6 Randomization in Hashing

So far we have been satisfied with expected case behaviour of hashing. Can we say something non-trivial about the worst-case behaviour? It seems unlikely until we make some assumptions. One model worth studying is the case where the set of the in the table never changes. This is called as the *static keys* scenario. There are situations where this happens. Examples include the set of filenames on a read-only disk, the set of keywords in a programming language, $[0, 100]$, and so on. In this case, it turns out that we can do much better even in the worst case.

Let us call a hashing technique *perfect hashing* if the time units spent for a search is $O(1)$ in the worst-case. We present such a technique below.

We use a two-level hashing scheme. The n keys are first hashed to a primary hash table of size m using a hash function h from a universal family. To resolve collisions, we create a secondary hash table at each colliding slot. Each such secondary hash table is assocaited with a hash function different from h. By choosing m, h, the secondary hash functions, and the size of the secondary hash tables carefully we can guarantee that there would be no collisions.

An example is given below.

We show two results that help us in choosing the parameters.

6.1 Universal Hashing

Consider the case of hashing the variables in a program by a compiler. If the program has variables of the kind $x, x1, x2$, and similarly close names, some hash functions might have trouble in spreading these far apart. This results in lots of collisions. In fact, given any hash function, n keys can be chosen so that all the keys map to a single value. How to avoid such pathological scenarios?

One way to proceed is to choose the hash function randomly from a possible set of functions. This choice shall be made independent of the keys. This is called as *universal hashing*. The advantage of randomizing the choice of the hash function is to guarantee certain provably good performance on average for any input.

Let $\mathcal{H}$ be a finite collection of hash functions so that each $h \in \mathcal{H}$ maps U to $[0, m-1]$. The collection $\mathcal{H}$ is said to be *universal* if for every pair of distinct keys $k, \ell \in U$, the number of hash functions h in $\mathcal{H}$ so that $h(k) = h(\ell)$ is at most $\mathcal{H}/m$. Put in words, the probability that k and ℓ collide under a random chosen function from $\mathcal{H}$ is same as the probability of picking k and ℓ from $[0, m-1]$ uniformly and independently at random.

First let us look at the implications of such a collection.

Theorem 6.1 *Let h be chosen uniformly at random from $\mathcal{H}$ and used to hash n keys to a table of size m using chaining to resolve collisions. Let n_i denote the length of the list at slot i. Then, the expected length of $n_{h(k)}$ is α if k is not in the table and $1+\alpha$ if k is in the table.*

Proof. The difference in the above theorem to the earlier theorem is that here the expectation is with respect to the choice of the hash function and not with respect to the distribution of the input.

Let $X_{k\ell}$ denote the random variable so that $X_{k\ell} = 1$ if $h(k) = h(\ell)$ and 0 otherwise. Then, $E[X_{kl}] \leq 1/m$ by the universal nature of $\mathcal{H}$.

For key k, let Y_k be the random variable that counts the number of keys in the table other than k map to k. By definition, $Y_k = \sum_{\ell \in T, \ell \neq k} X_{k\ell}$ and

$$E[Y_k] = E[\sum_{\ell \in T, \ell \neq k} X_{k\ell} = \sum_{\ell \in T, \ell \neq k} E[X_{k\ell}] = \sum_{\ell \in T, \ell \neq k} 1/m$$

Now we have two cases.

- k **not in** T: In this case, $E[Y_k] = \sum_{\ell \in T} 1/m = \alpha$. Also, $E[n_{h(k)}] = E[Y_k] = \alpha$.
- k **in** T: In this case, $E[Y_k] = \frac{n-1}{m}$ and $n_{h(k)} = Y_k + 1$. Hence, $E[n_{h(k)}] = 1 + \frac{n-1}{m} = 1 + \alpha - (1/m) \leq 1 + \alpha$.

□

So we have at least characterize the properties and capabilities of a universal family of hash functions. But does there exist such a family? Fortuanately, there is one.

A Universal Class of Hash Functions Let p be a prime number and denote by $\mathbb{Z}_p$ the set $\{0, 1, 2, \cdots, p-1\}$ and by $\mathbb{Z}_{p^*}$ the set $\mathbb{Z}_p \setminus \{0\}$. Let the size of the table be so that $m < p$. We define a hash function below.

$$h_{a,b}(k) = ((ak + b) \bmod p) \bmod m$$

The family we define is

$$\mathcal{H} = \{h_{a,b} | a \in \mathbb{Z}_{p^*}, b \in \mathbb{Z}_p\}$$

An example for h is given below. The size of $\mathcal{H}$ is $p(p-1)$. We now show that the above class is universal.

Theorem 6.2 *The class $\mathcal{H}$ defined above is universal.*

Proof. We show that the probability that two distinct keys k and ℓ collide under a function h chosen uniformly and independently at random from $\mathcal{H}$ is at most $1/m$.

For $k, \ell \in \mathbb{Z}_p$, let:

$$\begin{aligned} r &= ak + b \bmod p, \text{ and} \\ s &= a\ell + b \bmod p \end{aligned}$$

Since p is a prime, it holds that $r \neq s$. [Otherwise, $a(k-\ell) = 0 \bmod p$ with $0 < a, k - \ell < p$ implying that p is not prime.

So $h(k) \neq h(\ell)$ at the mod p level. Moreover, for each pair (a, b) chosen appropriately, we get a different pair (r, s) as we can solve for a and b given r and s for distinct k and ℓ.

So we only have to worry about collisions at the mod m level. Given distinct r and s, picked uniformly at random (as (a, b) is chosen u.a.r.) and independently from $\mathbb{Z}_{p^*} \times \mathbb{Z}_p$, we want to compute the probability that $r \equiv s (\bmod\, m)$. The number of such r's is in fact, $s, s + m, s + 2m, \cdots$,. Their number is at most $\lceil p/m \rceil - 1 \leq \frac{p}{m}$.

So the probability that r collides with s is at most $1/p \cdot p/m = 1/m$. □

Thus, fortuanately, universal families exist.

Theorem 6.3 *Let n keys be stored in a table of size $m = n^2$ using a hash function chosen u.a.r. from a universal family of hash functions. Then, the probability of there being any collisions is at most $1/2$.*

Proof. Let h be chosen as above. For each pair of distinct keys k and ℓ let $X_{k\ell}$ be a random variable that takes a value 1 if k and ℓ collide under h and 0 otherwise. Then, $E[X_{k\ell}] < 1/m$.

Let the random variable X count the number of colliding pairs. Then, $X = \sum_{k,\ell} X_{k\ell}$. And,

$$E[X] = E[\sum_{k,\ell} X_{k\ell} = \sum_{k,\ell} E[X_{k\ell}] = \binom{n}{2} \frac{1}{n^2} = \frac{n^2 - n}{2n^2} < 1/2.$$

Now applying Markov inequality, $\Pr[X \geq 1] \leq E[X] = 1/2$. □

However, the size of the primary table cannot be as big as n^2. So we cannot use the above theorem directly. But let us look at the following theorem.

Theorem 6.4 *Let $m = n$ and h be a hash function chosen from a universal family u.a.r. If n keys are hashed then,*

$$E\left[\sum_{i=0}^{m-1} n_i^2\right] < 2n.$$

Proof.

$$\begin{aligned} & E\left[\textstyle\sum_{i=0}^{m-1} n_i^2\right] \\ = \; & E\left[\textstyle\sum_{i=0}^{m-1} \left(n_i + 2\binom{n_i}{2}\right)\right] \\ = \; & E\left[\textstyle\sum_{i=0}^{m-1} n_i\right] + 2E\left[\textstyle\sum_{i=0}^{m-1} \binom{n_i}{2}\right] \\ = \; & E[n] + 2E\left[\textstyle\sum_{i=0}^{m-1} \binom{n_i}{2}\right] \\ = \; & n + 2E\left[\textstyle\sum_{i=0}^{m-1} \binom{n_i}{2}\right] \end{aligned}$$

However, $\sum_{i=0}^{m-1} \binom{n_i}{2}$ is the number of collisions. The number of collisions is at most $\binom{n}{2} \cdot 1/n$ as h is chosen from a universal family and $m = n$. Thus,

$$E\left[\sum_{i=0}^{m-1} n_i^2\right] \leq n + \binom{n}{2} \cdot 1/m = n + \frac{n(n-1)}{2n} = 2n - 1 < 2n.$$

□

To put everything together, suppose that n_j keys hash to slot j. If the size of the secondary table is chosen to be n_j^2 then the total size of the secondary hash tables is at most $2n$ on expectation. With advanced techniques, we can show how to choose the function h and the secondary hash functions so that the actual size of the secondary tables is not far beyond the expected size.