# Data Structures 1

NTIN066

Jirka Fink

Department of Theoretical Computer Science and Mathematical Logic
Faculty of Mathematics and Physics
Charles University in Prague

Summer semester 2025/26

Last change on April 8, 2026

## Basic Concepts

- Notation: $[n] = \{0, \ldots, n-1\}$
- We have a universe $U$ of all elements
- We want to store a subset $S \subseteq U$ of size $n$
- We store $S$ in an array of size $m$ using a hash function $h : U \to M$, where $M = [m]$
- A hashing system $\mathcal{H}$ refers to any set of hash functions
- Two elements $x, y \in S$ collide if $h(x) = h(y)$
- We consider the universe $U = [u]$ for any $u \in \mathbb{N}$, unless stated otherwise

# $c$-Universal Hashing System

## $c$-Universal System (Equivalent definitions)

A system of hash functions $\mathcal{H}$ is $c$-universal if for all distinct $x, y \in U$

- the number of hash functions $h \in \mathcal{H}$ satisfying $h(x) = h(y)$ is at most $\frac{c|\mathcal{H}|}{m}$
- a randomly chosen $h \in \mathcal{H}$ satisfies $P[h(x) = h(y)] \leq \frac{c}{m}$.

# $c$-Universal Hashing System

## $c$-Universal System (Equivalent definitions)

A system of hash functions $\mathcal{H}$ is $c$-universal if for all distinct $x, y \in U$

- the number of hash functions $h \in \mathcal{H}$ satisfying $h(x) = h(y)$ is at most $\frac{c|\mathcal{H}|}{m}$
- a randomly chosen $h \in \mathcal{H}$ satisfies $P[h(x) = h(y)] \leq \frac{c}{m}$.

## Example of a $c$-Universal Hashing System

- Parameters: $p$ and $m$, where $p \geq u \geq m$ and $p$ is a prime number
- Hash function $h_a(x) = (ax \bmod p) \bmod m$
- The hashing system $\mathcal{H} = \{h_a;\ a \in [p]\}$ is $c$-universal
- The hash function from the system $\mathcal{H}$ is determined by the value of $a$
- Thus, a random selection of a hash function from $\mathcal{H}$ is a random generation of $a \in [p]$

# Why so complicated approach is needed?

## The simpliest hash function

Can we use a simple hash function $h(x) = x \mod m$? When this hash function is sufficient and useful?

# Why so complicated approach is needed?

## The simpliest hash function

Can we use a simple hash function $h(x) = x \mod m$? When this hash function is sufficient and useful?

## Hashing a pair of integers in C++

```
template <class T1, class T2>
size_t operator()(const pair<T1, T2>& p) const {
        return hash<T1>{}(p.first) ^ hash<T2>{}(p.second);
}
```

Can we use this function to hash edges of a graph?

# Why so complicated approach is needed?

## The simpliest hash function

Can we use a simple hash function $h(x) = x \mod m$? When this hash function is sufficient and useful?

## Hashing a pair of integers in C++

```
template <class T1, class T2>
size_t operator()(const pair<T1, T2>& p) const {
        return hash<T1>{}(p.first) ^ hash<T2>{}(p.second);
}
```

Can we use this function to hash edges of a graph?

## When do we need more than one hash function?

- Python: CVE-2012-1150, CVE-2013-7040
- JRuby: CVE-2011-4838
- PHP: CVE-2011-4885
- Ruby: CVE-2011-4815
- Apache Geronimo: CVE-2011-5034

# (k,c)-independent hash function system

## (2,c)-independent system of hash functions (Equivalent definitions)

A system of hash functions $\mathcal{H}$ 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 $h \in \mathcal{H}$ satisfying $h(x_1) = z_1$ and $h(x_2) = z_2$ is at most $\frac{c|\mathcal{H}|}{m^2}$
- a randomly chosen $h \in \mathcal{H}$ satisfies $P[h(x_1) = z_1 \text{ and } h(x_2) = z_2] \leq \frac{c}{m^2}$.

# (k,c)-independent hash function system

## (2,c)-independent system of hash functions (Equivalent definitions)

A system of hash functions $\mathcal{H}$ 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 $h \in \mathcal{H}$ satisfying $h(x_1) = z_1$ and $h(x_2) = z_2$ is at most $\frac{c|\mathcal{H}|}{m^2}$
- a randomly chosen $h \in \mathcal{H}$ satisfies $P[h(x_1) = z_1 \text{ and } h(x_2) = z_2] \leq \frac{c}{m^2}$.

## $(k, c)$-independent system of hash functions

Let $k \in \mathbb{N}$, $K = \{1, \ldots, k\}$ and $c \geq 1$.
A system of hash functions $\mathcal{H}$ is $(k, c)$-independent if a randomly chosen $h \in \mathcal{H}$ satisfies $P[h(x_i) = z_i \, \forall i \in K] \leq \frac{c}{m^k}$ for all pair-wise distinct $x_1, \ldots, x_k \in U$ and all $z_1, \ldots, z_k \in M$.

# (k,c)-independent hash function system

## (2,c)-independent system of hash functions (Equivalent definitions)

A system of hash functions $\mathcal{H}$ 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 $h \in \mathcal{H}$ satisfying $h(x_1) = z_1$ and $h(x_2) = z_2$ is at most $\frac{c|\mathcal{H}|}{m^2}$
- a randomly chosen $h \in \mathcal{H}$ satisfies $P[h(x_1) = z_1 \text{ and } h(x_2) = z_2] \leq \frac{c}{m^2}$.

## $(k, c)$-independent system of hash functions

Let $k \in \mathbb{N}$, $K = \{1, \ldots, k\}$ and $c \geq 1$.
A system of hash functions $\mathcal{H}$ is $(k, c)$-independent if a randomly chosen $h \in \mathcal{H}$ satisfies $P[h(x_i) = z_i \, \forall i \in K] \leq \frac{c}{m^k}$ for all pair-wise distinct $x_1, \ldots, x_k \in U$ and all $z_1, \ldots, z_k \in M$.

## $k$-independent system of hash functions

- The system $\mathcal{H}$ is $k$-independent if it is $(k, c)$-independent for some $c \geq 1$.
- The system $\mathcal{H}$ is strongly $k$-independent if it is $(k, 1)$-independent.

# Universal and Independent Hash Function System: Properties

1. The system $\mathcal{H} = \{h_a(x) = a;\ a \in M\}$ is 1-independent, but useless

# Universal and Independent Hash Function System: Properties

1. The system $\mathcal{H} = \{h_a(x) = a;\ a \in M\}$ is 1-independent, but useless
2. In the definition of a $k$-independent system, we cannot require that the slots $z_1, \ldots, z_k$ be pairwise distinct

# Universal and Independent Hash Function System: Properties

1. The system $\mathcal{H} = \{h_a(x) = a;\ a \in M\}$ is 1-independent, but useless
2. In the definition of a $k$-independent system, we cannot require that the slots $z_1, \ldots, z_k$ be pairwise distinct
3. A $(k, c)$-independent system of hash functions is $(k-1, c)$-independent

# Universal and Independent Hash Function System: Properties

1. The system $\mathcal{H} = \{h_a(x) = a;\ a \in M\}$ is 1-independent, but useless
2. In the definition of a $k$-independent system, we cannot require that the slots $z_1, \ldots, z_k$ be pairwise distinct
3. A $(k, c)$-independent system of hash functions is $(k-1, c)$-independent
4. A $(2, c)$-independent system of hash functions is $c$-universal

# Universal and Independent Hash Function System: Properties

1. The system $\mathcal{H} = \{h_a(x) = a;\ a \in M\}$ is 1-independent, but useless
2. In the definition of a $k$-independent system, we cannot require that the slots $z_1, \ldots, z_k$ be pairwise distinct
3. A $(k, c)$-independent system of hash functions is $(k-1, c)$-independent
4. A $(2, c)$-independent system of hash functions is $c$-universal
5. There exists a 1-universal system that is not 2-independent

# Universal and Independent Hash Function System: Properties

1. The system $\mathcal{H} = \{h_a(x) = a;\ a \in M\}$ is 1-independent, but useless
2. In the definition of a $k$-independent system, we cannot require that the slots $z_1, \ldots, z_k$ be pairwise distinct
3. A $(k, c)$-independent system of hash functions is $(k-1, c)$-independent
4. A $(2, c)$-independent system of hash functions is $c$-universal
5. There exists a 1-universal system that is not 2-independent
6. There exists a strongly $k$-independent system that is not $(k+1)$-independent

# Universal and Independent Hash Function System: Properties

1. The system $\mathcal{H} = \{h_a(x) = a;\ a \in M\}$ is 1-independent, but useless
2. In the definition of a $k$-independent system, we cannot require that the slots $z_1, \ldots, z_k$ be pairwise distinct
3. A $(k, c)$-independent system of hash functions is $(k-1, c)$-independent
4. A $(2, c)$-independent system of hash functions is $c$-universal
5. There exists a 1-universal system that is not 2-independent
6. There exists a strongly $k$-independent system that is not $(k+1)$-independent
7. For every hash function system $\mathcal{H}$ and for all pair-wise distinct $x_1, \ldots, x_k \in U$, there exist $z_1, \ldots, z_k \in M$ such that $P[h(x_i) = z_i\ \forall i \in K] \geq \frac{1}{m^k}$

# Universal and Independent Hash Function System: Properties

1. The system $\mathcal{H} = \{h_a(x) = a;\ a \in M\}$ is 1-independent, but useless
2. In the definition of a $k$-independent system, we cannot require that the slots $z_1, \ldots, z_k$ be pairwise distinct
3. A $(k, c)$-independent system of hash functions is $(k-1, c)$-independent
4. A $(2, c)$-independent system of hash functions is $c$-universal
5. There exists a 1-universal system that is not 2-independent
6. There exists a strongly $k$-independent system that is not $(k+1)$-independent
7. For every hash function system $\mathcal{H}$ and for all pair-wise distinct $x_1, \ldots, x_k \in U$, there exist $z_1, \ldots, z_k \in M$ such that $P[h(x_i) = z_i\ \forall i \in K] \geq \frac{1}{m^k}$
8. If $\mathcal{H}$ is strongly $k$-independent, then for distinct $x_1, \ldots, x_k \in U$ and for $z_1, \ldots, z_k \in M$
   - $P[h(x_i) = z_i\ \forall i \in K] = \frac{1}{m^k} = \prod_{i=1}^{k} P[h(x_i) = z_i]$
   - $P[h(x_k) = z_k | h(x_i) = z_i\ \forall i = 1, \ldots, k-1] = \frac{1}{m}$

# Universal and Independent Hash Function System: Properties

1. The system $\mathcal{H} = \{h_a(x) = a;\ a \in M\}$ is 1-independent, but useless
2. In the definition of a $k$-independent system, we cannot require that the slots $z_1, \ldots, z_k$ be pairwise distinct
3. A $(k, c)$-independent system of hash functions is $(k-1, c)$-independent
4. A $(2, c)$-independent system of hash functions is $c$-universal
5. There exists a 1-universal system that is not 2-independent
6. There exists a strongly $k$-independent system that is not $(k+1)$-independent
7. For every hash function system $\mathcal{H}$ and for all pair-wise distinct $x_1, \ldots, x_k \in U$, there exist $z_1, \ldots, z_k \in M$ such that $P[h(x_i) = z_i\ \forall i \in K] \geq \frac{1}{m^k}$
8. If $\mathcal{H}$ is strongly $k$-independent, then for distinct $x_1, \ldots, x_k \in U$ and for $z_1, \ldots, z_k \in M$
   - $P[h(x_i) = z_i\ \forall i \in K] = \frac{1}{m^k} = \prod_{i=1}^{k} P[h(x_i) = z_i]$
   - $P[h(x_k) = z_k | h(x_i) = z_i\ \forall i = 1, \ldots, k-1] = \frac{1}{m}$
9. If $\mathcal{H}$ is $(k, c)$-independent, then $|\mathcal{H}| \geq \frac{m^k}{c}$ and to identify a function from $|\mathcal{H}|$ we need at least $k \log m - \log c$ bits

## Multiply-Shift

- We assume that $|U| = 2^w$ and $m = 2^l$
- $h_a(x) = (ax \mod 2^w) >> (w - l)$
- $\mathcal{H} = \{h_a; a \text{ is an odd } w\text{-bit integer } \}$

## Multiply-Shift

- We assume that $|U| = 2^w$ and $m = 2^l$
- $h_a(x) = (ax \mod 2^w) >> (w - l)$
- $\mathcal{H} = \{h_a; a \text{ is an odd } w\text{-bit integer } \}$

## Implementation in C

```
uint64_t hash(uint64_t x, uint64_t l, uint64_t a)
{ return (a*x) >> (64-l); }
```

## Multiply-Shift

- We assume that $|U| = 2^w$ and $m = 2^l$
- $h_a(x) = (ax \mod 2^w) >> (w - l)$
- $\mathcal{H} = \{h_a; a \text{ is an odd } w\text{-bit integer } \}$

## Implementation in C

```
uint64_t hash(uint64_t x, uint64_t l, uint64_t a)
{ return (a*x) >> (64-l); }
```

## Properties of the Multiply-Shift Scheme

- 2-universal
- Very efficient on real-world hardware
- Widely used in practice
- The entire computation must be carried out using unsigned integer types, since we require the lower $w$ bits of the product $ax$