Algorithms Theory

05 - Hashing

Prof. Dr. S. Albers

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 ⇔ $i \in S$

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 $O(|S|)$.

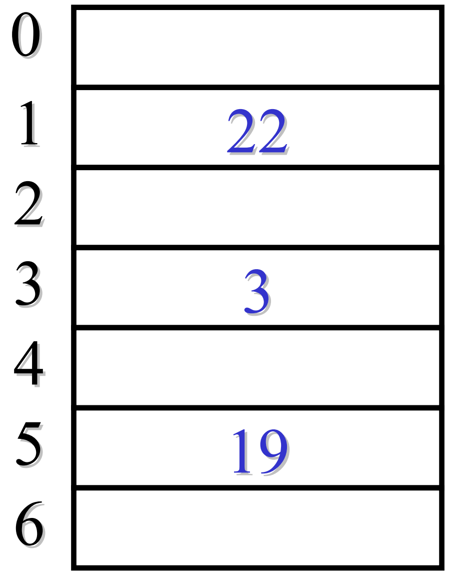
Compute the position where to store an element using a function defined on the keys.

Universe	$U = [0 \ldots N-1]$
Hash table	Array $T[0 \ldots m-1]$
Hash function	$h: U \rightarrow [0 \ldots m-1]$

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.

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.

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 $H \subseteq \{ h : [0 \ldots N\text{-}1] \rightarrow [0 \ldots m\text{-}1] \}$ is universal if for all $x,y \in U = [0 \ldots N\text{-}1]$, $x \neq y$:

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

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

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

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

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

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

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

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

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

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

$(ax+b) \bmod N = q \qquad (ay+b) \bmod N = r$

$(a'x+b') \bmod N = q \qquad (a'y+b') \bmod N = r$

imply $(a-a')(x-y) = cN$

Proof

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

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

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

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

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

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

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

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

different from x when *S* is stored.

Analysis of the operations

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

1. For any $x \in U$:

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

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

Proof

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

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

$$\leq |H| + \sum_{y\in S\setminus\{x\}}\frac{|H|}{m}$$

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

2. Follows from 1.

Perfect hashing

Choose a hash function that is injective (i.e. one-to-one) on the set *S* to be stored. (Assumption: *S* is known in advance.)

Two-level hashing scheme

1. In the first level, *S* is partitioned into "short lists" (hashing with chaining).
2. In the second level for each list, a separate injective hash function is used.

Construction of injective hash functions

Let $U = [0 \ldots N\text{-}1]$.

For $k \in \{1,\ldots,N\text{-}1\}$, let

$$h_k : U \rightarrow \{0,\ldots,m\text{-}1\}$$
$$x \rightarrow ((kx) \bmod N) \bmod m$$

Let $S \subseteq U$. Is it possible to choose k such that h_k restricted to S is injective?

h_k restricted to S is injective if for all $x,y \in S$, $x \neq y$,

$$h_k(x) \neq h_k(y)$$

A measure for the violation of injectivity

For $0 \leq i \leq m$-1 and $1 \leq k \leq N$-1 let

$$b_{ik} = |\{ x \in S : h_k(x) = i \}|$$

Then:

$$|\{ (x,y) \in S^2 : x \neq y \text{ and } h_k(x) = h_k(y) = i \}| = b_{ik} (b_{ik} - 1)$$

Define

$$B_k = \sum_{i=0}^{m-1} b_{ik} (b_{ik} - 1)$$

B_k measures to which extent h_k restricted to S is not injective.

Injectivity

Lemma 1: h_k restricted to S is injective $\Leftrightarrow$ $B_k < 2$

Proof:

$B_k < 2 \quad \Rightarrow \quad B_k \leq 1 \quad \Rightarrow \quad b_{ik}\,(b_{ik} - 1) \in \{0,1\}$ for all i

$\Rightarrow \quad b_{ik} \in \{0,1\} \quad \Rightarrow \quad h_k$ restricted to S is injective

h_k restricted to S is injective $\quad \Rightarrow \quad b_{ik} \in \{0,1\}$ for all i

$\Rightarrow \quad B_k = 0$

Injectivity

Lemma 2: Let N be a prime number, $S \subseteq U = [0\ldots N\text{-}1]$ with $|S| = n$. Then

$$\sum_{k=1}^{N-1} B_k \leq 2\frac{n(n-1)}{m}(N-1)$$

If $m > n(n\text{-}1)$, then there exists B_k with $B_k < 2$, i.e. there is an h_k that is injective on S.

Proof of Lemma 2

$$\sum_{k=1}^{N-1}\sum_{i=0}^{m-1} b_{ik}(b_{ik}-1)$$

$$=\sum_{k=1}^{N-1}\sum_{i=0}^{m-1} |\{(x,y)\in S^2 : x\neq y, h_k(x)=h_k(y)=i\}|$$

$$=\sum_{\substack{(x,y)\in S^2\\ x\neq y}} |\{k : h_k(x)=h_k(y)\}|$$

Let $(x,y) \in S^2$, $x \neq y$, be fixed. How many k exist with $h_k(x) = h_k(y)$?

Proof of Lemma 2

$$h_k(x) = h_k(y)$$
$$\Leftrightarrow ((kx) \bmod N) \bmod m = ((ky) \bmod N) \bmod m$$
$$\Leftrightarrow (kx \bmod N - ky \bmod N) \bmod m = 0$$
$$\Leftrightarrow k(x-y) \bmod N = cm$$

$q = k(x\text{-}y) \bmod N$
-- different k, k' yield different q, q'.
$k(x\text{-}y) \bmod N = q$ $\qquad\qquad$ $k'(x\text{-}y) \bmod N = q$

$(k\text{-}k')(x\text{-}y) = c'N$

-- only $\lceil (N\text{-}1)/m \rceil$ many q are mapped into the same residue class mod m

Results

Corollary 1: There are at least (N-1)/2 many k with $B_k \leq 4n(n\text{-}1)/m$.
Such a k can be determined in expected time O(m+n).

Proof: Suppose that there are less than (N-1)/2 many k with

$B_k \leq 4n(n\text{-}1)/m.$

Then there are at least (N-1)/2 many k with $B_k > 4n(n\text{-}1)/m$

$$\Rightarrow \sum_{k=1}^{N-1} B_k > \frac{N-1}{2} \frac{4n(n-1)}{m} = \frac{N-1}{m} 2n(n-1)$$

With probability ≥ ½ , a k chosen at random fulfills the condition. The expected number of trials is ≤ 2.

Results

Corollary 2:

a) Let $m = 2n(n-1)+1$. Then at least $(N-1)/2$ of the h_k are injective on S. Such an h_k can be found in expected time $O(m+n)=O(n^2)$.

b) Let $m = n$. Then for at least $(N-1)/2$ of the h_k it holds that $B_k \leq 4(n-1)$. Such an h_k can be found in expected time $O(n)$.

Two-level scheme

$S \subseteq U = [0 \ldots N\text{-}1]$ $|S| = n = m$

Idea: Use Corollary 2b and divide S into subsets of size $O(\sqrt{n})$.
Use Cor. 2a for each subset.

1. Choose k with $B_k \leq 4(n\text{-}1) \leq 4n$.
   $h_k : x \rightarrow ((kx) \bmod N) \bmod n$
2. $W_i = \{ x \in S : h_k(x) = i \}$, $b_i = |W_i|$, $m_i = 2b_i(b_i - 1) + 1$ for $1 \leq i \leq n\text{-}1$
   Choose k_i such that

$$h_{k_i} : x \rightarrow (k_i x \bmod N) \bmod m_i$$

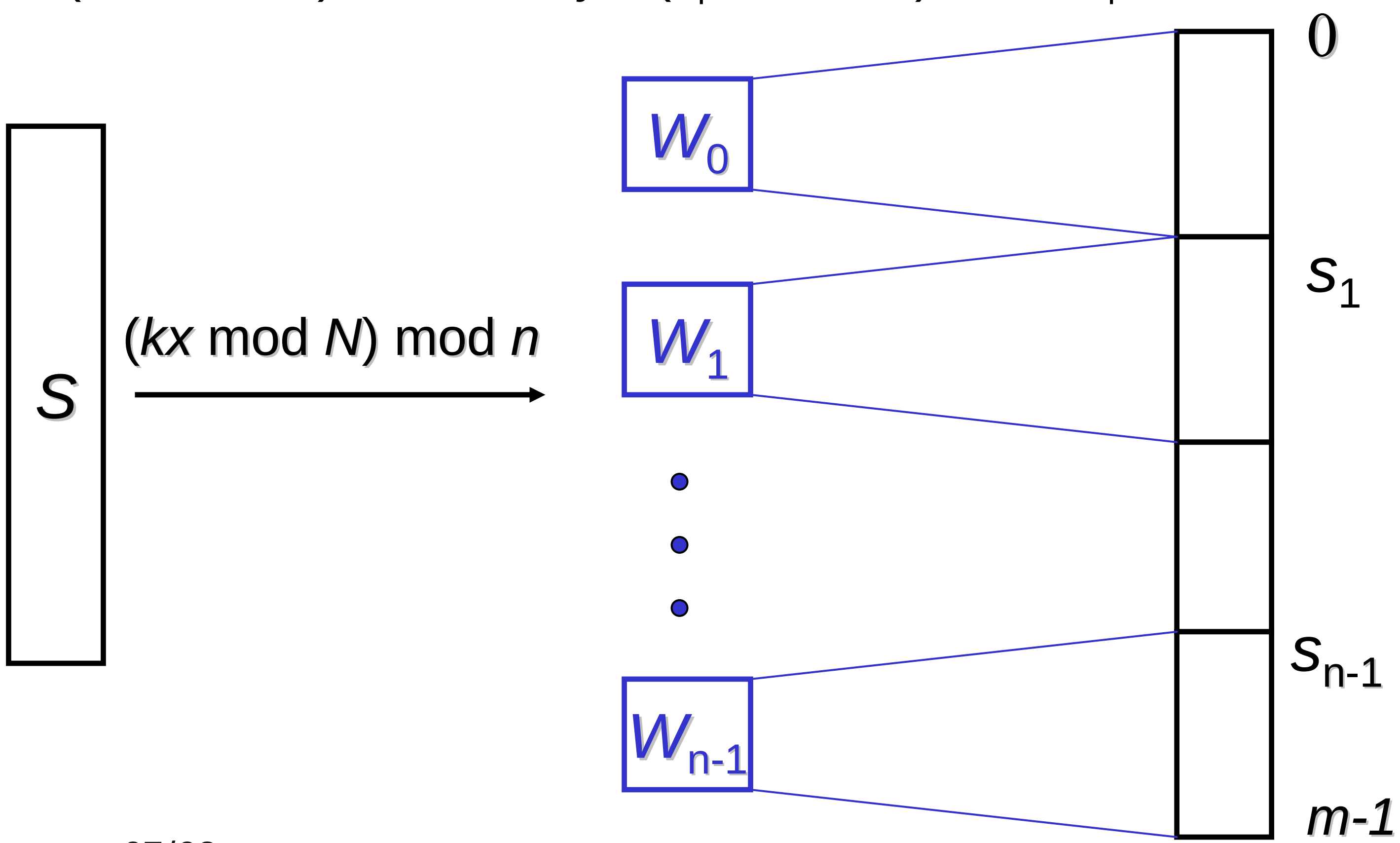
restricted to W_i is injective.

Two-level scheme

3. $s_i = \sum_{j<i} m_j$

Store $x \in S$ in table position $T[s_i + j]$ where

$i = (k\,x \bmod N) \bmod n$ $\quad$ $j = (k_i\,x \bmod N) \bmod m_i$



Space required for hash table and functions

$$m = \sum_{i=0}^{n-1} m_i = \sum_{i=0}^{n-1} (2b_i(b_i - 1) + 1) = n + 2B_k$$

$$\leq n + 8(n-1) \leq 9n$$

Additional space is required for storing k_i, m_i and s_i.

The total space requirement is O(n).

Construction time

- According to Cor. 2b, k can be found in expected time $O(n)$.
- W_i, b_i, m_i, s_i can be computed in time $O(n)$.
- According to Cor. 2a, each k_i can be computed in expected time $O(b_i^2)$.

Total expected time:

$$O\left(n + \sum_{i=0}^{n} b_i^2\right) = O(n + B_k) = O(n)$$

Main result

Theorem: Let N be a prime number and $S \subseteq U = [0 \ldots N-1]$ with $|S| = n$.

A perfect hash table of size $O(n)$ and a hash function with access time $O(1)$ can be constructed for S in expected time $O(n)$.