Hashing

Anil Maheshwari

anil@scs.carleton.ca
School of Computer Science
Carleton University
Canada

Outline

Introduction

Problem

Input

Universe $U = \{0, 1, 2, \ldots, m-1\}$ of m possible distinct keys.
$S =$ A subset of n records that have keys from U.
Records in S have distinct keys.

Operations on S

INSERT(S, x): $S \leftarrow S \cup \{x\}$
DELETE(S, x): $S \leftarrow S \setminus \{x\}$
SEARCH(S, k): Returns the record x if $key(x) = k$, otherwise NIL

Objective

Construct a hash map (a data structure)
$h : U \rightarrow [O(n)]$, where $n = |S|$.
For all subset of n keys of U, the number of memory access required for INSERT, DELETE, and SEARCH is $O(1)$.

Possible Approaches

BST

Binary Search Tree storing elements of S with respect to their keys.

Time/operation = $O(\log |S|)$ with $O(|S|)$ storage.

Possible Approaches

Direct Access Table $T[0, \ldots, m-1]$

Set $T[k] = \begin{cases} x \text{ if } x \in S \text{ and } key[x] = k \\ \text{NIL}, \text{ otherwise} \end{cases}$

INSERT(S, x): $T[key(x)] \leftarrow x$
DELETE(S, x): $T[key(x)] = \text{NIL}$
SEARCH(S, k): Return $T(k)$

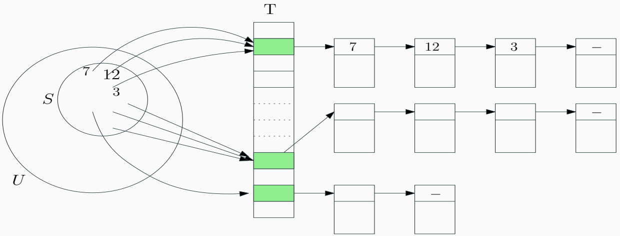
All operations cost $O(1)$ time and uses $O(|U|)$ storage.
Effective when $|S| \approx |U|$

Hash Tables

Hashing with Chaining

Hash Table

Hash function $h : U \to [0, \dots, n-1]$ maps keys of U to random slots in hash table T



Collisions

When multiple records (i.e. keys) map to the same slot of table T by the hash function h.

Chaining

Form a link list (chain) of all the records that are mapped to the same slot.

Analysis of Hashing with Chaining

Worst Case: All keys map to same slot $\implies$ SEARCH takes $\Theta(n)$ time.

Average Case: Assume that each key is equally likely to hash to any slot.

Load Factor $\alpha = \frac{|S|}{|T|} = \frac{n}{m}$= Average # Keys/slot

E[Time to search for an element $\notin S$] $= \Theta(1 + \alpha)$

If $n = O(m)$, $\alpha \in O(1)$ and E[Search Time] $= \Theta(1)$

Question: How to find hash functions that can distribute keys uniformly in slots of the table, irrespective of the distribution of keys?

Division Method

Assume size of table m is prime. If not, find a prime $p \in \{m, 2m\}$.

For any key $k \in U$, $h(k) = k \mod m$

Example: For $m = 101$, $h(220) = 220 \mod 101 = 18$

Issues: Sensitive to distribution of keys.

Multiplication Method

Let $0 < A < 1$.

$$h(k) = \lfloor m(kA \mod 1) \rceil = \lfloor m(kA - \lfloor kA \rfloor) \rfloor$$

Let $A = (\sqrt{5} - 1)/2 = 0.618$ (connected to Golden Ratio), $m = 2^8$, and $k = 220$.

$$\begin{aligned} h(220) &= \lfloor 2^8 (kA - \lfloor kA \rfloor) \rfloor \\ &= \lfloor 2^8 (220 * .618 - \lfloor 220 * 0.618 \rfloor) \rfloor \\ &= \lfloor 2^8 (135.96 - 135) \rfloor \\ &= \lfloor 2^8 (.96) \rfloor \\ &= 245 \end{aligned}$$

Issue: Sensitive to key distribution

Universal Hashing

Universal Hashing

Question: How to find hash functions that can distribute keys uniformly in slots of the table, irrespective of the distribution of keys?

+

This should hold even if an adversary knows your hash function!

Approach: Choose a hash function h uniformly at random from a family of hash functions $\mathcal{H}$.

Universal Family $\mathcal{H}$

Let $\mathcal{H}$ be a finite collection of hash functions from $U \to \{0, \ldots, m-1\}$. The family $\mathcal{H}$ is universal if $\forall x, y \in U$, $x \neq y$, $|h \in \mathcal{H} : h(x) = h(y)| = \frac{|\mathcal{H}|}{m}$

Equivalently,

Universal Family $\mathcal{H}$

If h is chosen uniformly at random from $\mathcal{H}$, $Pr[h(x) = h(y)] = \frac{1}{m}$.

Expected Number of Collisions

Claim

Choose $h \in \mathcal{H}$ uniformly at random. Use h to hash the records corresponding to the set of n keys of the universe U into m slots of the table. For a given record x, expected number of collisions with x in the table is $< \alpha = \frac{n}{m}$ = Load Factor

Proof: For each record y corresponding to the set of n keys, define an indicator r.v. $I_y = \begin{cases} 1, & \text{if } h(x) = h(y) \\ 0, & \text{otherwise} \end{cases}$

$E[I_y] = Pr(I_y = 1) = \frac{1}{m}$

Define $C = \sum_{y, y \neq x} I_y$ = Total number of collisions with x

$E[C] = E[\sum_{y, y \neq x} I_y] = \frac{n-1}{m} < \alpha$

$\square$

$\implies$ Cost per INSERT, SEARCH, DELETE operation $\approx \alpha$

Construction of a Universal Hash Family

A Universal Hash Family

Let m be prime, otherwise replace it by a prime in the range $\{m, 2m\}$. Let $a = (a_0, a_1, a_2, \ldots, a_r)$ be a $(r+1)$-digit base m number, where each $a_i \in \{0, 1, 2, \ldots, m-1\}$ is chosen uniformly at random. Define m^{r+1} hash functions indexed by a as the hash family $\mathcal{H}$

Express a key k as a $(r+1)$-digit number in base m, i.e. $k = (k_0, k_1, k_2, \ldots, k_r)$, where $0 \leq k_i \leq m-1$.

How does a hash function $h_a \in \mathcal{H}$ map key k to an index the table?

$$h_a(k) = a \cdot k \mod m = \sum_{i=0}^{r} a_i k_i \mod m$$

Universality of $\mathcal{H}$

$\mathcal{H}$ is Universal

The set of hash functions $\mathcal{H} = \{h_a(k)\}$ is universal.

Proof: To show universality, we need to show that the number of hash functions in $\mathcal{H}$ that map any two distinct keys k and l to same slot is $\leq \frac{|\mathcal{H}|}{m}$

Let $k = (k_0, \ldots, k_r)$ and $l = (l_0, \ldots, l_r)$ be the base m representation of k and l.

Since $k \neq l$, $\exists$ and index i such that $k_i \neq l_i$

WLOG, let $i = 0$, i.e. $k_0 \neq l_0$

Let us estimate for how many hash functions $h_a \in \mathcal{H}$, $h_a(k) = h_a(l)$.

For that to happen, $\sum\limits_{i=0}^{r} a_i k_i \equiv \sum\limits_{i=0}^{r} a_i l_i \pmod m$

$$\implies \sum_{i=0}^{r} a_i(k_i - l_i) \equiv 0 \mod m,$$

$$\Leftrightarrow a_0(k_0 - l_0) + \sum_{i=1}^{r} a_i(k_i - l_i) \equiv 0 \mod m$$

Proof contd.

$$\Leftrightarrow a_0(k_0 - l_0) + \sum_{i=1}^{r} a_i(k_i - l_i) \equiv 0 \mod m$$

$$\Leftrightarrow a_0(k_0 - l_0) \equiv -\sum_{i=1}^{r} a_i(k_i - l_i) \mod m$$

$$\Leftrightarrow a_0 \equiv \left(-\sum_{i=1}^{r} a_i(k_i - l_i)\right)(k_0 - l_0)^{-1} \mod m$$

Number Theory Fact

Let m be prime. For any non-zero $x \in Z_m$, $\exists$ a unique $z^{-1} \in Z_m$ such that $zz^{-1} \equiv 1 \mod m$

$\implies$ For k and l to hash to same slot,
$a_0 \equiv \left(-\sum_{i=1}^{r} a_i(k_i - l_i)\right)(k_0 - l_0)^{-1} \mod m$.

How many choices of $a's$ can satisfy the above?

We have m choices for each of $a_1, \ldots, a_r$, and only one choice for a_0.

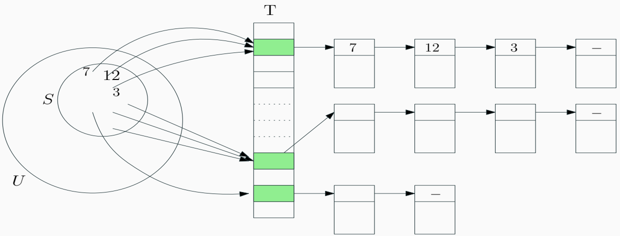
$\implies$ $\#a's$ satisfying $(h_a(k) = h_a(l))$ = $m^r = \frac{|\mathcal{H}|}{m}$

$\square$

Perfect Hashing

Membership Queries for a Static Set

Given n-keys, construct a static hash table of size $m = O(n)$ such that SEARCH takes $O(1)$ time in the worst case.



2-Level Scheme

1st Level: Use a random hash function from universal family to map keys into a table of size n.

2nd Level: If s_i elements are mapped to slot i of 1st level table, create a secondary Hash Table for these elements of size s_i^2 using another random hash function from universal family.

E[# Collisions]

Expected number of collisions when n items are hashed to a table of size $m = n^2$ by a random hash function h from a universal family of hash functions is $< \frac{1}{2}$.

Proof: For any pair of keys x and y, $Pr[h(x) = h(y)] = \frac{1}{m}$.

We have $\binom{n}{2}$ pairs.

$E[\#\text{Collisions}] = \frac{1}{m}\binom{n}{2} = \frac{n(n-1)}{2m} < \frac{1}{2}$

$\square$

Pr(No Collisions)

Probability that there no collisions is $> \frac{1}{2}$

Proof: Consider complementary event. By Markov's inequality ($Pr(X \geq t) \leq \frac{E[X]}{t}$), we have that $Pr(\#\text{Collisions} \geq 1) \leq \frac{E[\#\text{Collisions}]}{1} < \frac{1}{2}$.

$\square$

An Identity

Let s_i be the number of elements hashed into the slot i of 1st level table. Then $E[s_i^2] = 2 - \frac{1}{n}$

Proof: Let $x_k = \begin{cases} 1, & \text{if key } k \text{ is placed in slot } i \text{ in level } 1 \text{ table} \\ 0, & \text{otherwise} \end{cases}$

Observe that $s_i = \sum_{k \in \text{key}} x_k$

$$E[s_i^2] = E\left[\left(\sum_{k=0}^{n-1} x_k\right)\left(\sum_{j=0}^{n-1} x_j\right)\right] = E\left[\sum_{k=0}^{n-1}\sum_{j=0}^{n-1} x_k x_j\right] =$$
$$E\left[\sum_{k=0}^{n-1} x_k^2 + \sum\sum_{k \neq j} x_k x_j\right]$$

Note $E[x_k^2] = \frac{1}{n}$, and for $j \neq k$, $E[x_k x_j] = E[x_k]E[x_j] = \frac{1}{n^2}$

Therefore, $E[s_i^2] = \sum_{k=0}^{n-1} \frac{1}{n} + \sum\sum_{j \neq k} \frac{1}{n^2} = 2 - \frac{1}{n}$

□

Analysis of Table Size

Size of 1st level Table = n.

E[Size of 2nd Level Table] = $E\left[\sum_{i=1}^{n} s_i^2\right]$

$$
\begin{aligned}
E\left[\sum_{i=1}^{n} s_i^2\right] &= \sum_{i=1}^{n} E[s_i^2] \\
&= \sum_{i=1}^{n} \left(2 - \frac{1}{n}\right) \\
&= O(n)
\end{aligned}
$$

2-level Hash Table Contd.

Expected Lookup Time:

E[Time for 1st Level + Time for 2nd Level]
$= 1 + O(1) = O(1)$

Expected Space Used:

E[Hash functions + 1st Level + 2nd Level]
$= (n+1) + n + \sum_{i=1}^{n} E[s_i^2] = O(n)$

Suppose E[Space Used] $\leq 6n$.
By Markov's inequality, Pr(Actual Space Used $> 12n) \leq \frac{6n}{12n} = \frac{1}{2}$

References

References

1. Probability and Computing (Chapter 13) by Mitzenmacher and Upfal, Cambridge Univ. Press 2005.
2. Introduction to Algorithms (Chapter 11), Cormen, Leiserson, Rivest and Stein, MIT Press 2009.