Hashing

Lecturer: Debmalya Panigrahi *Scribe: Tianqi Song, Tianyu Wang*

1 Overview

In this lecture, we study hashing.[1]

2 Hashing

We use hashing to solve the following problem: we have keys which we want to add, remove, and lookup. Keys belong to some universe. Hashing is useful when the number of keys stored is small compared to the universe, otherwise and array will work.

We now define the problem more formally:

Definition 1. *Let U be a universe. Let S be the hash table, $|S| = n$, $(n \ll U)$. A hash function h is a function which maps from U to the n slots in S.*

We first construct a simple hash function, h_1.

Definition 2. *Let U be a universe and n be the number of keys to be stored in the hash table. Split U into n contiguous sections, and assign key k an index in the hash table as follows:*

$$h_1(x) = i, \text{ where } x \text{ belongs to the } i\text{th section of } U.$$

Our goal is to avoid *collisions*.

Definition 3. *Let $x, y \in U$ and h be a hash function. x and y collide if $h(x) = h(y)$ and $x \neq y$.*

Collisions are unavoidable, but should be minimized. For hash function h_1, the fraction of pairs x, y which collide is

$$\frac{n\binom{\frac{|U|}{n}}{2}}{\binom{|U|}{2}} = \Theta(\frac{1}{n}).$$

For any way we divide up universe, $\frac{|U|^2}{n}$ is the fewest number of total possible collisions, because $\frac{\Sigma_{i=1}^{n} x_i^2}{|U|^2} \geq \frac{1}{n}$, where x_i is the number of elements in U that are mapped to key i. We want to find a hash function which satisfies the following property:

Property 1. *For all $x, y \in U$ and $x \neq y$,* $\Pr[h(x) = h(y)] \leq \frac{1}{n}$.

Note that h_1 does not satisfy Property 1, since for any x, y in the same section of U, $\Pr[h(x) = h(y)] = 1$. It is clear that no deterministic algorithm can satisfy Property 1, so we must use randomization. We define h_2 with this in mind.

[1]Some materials are from a previous note by Samuel Haney for this class in Fall 2014.

Definition 4. *Let U be a universe and n be the number of keys to be stored in the hash table. Assign $h_2(x)$*

$$h_2(x) = X, \text{ where } X \text{ is sampled uniformly at random from } \{1, \ldots, n\}.$$

Claim 1. *Hash function h_2 satisfies Property 1.*

Proof.

$$\begin{aligned} \Pr[x, y \text{ collide}] &= \Pr[h(x) = h(y)] \\ &= \sum_{k \in S} \Pr[h(x) = h(y) = k] \\ &= \sum_{k \in S} \frac{1}{n^2} \\ &= \frac{1}{n}. \end{aligned}$$

□

The problem with h_2 is that we cannot efficiently find where each key is stored. We would need to store the location of each key in the hash table, which is exactly the problem we are trying to solve. Therefore, we would also like our hash functions to satisfy the following property.

Property 2. *h can be stored succinctly and $h(x)$ can be recovered efficiently.*

We will propose one more hashing scheme which will satisfy both Property 1 and Property 2. First, define the concept of $field$ as follows:

Definition 5 (Field)**.** *A field F is a set of elements equipped with operations $+$ and $\cdot$ such that the following axioms hold:*

1. *(closure under multiplication) $a \cdot b \in F$ for all $a, b \in F$*
2. *(closure under addition) $a + b \in F$ for all $a, b \in F$*
3. *(commutativity of addition) $a + b = b + a$ for all $a, b \in F$*
4. *(commutativity of multiplication) $a \cdot b = b \cdot a$ for all $a, b \in F$*
5. *(existence of additive identity) There is an element $0 \in F$ (additive identity), such that $a + 0 = a$ for all $a \in F$.*
6. *(existence of additive inverse) For all $a \in F$, there is an element $-a \in F$, such that $a + (-a) = 0$.*
7. *(existence of multiplicative identity) There is an element $1 \in F$ and $1 \neq 0$, such that $a \cdot 0 = a$ for all $a \in F$*
8. *(existence of multiplicative inverse) For all $a \in F$ and $a \neq 0$, there is an element $a^{-1} \in F$, such that $a \cdot a^{-1} = 1$.*
9. *(distributivity of multiplication over addition) For all $a, b, c \in F$, $(a + b) \cdot c = a \cdot c + b \cdot c$.*

Examples of fields include the set of real numbers equipped with addition and multiplication, the set of complex numbers equipped with addition and multiplication, and the set of rational numbers equipped with addition and multiplication. In addition, each prime number p gives rise to a prime field.

Definition 6 (Prime Field)**.** *Let F_p be the set $\{0,1,\ldots,p\}$, where p is prime, with operations $+$ and $\cdot$ defined as follows:*

- $a+b \stackrel{\mathsf{def}}{=} (a+b) \mod p$, *and*
- $a\cdot b \stackrel{\mathsf{def}}{=} (a\cdot b) \mod p$.

Note that F_p is closed under $+$ and $\cdot$, and all the axioms are satisfied. In addition, The prime field F_p $a\cdot(b+c) \stackrel{\mathsf{def}}{=} (a\cdot b+a\cdot c) \mod p$ for all $a,b,c \in F_p$

Definition 7. *Let U be a universe and n be the number of keys to be stored in the hash table. Assume $n=p$ is prime. For $x \in U$, let $x_1,\cdots,x_k$ be the digits of x when written in base p (i.e. $x_i \in F_p$, and $x=\sum_{i=1}^{k} x_i \cdot p^{k-1}$). Then,*

$$h_3(x) = \left(\sum_{i=1}^{k} a_i \cdot x_i\right) \mod p,$$

where a_i is chosen uniformly at random from F_p, for all i.

Claim 2. *Hash function h_3 satisfies Property 1 and Property 2.*

Proof. First, note that Property 2 is satisfied. Only $a_1,\ldots,a_k$ need to be stored, and $h(x)$ is computed with a simple sum. Proving Property 1 is more challenging.

Suppose $x \neq y$ and $x,y \in U$.

$$\begin{aligned}\Pr[h(x)\neq h(y)] &= \Pr\left[\left(\sum_i a_i\cdot x_i\right) \equiv \left(\sum_i a_i\cdot y_i\right) \mod p\right] \\ &= \Pr\left[\sum_i (a_i x_i - a_i y_i) \equiv 0 \mod p\right]. \qquad (1)\end{aligned}$$

by definition of h. Because $x \neq y$, there is an index j such that $x_j \neq y_j$ (i.e., $(x_j - y_j) \neq 0$). Continuing from 1, we have

$$\Pr\left[\sum_i (a_i x_i - a_i y_i) \equiv 0 \mod p\right] = \Pr\left[\sum_{i\neq j} (a_i(x_i-y_i) + a_j(x_j-y_j)) \equiv 0 \mod p\right]. \qquad (2)$$

Next, fix all a_i for $i \neq j$, and let

$$\alpha = \left(-\sum_{i\neq j} a_i(x_i-y_i)\right) \mod p.$$

α is a fixed number, and $\alpha \in F_p$. Let $\beta = x_j - y_j$ and note that $\beta \neq 0$. Continuing 2 gives

$$\begin{aligned}\Pr\left[\sum_{i\neq j} (a_i(x_i-y_i) + a_j(x_j-y_j)) \equiv 0 \mod p\right] &= \Pr[a_j(x_j-y_j) \equiv \alpha \mod p] \\ &= \Pr[a_j\beta \equiv \alpha \mod p].\end{aligned}$$

Multiplying all the elements of a field by some set nonzero element gives a permutation of the field. Since a_j is chosen uniformly at random from F_p, we get

$$\Pr[a_j\beta \equiv \alpha \mod p] = \frac{1}{p}.$$

Therefore, Property 1 is satisfied. $\square$

Definition 8. *A collection of hash functions H is universal if $\forall x, y \in U$ and $x \neq y$, $\Pr_{h\in H}[h(x) = h(y)] \leq \frac{1}{n}$ where h is chosen uniformly at random from H.*