

Chapter 6, Hash tables:

- ▶ General introduction; separate chaining (last week Monday)
- ▶ Open addressing (last week Wednesday)
- ▶ Hash functions (last week Friday)
- ▶ Perfect hashing (**Today**)
- ▶ Hash table performance (Wednesday)
- ▶ (Start Ch 7, Strings, Friday)

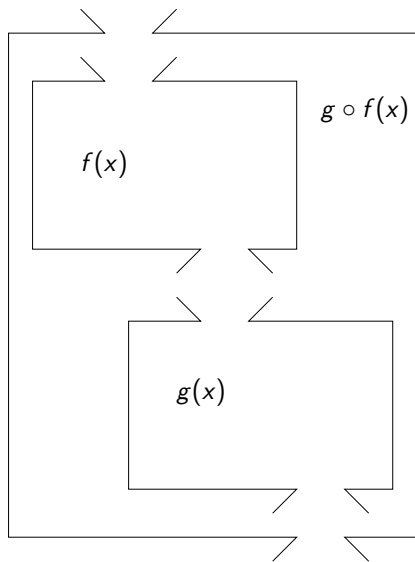
Today:

- ▶ Perfect hashing anticipated
 - ▶ Motivation
 - ▶ Goals
- ▶ Perfect hashing accomplished
 - ▶ Definition of universal hashing
 - ▶ Hash function class \mathcal{H}_{pm}
 - ▶ Theorems and proofs
- ▶ Perfect hashing applied
 - ▶ The design of a perfect hashing scheme
 - ▶ The given code for the project

A hashing scheme must reduce the occurrence of collisions and “deal” with them when they happen.

- ▶ *Separate chaining*, where $m < n$, deals with collisions by chaining keys together in a bucket.
- ▶ *Open addressing*, where $n < m$, deals with collisions by finding an alternate location.
- ▶ *Perfect hashing* deals with collisions by preventing them altogether.

This topic is parallel with the *optimal BST problem*: What if we knew the keys ahead of time? What if we got to choose the hash function based on what keys we have?



Let \mathcal{H} stand for a *class* of hash functions (a set of hash functions defined by some formula).

Let m be the number of buckets.

\mathcal{H} is *universal* if

$$\forall k, l \in \text{Keys}, |\{h \in \mathcal{H} \mid h(k) = h(l)\}| \leq \frac{|\mathcal{H}|}{m}$$

\mathcal{H} is universal if

$$\forall k, l \in \text{Keys}, |\{h \in \mathcal{H} \mid h(k) = h(l)\}| \leq \frac{|\mathcal{H}|}{m}$$

One particular *family* of *classes* of hash functions, given p , a prime number greater than all keys, and m , the number of buckets, is denoted \mathcal{H}_{mp} :

$$\mathcal{H}_{mp} = \{ h_{ab}(k) = ((ak + b) \bmod p) \bmod m \mid a \in [1, p) \text{ and } b \in [0, p) \}$$

Theorem \mathcal{H}_{pm} is universal.

Proof. Suppose p and m as specified earlier. Suppose $k, \ell \in \text{Keys}$, and $h_{ab} \in \mathcal{H}_{pm}$ (which implies supposing that $a \in [1, p)$ and $b \in [0, p)$).

Let $r = (a \cdot k + b) \bmod p$ and $s = (a \cdot \ell + b) \bmod p$

Subtracting gives us

$$\begin{aligned} r - s &\equiv (a \cdot k + b) - (a \cdot \ell + b) \pmod{p} \\ &\equiv a \cdot (k - \ell) \pmod{p} \end{aligned}$$

Now a cannot be 0 because $a \in [1, p)$. Similarly $k - \ell$ cannot be 0, since $k \neq \ell$. Hence $a \cdot (k - \ell) \neq 0$.

Since p is prime and greater than a , k , and ℓ , it cannot be a factor of $a \cdot (k - \ell)$. In other words, $a \cdot (k - \ell) \bmod p \neq 0$. By substitution, $r - s \neq 0$, and so $r \neq s$.

By another substitution, $(a \cdot k + b) \bmod p \neq (a \cdot \ell + b) \bmod p$.

Define the following function, given k and ℓ , which maps from (a, b) pairs to (r, s) pairs (formally, $[1, p) \times [0, p) \rightarrow [1, p) \times [0, p)$):

$$\phi_{k\ell}(a, b) = ((a \cdot k + b) \bmod p, (a \cdot \ell + b) \bmod p)$$

Now consider the inverse of that function.

$$\begin{aligned}\phi_{k\ell}^{-1}(r, s) &= (((r - s) \cdot (k - \ell)^{-1}) \bmod p, (r - ak) \bmod p) \\ &= (a, b)\end{aligned}$$

The existence of ϕ^{-1} implies that ϕ is a one-to-one correspondence. Hence for each (a, b) pair, there is a unique (r, s) pair. Since the pair (a, b) specifies a hash function, that means that for each hash function in the family \mathcal{H}_{pm} , there is a unique (r, s) pair.

There are $p-1$ possible choices for a and p choices for b , so there are $p \cdot (p-1)$ hash functions in family \mathcal{H}_{pm} . Likewise there are p choices for r , and for each r there are $p-1$ choices for s (since $s \neq r$). Thus we can partition the set \mathcal{H}_{pm} into p subsets by r value, each subset having $p-1$ hash functions.

For a given r , at most one out of every m can have an s that is equivalent to $r \pmod{m}$, in other words, at most $\frac{p-1}{m}$ hash functions.

Now sum that for all p of the subsets of \mathcal{H}_{pm} , and we find that the number of hash functions for which k and ℓ collide are

$$p \cdot \frac{p-1}{m} = \frac{p \cdot (p-1)}{m} = \frac{|\mathcal{H}_{pm}|}{m}$$

Therefore \mathcal{H}_{pm} is universal by definition. \square

Theorem [Probability of any collisions.] If $Keys$ is a set of keys, $m = |Keys|^2$, p is a prime greater than all keys, and $h \in \mathcal{H}_{pm}$, then the probability that any two distinct keys collide in h is less than $\frac{1}{2}$.

Proof. Suppose we have a set $Keys$, $m = |Keys|^2$, p is a prime greater than all keys, and $h \in \mathcal{H}_{pm}$.

Consider the number of pairs of unique keys. The number of pairs of keys is

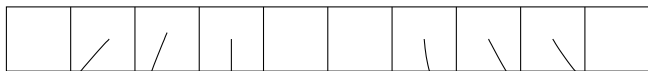
$$\binom{n}{2} = \frac{n!}{2! \cdot (n-2)!} = \frac{n!}{2 \cdot (n-2)!} = \frac{n \cdot (n-1) \cdot \cancel{(n-2)!}}{2 \cdot \cancel{(n-2)!}} = \frac{n \cdot (n-1)}{2}$$

Since \mathcal{H}_{pm} is universal, each pair collides with probability $\frac{1}{m}$. Multiply that by the number of pairs, and the expected number of collisions is

$$\begin{aligned}\frac{n \cdot (n-1)}{2} \cdot \frac{1}{m} &< \frac{n^2}{2} \cdot \frac{1}{m} \quad \text{since } n \cdot (n-1) < n^2 \\ &= \frac{n^2}{2} \cdot \frac{1}{n^2} \quad \text{since } m = n^2 \\ &= \frac{1}{2} \quad \text{by cancelling } n^2\end{aligned}$$

With the expected number of collisions less than one half, the probability there are any collisions is also less than $\frac{1}{2}$. \square

$$h(k) = (93, 0) \in \mathcal{H}_{101\ 10}$$



$$h_3(k) = (47, 22) \in \mathcal{H}_{101\ 4}$$

78	88		
----	----	--	--

$$h_8(k) = (0, 0) \in \mathcal{H}_{101\ 0}$$

95

$$h_2(k) = (56, 15) \in \mathcal{H}_{101\ 9}$$

73					68	39		
----	--	--	--	--	----	----	--	--

$$h_7(k) = (0, 0) \in \mathcal{H}_{101\ 0}$$

85

$$h_1(k) = (0, 0) \in \mathcal{H}_{101\ 0}$$

53

$$h_6(k) = (1, 100) \in \mathcal{H}_{101\ 4}$$

65	94		
----	----	--	--

Coming up:

Do **Open Addressing Hashtable** project (suggested by Wenesday, Apr 19)

Do **Perfect hashing** project (suggested by Wed, Apr 26)

Due **Today, Apr 17** (end of day)

Read Sections 7.(4 & 5)

(No practice problems or quiz)

Due **Wed, Apr 19** (end of day)

Read Section 7.6

Take quiz

Due **Fri, Apr 21** (end of day)

Read Section 8.1

Do Exercises 8.(4 & 5)

Take quiz

(Accompanying programming practice problem and quiz due next week Monday)