CS 223 Lec 7

- finished chaining example code

Properties of Hash Tables

"load" of hash table

$$\alpha = \frac{n}{m} \begin{array}{l} = \text{\# of elements} \\ = \text{\# of slots} \end{array}$$

feature of a good hash function: if it can maintain that every slot has length $\Theta(1+\alpha)$.

Types of Hash Fns

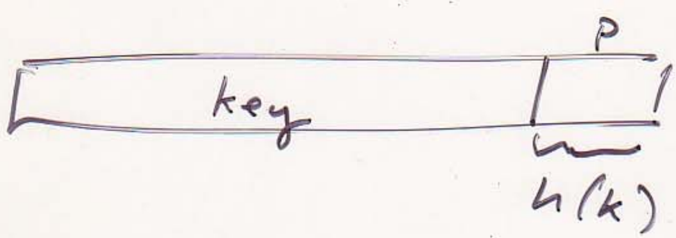
Division Method

$$h(k) = \underline{k \bmod m}$$

key $\qquad \in \{0, 1, 2, \dots m-1\}$

- avoid: m being a power of 2.

$$m = 2^p$$



Multiplication Method

$$h(k) = \lfloor m\,(kA \bmod 1) \rfloor$$

$$h : U \to \{0, 1, 2, \ldots, m-1\}$$

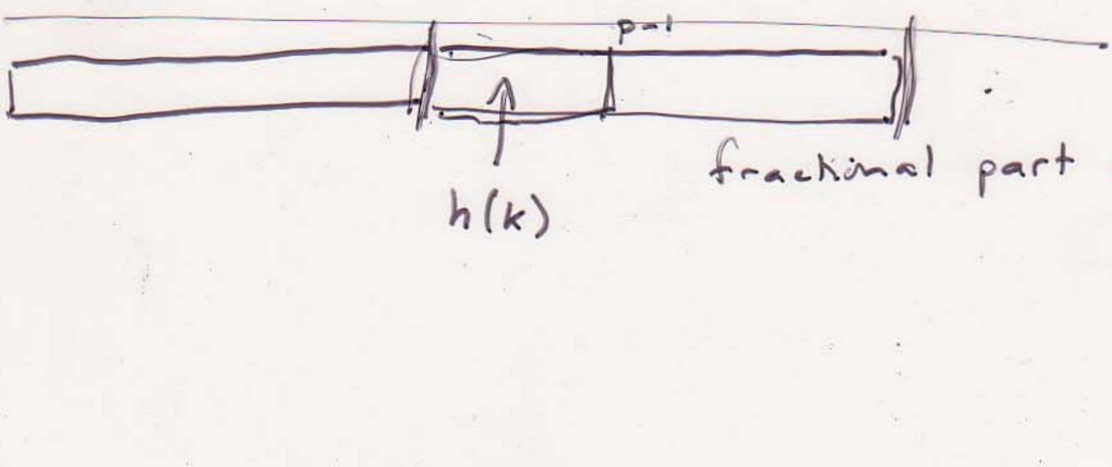
A constant

$0 < A < 1$

$$= kA - \lfloor kA \rfloor$$

(fractional part of kA)

nice feature: works well when m is a power of 2



$m = 2^p$

$A = s/2^w$

Knuth (one of forefathers of CS)

3.5

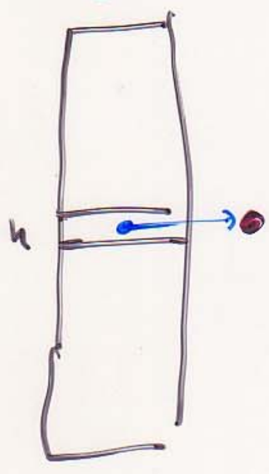
$$\Rightarrow \quad A = \frac{(\sqrt{5} - 1)}{2}$$

good choice.

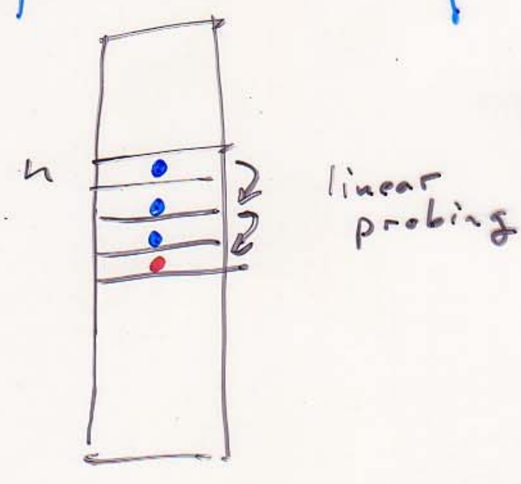
Open Addressing

Alternative to chaining
for handling collisions

chaining



open addressing



Formally, we require

$$\alpha < 1.$$

$$h : U \times \{0, 1, 2, \ldots, m-1\} \rightarrow \{0, 1, 2, \ldots, m-1\}$$

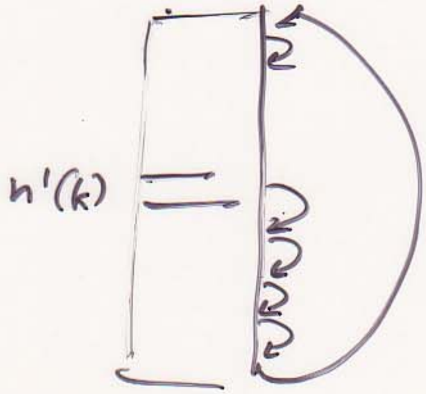
<u>require of h</u>

h must be <u>onto</u>

<u>Example</u> "Linear Probing"

$$h(k, i) = [h'(k) + i] \bmod m$$

h' is given hash fn



<u>Problem</u>
long runs

Double Hashing

$$h(k,i) = \left(h_1(k) + i \cdot \underline{h_2(k)}\right) \bmod \underline{m}$$

must be relatively prime

$\Downarrow$

note: ensures h is onto.

$$h_1(k) = k \bmod m$$

$$h_2(k) = 1 + (k \bmod m')$$

$$m' < m$$

(m is prime)

Issue: deletions

instead of setting deleted elements to null, set them to "deleted"

Application

digital signatures

$\boxed{\text{.doc}}$

$\Downarrow$ hash fn

<digital sig>

1024 bits

Common Hash Fn : MD5
(MIT)

2004 $\Rightarrow$ a collision with MD5 was found.