Hash Functions

$h(x) = x \bmod m$

if $m = 2^r$ for some r

- easy to compute
- only works if the low order bits are evenly distributed

if $m = p$ for some prime p

- no better in principle
- example: all keys are multiples of p

Perfect Hashing?

- we could directly construct a perfectly random hash function i.e.

  $h(x)$ is chosen uniformly at random from $[0 \ldots m-1]$ for each x

certainly universal but impractical

Universal Hashing

- rather than hoping for our input to be 'average' we instead introduce randomness into our hash scheme.
  - we will show how to choose the hash function itself randomly
  - $\forall\ E[l(x)] = O(1 + \frac{n}{m})$
- To achieve this we construct a family H of hash functions such that for any $x, y \in U$ and some constant k

$$\Pr_{h \in H}(h(x) = h(y)) \leq \frac{k}{m} \quad \forall\ x, y \mid x \neq y$$

Theorem:

The property $\Pr_{h \in H}(h(x) = h(y)) \leq \frac{k}{n}$ is enough to bound $E[\ell(x)]$.

Proof:

- let T be a hash table with n entries and n slots
- let C_{xy} be an indicator random variable where $C_{xy} = 1$ iff $h(x) = h(y)$
- then $\ell(x) = \sum_{y \in T} C_{xy}$

- If h is drawn from a near-universal family H

$$E[C_{xy}] = Pr(h(x) = h(y)) \leq \begin{cases} 1 & \text{if } x = y \\ \frac{k}{m} & \text{otherwise} \end{cases}$$

$$E[\ell(x)] = E\left[\sum_{y \in T} C_{xy}\right]$$

$$= \sum_{y \in T} E[C_{xy}]$$

$$= \sum_{y \in T} \frac{k}{m}$$

$$= \frac{kn}{m} \in O\left(\frac{n}{m}\right)$$

Designing a Universal Hash Family

- choose a prime number p so that every possible key k is in the range $[0 \ldots p-1]$
- Let $Z_p = [0 \ldots p-1]$ and $Z_p^* = [1 \ldots p-1]$
- Since $|U|$ is greater than the size of the hash table it follows that $p > m$
- Define hash function h_{ab} for any $a \in Z_p^*$ and $b \in Z_p$ using the following

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

- The family of hash functions is

$$H_{pm} = \{ h_{ab} : a \in Z_p^* \text{ and } b \in Z_p \}$$

- H_{pm} contains $p(p-1)$ hash functions

Theorem

The class H_{pm} of hash functions is universal.

Proof:

- consider two keys k and ℓ from Z_p such that $k \neq \ell$
- for a given h_{ab} let

$$r = (ak + b) \bmod p$$

$$s = (a\ell + b) \bmod p$$

- Note that $r \neq s$. Observe that

$$r - s \equiv a(k - \ell) \bmod p$$

- since p is prime and both a and $(k-\ell)$ are non-zero their product must also be non-zero mod p Theorem 31.6 CLRS

- Since $r \neq s$ there are no collisions at the 'mod p' level
- Each of the $p(p-1)$ possible choices of a and b yield a different resulting r and s with $r \neq s$

$$a = ((r-s)((k-\ell)^{-1} \bmod p)) \bmod p$$

$$b = (r - ak) \bmod p$$

- Since there are $p(p-1)$ possible r, s pairs with $r \neq s$ there is a one-to-one correspondance between pairs a, b and r, s
- Thus if we pick a, b uniformly at random from $Z_p^* \times Z_p$ for any keys k, ℓ the resulting r, s is equally likely to be any pair of distinct values mod p.

- Therefore the probability that k and ℓ collide is equal to the probability that

$$r \equiv s \bmod m$$

where r, s are randomly chosen as distinct values mod p

- for a given r, of the $p-1$ possible values for s the number of values such that

$$s \neq r \quad \text{and} \quad s \equiv r \bmod m$$

is at most

$$\left\lceil \frac{p}{m} \right\rceil - 1 \leq \frac{p+m-1}{m} - 1$$

CLRS 19 54

$$= \frac{p-1}{m}$$

- The probability that s collides with r after mod m is at most

$$\frac{\frac{(p-1)}{m}}{p-1} = \frac{1}{m}$$

- Therefore for any distinct k, ℓ

$$P_r\left(h_{ab}(k) = h_{ab}(\ell)\right) \leq \frac{1}{m}$$

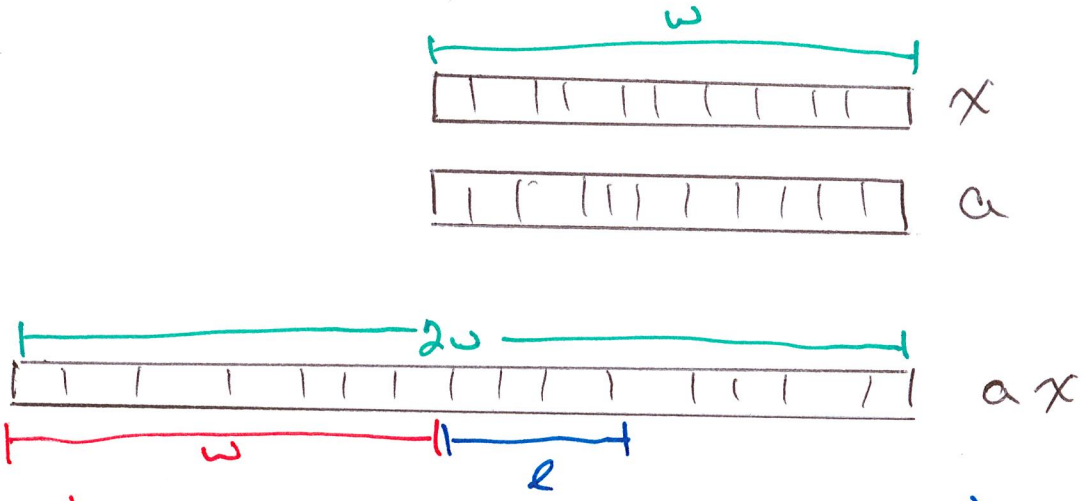
Universal Hashing #2

(Dietzfelbinger et al)

- This works using powers of 2
- Assume $U = [0 \dots 2^w - 1]$ and $m = 2^\ell$ for some integers w and ℓ
  - our goal is to hash w-bit integers into ℓ-bit integers
- Our hash functions will be indexed by the odd integers $a \in [1, 3, 5, \dots, 2^w - 1]$ and defined as

$$h_a(x) = \left\lfloor \frac{a x \bmod 2^w}{2^{w-\ell}} \right\rfloor$$

- To see what this function does it is easiest to look at the bit level



thrown away from the mod 2^w

division by a power of 2 is just bit shift so the lower order $w-\ell$ bits are discarded