of the hash value of $x$ one chooses for $h(x)$ a uniform random number from \{0, \ldots, m−1\}

(e) For a set of “good” hash functions $h_1, \ldots, h_\ell$ with $\ell \in \Theta(\log m)$, we first compute $h_1(x)$, then $h_2(h_1(x))$ etc. until $h_\ell(h_{\ell−1}(\ldots h_1(x))\ldots)$. That is, the function is $h : k \mapsto h_\ell(h_{\ell−1}(\ldots h_1(x))\ldots)$

Exercise 2: (No) Families of Universal Hash Functions

Let $S = \{0, \ldots, M−1\}$ and $H_1 := \{h : x \mapsto a \cdot x^2 \mod m \mid a \in S\}$. Show that $H_1$ is not $c$-universal for constant $c \geq 1$ (that is $c$ is fixed and must not depend on $m$).