COMPUT 675 Randomized Algorithms	Fall 2005

Lecture 16: November 1

Lecturer: Mohammad R. Salavatipour *Scribe: Bruce Fraser*

16.1 Hashing

Consider a universe of objects U, where $|U| = m$ is very large. The *set* abstract data type implements $S \subset U, |S| = s << m$, and has the following operations:

1. `MakeSet(S)`
2. `Insert(i,S)`
3. `Delete(i,S)`
4. `Find(i,S)`

The Set ADT can be implemented with balanced trees (red-black trees, for instance) such that each operation has a running time in $\Theta(\log n)$ in the worst case. However, using hashing we can achieve $O(1)$ expected run time per operation.

Herein we assume $\log m$ = medium word size (for example 32 bits).

Formally, we have a hash table $T[0, ..., n-1]$ and a hash function $h : U \to T$, we store $x \in U$ in location $h(x)$ of T. Typically, the hash table size n, is much smaller than the size m of the universe.

16.1.1 Collisions

Definition 16.1 *A hash function $h : U \to T$ is perfect for set $S \subset U$ if it is collision free (i.e. if $x, y \in S$, $x \neq y \Rightarrow h(x) \neq h(y)$).*

For any fixed set S there is perfect hash function but if if $n < m$, there is no hash function that is perfect for every set S. What if the hash function is completely random?

If $|S| = n$, and hash function is uniformly random, analysis of the collision problem reduces to the Balls and Bins problem. The expected number hashings per bucket is 1, and the maximum number of collisions for any bucket is $\Theta\left(\frac{\ln n}{\ln \ln n}\right)$. But we are more interested in expected number of collisions per bucket.

Definition 16.2 *A family H of hash functions $h : U \to T$ is 2-universal if for all $x, y \in U (x \neq y)$ and for h selected uniformly randomly from H implies that $Pr[h(x) = h(y)] \leq \frac{1}{n}$.*

Note that this is the same probability that a truly random hash function would give. In fact, most construction of 2-universal hash functions imply a stronger property that is called strongly 2-universal.

Definition 16.3 *A family H of hash functions is strongly 2-universal if for $x \neq y \in U$ and all $n_1, n_2 \in \{0, ..., n-1\}$*

$$Pr[h(x) = n_1 \text{ and } h(y) = n_2] = \frac{1}{n^2}$$

.

A natural question would be: Can we get a better family of hash functions? It turns out that the answer is no (at least not buy much):

Lemma 16.4 *For any family of hash functions $h : U \to T, \exists x, y \in U$ such that $Pr[h(x) = h(y)] \geq \frac{1}{n} - \frac{1}{m}$.*

Theorem 16.5 *Consider any seq. of operations with at most s inserts on a hash table of size n, where h is selected uniformly randomly from a family H of 2-universal hash functions. Then the expected cost of each operation is $\leq 1 + \frac{s}{n}$.*

Proof: Say on an operation on element $x \in U$, let Z be the number of elements in $h(x)$. Cost is $1 + Z = \sum_{y \in U} Z_y$ (Where $Z_y = 1$ if y is mapped to $h(x)$).

$$E[Z] = E[\sum_{y \in U} Z_y] = sPr[h(y) = h(x)] \leq \frac{s}{n}$$

■

Family of all hash functions is 2-universal. However, that family is much too large to handle, with a size of n^m. What follows is a procedure for constructing a 2-universal family.

Pick a prime number p, $m \leq p \leq 2m$, $m = |U|$. Let $\mathbb{Z}_p = \{0, 1, ..., p-1\}$. $\forall a, b \in \mathbb{Z}_p$ with $a \neq 0$ we define a hash function $h_{ab} : U \to \mathbb{Z}_p$ as follows: $h_{ab}(x) = ((ax + b) \mod p) \mod n$. Similarly, we define a family $H = \{h_{ab} | a, b \in \mathbb{Z}_p, a \neq 0\}$.

Is H 2-universal?

Fact: There are $p(p-1)$ functions in H and each function only needs $O(\log m)$ bits to be represented.

Theorem 16.6 *H is a 2-universal family.*

Proof: For any $x, y \in \mathbb{Z}_p$ need to bound,

$$\Pr[h_{ab}(x) = h_{ab}(y)] \text{ if } x \neq y$$

Let's define $h_{ab}(x) = g_{ab}(y) \mod n$ where $g_{ab}(x) = (ax + b) \mod p$. For $h_{ab}(x) = h_{ab}(y)$ we must have $g_{ab}(x) = g_{ab}(y) \mod n$. Let $r, s \in \mathbb{Z}_p$. We claim:

$$\Pr[g_{ab}(x) = rand g_{ab}(y) = s] = \begin{cases} 0 & \text{if } r = s \\ \frac{1}{p(p-1)} & otherwise \end{cases}$$

If $g_{ab}(x) = r$ and $g_{ab}(y) = s$ then $ax + b = r \mod p$ $ay + b = s \mod p$. TSo $a = \frac{r-s}{x-y}$. This system has a unique solution for a and b in $\mathbb{Z}_p$, since $x \neq y$, a is non-zero if and only if $r \neq s$. For exactly one a, b: $g_{ab}(x) = r$, $g_{ab}(y) = s$. Since H has $p(p-1)$ functions and we pick h u.r. from H

$$\Pr[g_{ab}(x) = r \wedge g_{ab}(y) = s] \leq \frac{1}{p(p-1)}.$$

We have $h_{ab}(x) = h_{ab}(y)$ if and only if $r = s \mod n$. Therefore:

$$\Pr\left[h_{ab}(x) = h_{ab}(y)\right] = \frac{1}{p(p-1}) \times |\{(r,s)|r \neq s \text{ and r = s} \mod \text{n}\}|$$

For each choice of r, there are $\lceil \frac{p}{n} \rceil - 1$ other s s.t. $r = s \mod n$. Therefore the set in the right-hand-side of above has sizes at most $p\left(\lceil \frac{p}{n} - 1\right) \leq \frac{p(p-1)}{n}$. Thus $\Pr[h_{ab}(x) = h_{ab}(y)] \leq \frac{1}{p(p-1)} \cdot \frac{p(p-1)}{n} = \frac{1}{n}$. ■

16.1.2 Static Dictionaries

Say the set S is fixed (given in advance) and we only are concerned with the `find` operation. Our goal is to find an algorithm with a worst case time of $O(1)$.

Definition 16.7 *A family H of hash functions is perfect if for every subset $S \subset U$ there is a hash $h \in H$ such that h is perfect for S.*

Theorem 16.8 *There is a perfect family H of size $O(ne^n \ln m)$ such that H is perfect for every set of size $|S| \leq n$.*

Proof: Pick a random hash function $h : \mathcal{U} \to \mathcal{N}$.

$$\Pr[h \text{ being perfect }] = \frac{n!}{n^n} \sim \frac{1}{e^n},$$

by Stirling's approximation. If we pick $\frac{n^n}{n!} = e^{n+1}$ functions,

$$\Pr[\text{ all being non-perfect }] \leq \frac{1}{e}$$

We need to find the value of t such that for te^{n+1} randomly chosen functions

$$\Pr[\text{ all are non-perfect }] \leq \frac{1}{e^t}$$

Setting $t = ne^n \ln m$, $Pr[allarenon-perfect] \leq \frac{1}{m^n}$.

The number of sets of size n is less than or equal to m^n.

$$\Pr[\text{All are not perfect for at least a set } S] \leq m^n \left(\frac{1}{2m^n}\right) \leq \frac{1}{2}$$

■

This result is only existential. There is another way of building perfect has families (due to Fredman, Komlós, and Szemerédi 84). Consider random hashing, how big s can be w.r.t. n without having any collisions? Consider Balls and Bins, with s balls n bins.

$$\text{E[number of collisions]} = \text{E}\left[\sum_{i,j} c_{ij}\right] = \binom{s}{2}\frac{1}{n} = \frac{s^2}{2n}.$$

If $s = \sqrt{n}$ then with probability of at least $\frac{1}{2}$ we don't have any collisons.

We employ a second hash function. Loosely speaking, for those elements that collide, use a second hash function. Suppose the primary hash function creates buckets of size $b_1, \ldots, b_r$. Our scheme is to use two layers of hash tables to handle collisions. Choose a secondary hash function from a 2-universal family with table sizes b_i^2. This has probability of at least 1/2 of being collision-free. Total size of hash tables in both layers is size of primary plus $\sum_i b_i^2$.

Lemma 16.9 *If we use a 2-universal hash family to hash a set into a table of size $n > |S|$, for bucket sizes b_i:*

$$\Pr\left[\sum b_i^2 \geq 4s\right] \leq \frac{1}{2}$$

.

Proof: Note that $b_i^2 = b_i + 2\binom{b_i}{2}$. So:

$$\begin{aligned}
\sum b_i^2 &= \sum b_i + 2\sum\binom{b_i}{2} \\
&\leq s + 2\binom{s}{2}\frac{1}{n} \\
&\leq s + \frac{s^2}{2n} \leq 2s
\end{aligned}$$

Now Markov's Inequality shows that $\Pr[\sum_i b_i^2 > 4s] \leq \frac{1}{2}$. ■

So if we pick the primary hash function from a 2-universal family with $n \geq s$ then with probability $\geq 1/2$ we have $\sum_i b_i^2 \leq 4s$. Once we have our primary and our secondary hash functions, we have a 2-level hash function that has $O(1)$-time for find.