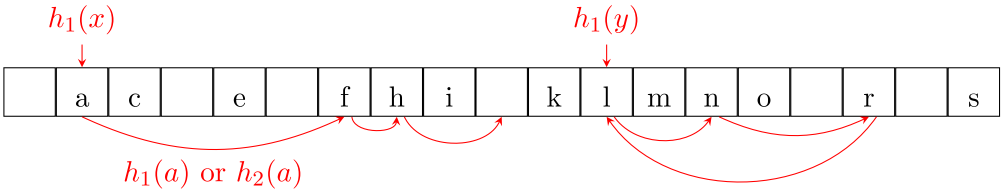
Data Structure I: Tutorial 7

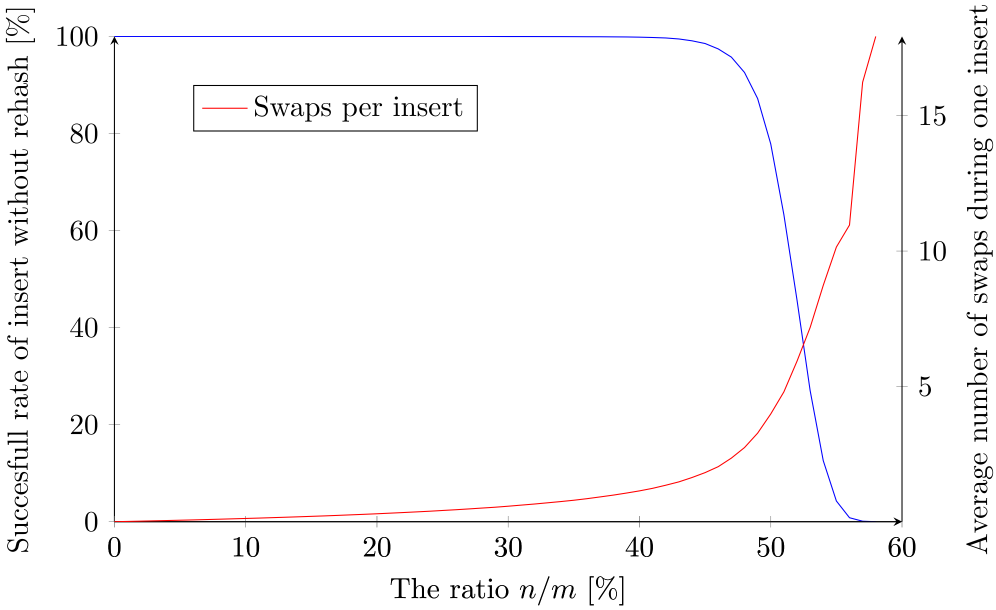
Exercise 1. *Linear probing stores n elements directly inside an array of size $m \geq n$ so that every position of the array contains at most one element. A new element x is inserted at the first position after $h(x)$ (modulo m). How to implement operations Find and Delete?*

Cuckoo hashing: Given two hash functions h_1 and h_2, a key x can be stored in $h_1(x)$ or $h_2(x)$. One position can store at most one element. Insertion of an element x is illustrated here.




```
pos ← h1(x)
for 6 log m-times do
    if T[pos] is empty then
        T[pos] ← x
        return
    swap(x, T[pos])
    if pos == h1(x) then
        pos ← h2(x)
    else
        pos ← h1(x)
rehash()
insert(x)
```

Exercise 2. *Rehash function creates a new pair of hashing function and reinsert all elements based on them. However, be very careful that rehash uses the function insert which may fail, so another rehash may be needed, and so on. Furthermore, keep in mind that function insert always has one element which is not stored in the array, so do not lose it during rehash. Discuss implementation of the function rehash.*



Definition 3. *A system $\mathcal{H}$ of hashing functions is c-universal, if for every $x, y \in U$ with $x \neq y$ the number of functions $h \in \mathcal{H}$ satisfying $h(x) = h(y)$ is at most $\frac{c|\mathcal{H}|}{m}$ where $c \geq 1$. Equivalently, a system $\mathcal{H}$ of hashing functions is c-universal, if uniformly chosen $h \in \mathcal{H}$ satisfies $P[h(x) = h(y)] \leq \frac{c}{m}$ for every $x, y \in U$ with $x \neq y$.*

Definition 4. *A set $\mathcal{H}$ of hash functions is $(2, c)$-independent if for every $x_1, x_2 \in U$ with $x_1 \neq x_2$ and $z_1, z_2 \in M$ the number of functions $h \in \mathcal{H}$ satisfying $h(x_1) = z_1$ and $h(x_2) = z_2$ is at most $\frac{c|\mathcal{H}|}{m^2}$. Equivalently, a set $\mathcal{H}$ of hash functions is $(2, c)$-independent if randomly chosen $h \in \mathcal{H}$ satisfies $P[h(x_1) = z_1$ and $h(x_2) = z_2] \leq \frac{c}{m^2}$ for every $x_1 \neq x_2$ elements of U and $z_1, z_2 \in M$.*

Note that buckets z_1 and z_2 can be the same but elements x_1 and x_2 must be distinct.

Definition 5. *A set $\mathcal{H}$ of hash functions is (k, c)-independent if randomly chosen $h \in \mathcal{H}$ satisfies $P[h(x_i) = z_i$ for every $i = 1, \ldots, k] \leq \frac{c}{m^k}$ for every pair-wise different elements $x_1, \ldots, x_k \in U$ and $z_1, \ldots, z_k \in M$.*
A set $\mathcal{H}$ of hash functions is k-independent if it is (k, c)-independent for some $c \geq 1$.

Definition 6. *Let $p \geq |U| \geq m$ be a prime where $U = [u]$. We define the hash function*

$$h_{a,b}(x) = (ax + b \mod p) \mod m.$$

Hashing system Multiply-mod-prime is

$$\mathcal{H} = \{h_{a,b};\ a, b \in [p]\}$$

Theorem 7. *Hash system Multiply-mod-prime is 2-universal and 2-independent.*

Definition 8. *Assume that $u = 2^w$ and $m = 2^l$ for some integer w, l. We define the hash function*

$$h_a(x) = (ax \mod 2^w) >> (w - l)$$

Hashing system Multiply-shift is

$$\mathcal{H} = \{h_a;\ a \text{ is odd } w\text{-bits integer }\}$$

Theorem 9. *Hash system Multiply-shift is 1-universal and 2-independent.*

Exercise 10.

- *Prove that if a hashing system is (k,c)-independent, then it is $(k-1,c)$-independent.*
- *Prove that if a hashing system is $(2,c)$-independent, then it is also c-universal.*
- *Prove that 1-independent hashing system may not be c-universal.*

Definition 11. *Tabular hashing*

- *Assume that $u = 2^w$ and $m = 2^l$ and w is a multiple of an integer d*
- *Binary code of $x \in U$ is split to d parts $x^0, \ldots, x^{d-1}$ by $\frac{w}{d}$ bits*
- *For every $i \in [d]$ generate a totally random hashing function $T_i : [2^{w/d}] \to M$*
- *Hashing function is $h(x) = T_0(x^0) \oplus \cdots \oplus T_{d-1}(x^{d-1})$*

$\oplus$ denotes bit-wise exclusive or (XOR).

Theorem 12. *Tabular hashing is 3-independent, but it is not 4-independent.*