# Algorithms Theory

## 05 - Hashing

Dr. Alexander Souza

# Overview

- Introduction
- Universal hashing
- Perfect hashing

# The dictionary problem

**Given:** Universe $U$ = [0…$N$-1] , where $N$ is a natural number.

**Goal:** Maintain set $S \subseteq U$ under the following operations.

- Search($x$,$S$): Is $x \in S$?
- Insert($x$,$S$): Insert $x$ into $S$ if not already in $S$.
- Delete($x$,$S$): Delete $x$ from $S$.

# Trivial implementation

Array A[0…$N$-1] where $A[i] = 1 \Leftrightarrow i \in S$

Each operation takes time O(1) but the required memory space is $\Theta(N)$.

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| A | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 |
| | 0 | | | | | | | | | | $N$–1 |

**Goal:** Space requirement O(|$S$|) and expected time O(1) per operation.

# Idea of hashing

Use an array of length O(|S|).

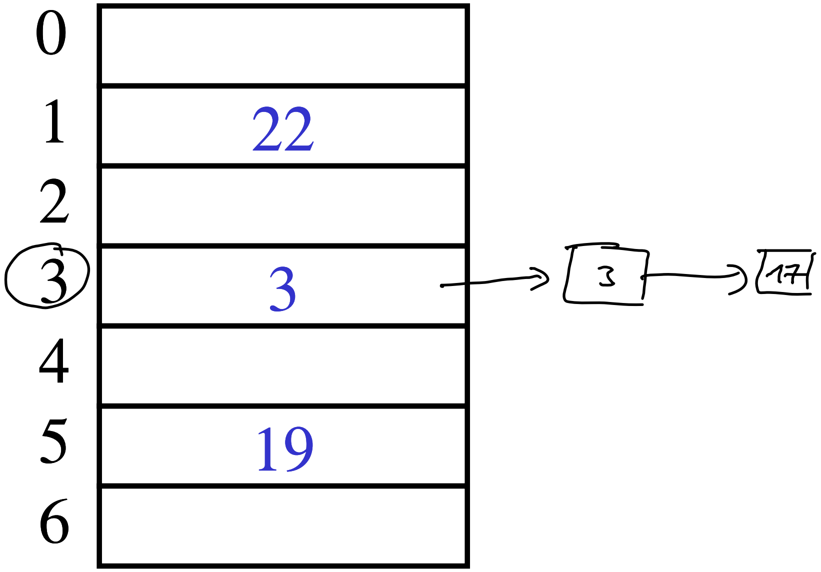
Compute the position where to store an element using a function defined on the keys.

| | | |
|---|---|---|
| Universe | $U = [0 \ldots N\text{-}1]$ | $m = O(\lvert S \rvert)$ |
| Hash table | Array $T[0 \ldots m\text{-}1]$ | |
| Hash function | $h: U \rightarrow [0 \ldots m\text{-}1]$ | |

Element $x \in S$ is stored in $T[h(x)]$.

# Example

$N = 100$; $U = [0\ldots99]$; $m = 7$; $h(x) = x \bmod 7$; $S = \{3, 19, 22\}$



If 17 is inserted next, a collision arises because $h(17) = 3$.

# Possible collision resolutions

- Hashing with chaining: $T[i]$ contains a list of elements.
- Hashing with open addressing: Instead of one address for an element there are $m$ many that are probed sequentially.
- Universal hashing: Choose a hash function such that only few collisions occur. Collisions are resolved by chaining.
- Perfect hashing: Choose a hash function such that no collisions occur.

Set $S$ is know at the outset

# Universal hashing

**Idea:** Use a class $H$ of hash functions. The hash function $h \in H$ actually used is chosen uniformly at random from $H$.

**Goal:** For each $S \subseteq U$, the expected time of each operation is $O(1 + \beta)$ , where $\beta = |S|/m$ is the load factor of the table.

Worst case : $O(\text{longest chain})$

$|S|$

Expectation : $O\left(\frac{|S|}{m}\right) = O(\beta)$

**Property of *H*:** For two arbitrary elements $x,y \in U$, only "few" $h \in H$ lead to a collision ($h(x) = h(y)$).

# Universal hashing

**Definition:** Let $N$ and $m$ be natural numbers. A class $\underline{\boxed{H} \subseteq \{ h : [0\ldots N-1] \rightarrow [0\ldots m-1] \}}$ is $\underline{\text{universal}}$ if for all $\underline{x,y} \in U = [0\ldots N-1]$, $\underline{x \neq y}$ :

collision for x, y

$$\frac{|\{h \in H : h(x) = h(y)\}|}{|H|} \leq \frac{1}{m}$$

all mappings in H

**Intuitively:** An $h$ chosen uniformly at random is as good as if the table positions of the elements are chosen uniformly at random.

# A universal class of functions

Let $N$, $m$ be natural numbers, where $N$ is prime.

For numbers $a \in \{1, \dots, N-1\}$ and $b \in \{0, \dots, N-1\}$, let

$h_{a,b}: U = [0 \dots N-1] \to \{0, \dots, m-1\}$ be defined as:

$$h_{a,b}(x) = ((ax + b) \bmod N) \bmod m$$

$$H = \{ h_{a,b} : a \in \{1, \dots, N-1\}, b \in \{0, \dots, N-1\}\}$$

$$|H| = (N-1) \cdot N$$

**Theorem:** $H = \{h_{a,b}(x) \mid 1 \leq a < N \text{ and } 0 \leq b < N\}$ is a universal class of hash functions.

# Proof

$$|\{h_{a,b} : h_{a,b}(x) = h_{a,b}(y)\}| \leq \frac{|H|}{m} = \frac{N\cdot(N-1)}{m}$$

$N\cdot(N-1)$ pairs $a,b$

$N\cdot(N-1)$ pairs ~~#~~ $q,r$

no pair $(q,r)$ occurs twice

Consider a fixed pair $x, y$ with $x \neq y$ .

$h_{a,b}(x) = ((ax+b) \bmod N) \bmod m \quad h_{a,b}(y) = ((ay+b) \bmod N) \bmod m$

First ~~Set~~ Step: without mod ~~[illegible]~~

1. Pairs $(q,r)$ with $q = \underline{(ax+b) \bmod N}$ and $r = \underline{(ay+b) \bmod N}$ for variable $a,b$ take the whole range $0 \leq q,r < N$ with $q \neq r$

$0 = q - r = (ax+b) - (ay+b) \bmod N = a(x-y) \bmod N \Rightarrow a\cdot(x-y) = c\cdot N$

-- $q \neq r$: $q = r$ implies $a(x-y) = cN$

N a factor of a
or N a factor of x - y ↯

N prime
$1 \leq a \leq N-1$
$|x-y| \leq N-1$

Assume -- different pairs $a,b$ yield different pairs $(q,r)$.

$\exists (a,b), (a',b')$ $(ax+b) \bmod N = q$ $\quad (ay+b) \bmod N = r$ $\}$ $a(x-y) \bmod N = q-r$

$h_{a,b}(x) = h_{a',b'}(x)$ $(a'x+b') \bmod N = q$ $\quad (a'y+b') \bmod N = r$ $\}$ $a'(x-y) \bmod N = q-r$

$h_{a,b}(y) = h_{a',b'}(y)$ imply $(a-a')(x-y) = cN$

$(a-a')(x-y) \bmod N = 0$

↯

N prime

$|a-a'| \leq N-1$

$|x-y| \leq N-1$

# Proof

Fixed pair $x,y$ with $x \neq y$ .

$h_{a,b}(x) = (\underbrace{(ax+b) \bmod N}_{q}) \underline{\bmod m}$ $\quad h_{a,b}(y) = (\underbrace{(ay+b) \bmod N}_{r}) \underline{\bmod m}$

2. How many pairs $(q,r)$ with $q = (ax+b) \bmod N$ and $r = (ay+b) \bmod N$ are mapped into the $\underline{\text{same residue class mod } m}$?

   For a $\underline{\text{fixed } q}$, there are only $\underline{(N-1)/m}$ numbers $r$, with $q \bmod m = r \bmod m$ and $q \neq r$.

$q \bmod m$

$q+m \bmod m, \quad q-m \bmod m$

$q+2m \bmod m, \quad q-2m \bmod m$

$\vdots$

$\Rightarrow \quad |\{h \in H : h(x) = h(y)\}| \leq \underline{N(N-1)/m} = \underline{|H|/m}$

$\Rightarrow$ H is a universal class.

# Analysis of the operations

Assumptions:
1. *h* is chosen uniformly at random from a universal class *H*.
2. Collisions are resolved by chaining.

For $h \in H$ and $x,y \in U$ let

$$\delta_h(x,y) = \begin{cases} 1 & h(x) = h(y) \text{ and } x \neq y \\ 0 & \text{otherwise} \end{cases}$$

collision

$S \subseteq U$

$$\delta_h(x,S) = \sum_{y \in S} \delta_h(x,y)$$ is the number of elements in $T[h(x)]$

same position as $x$

different from x when *S* is stored.

# Analysis of the operations

*h* fixed, *S* fixed

- Search(*x*, *S*)

  $1 + \delta_h(x, S)$

- Insert(*x*, *S*)

  $1 + \delta_h(x, S)$

- Delete(*x*, *S*)

  $1 + \delta_h(x, S)$

In expectation when h is chosen u. a. r from H :

$$\sum_{h \in H} \frac{1}{|H|} (1 + \delta_h(x, S))$$

# Analysis of the operations

**Theorem:** Let $H$ be a universal class and $S \subseteq U = [0\ldots N\text{-}1]$ with $|S| = n$.

1. For any $x \in U$:

$$\frac{1}{|H|}\sum_{h\in H}(1+\delta_h(x,S)) \le \begin{cases} 1+n/m & x\notin S \\ 1+(n-1)/m & x\in S \end{cases} \le 1+\frac{n}{m}$$

load factor

2. The expected time of the operations 'Search', 'Insert', and 'Delete' is O(1 + β), where β = $n/m$ is the load factor.

# Proof

1. $$\sum_{h\in H}(1+\delta_h(x,S)) \overset{\text{Def.}}{=} \underline{|H|} + \sum_{\underline{h\in H}}\sum_{\underline{y\in S}}\delta_h(x,y)$$

$|\{h \in H : h(x) = h(y)\}| \le \frac{|H|}{m}$

$H$ universal

$$= |H| + \sum_{\boxed{y\in S}}\overbrace{\sum_{h\in H}\delta_h(x,y)}$$

$\delta_h(x,x) = 0$

$$\le |H| + \overset{\checkmark}{\sum_{\boxed{y\in S\setminus\{x\}}}} \frac{|H|}{m}$$

$$\le \begin{cases} |H|\cdot(1+n/m) & \underline{x\notin S} \\ |H|\cdot(1+(n-1)/m) & \underline{x\in S} \end{cases}$$

✓

2. Follows from 1. ✓