5112

11-16

Hash Tables

Part II

Can we have a hash table with no collisions?

What if we hash into a big array? Hash n items into m slots.

$$E\{\#\text{ of colliding pairs}\} = \sum_{i<j} \Pr\{h(x_i) = h(x_j)\}$$

$$\leq \sum_{i<j} \frac{1}{m} = \frac{n(n-1)}{2} \cdot \frac{1}{m} = \frac{n(n-1)}{2m}$$

If $m \geq n(n-1)$, then this is $\leq \frac{1}{2}$.

Markov's Inequality

If X is non-negative R.V., then

$$\Pr\{X \geq a\} \leq E\{X\}/a.$$

Proof. $E\{X\} = \sum_{i \geq 0} i \Pr\{X = i\} \geq \sum_{i \geq a} i \Pr\{X = i\}$

$$\geq \sum_{i \geq a} a \Pr\{X = i\}$$

$$= a \sum_{i \geq a} \Pr\{X = i\} = a \Pr\{X \geq a\}.$$

□

If $m \geq n(n-1)$ then $E\{\#\text{ of colliding pairs}\} \leq \frac{1}{2}$

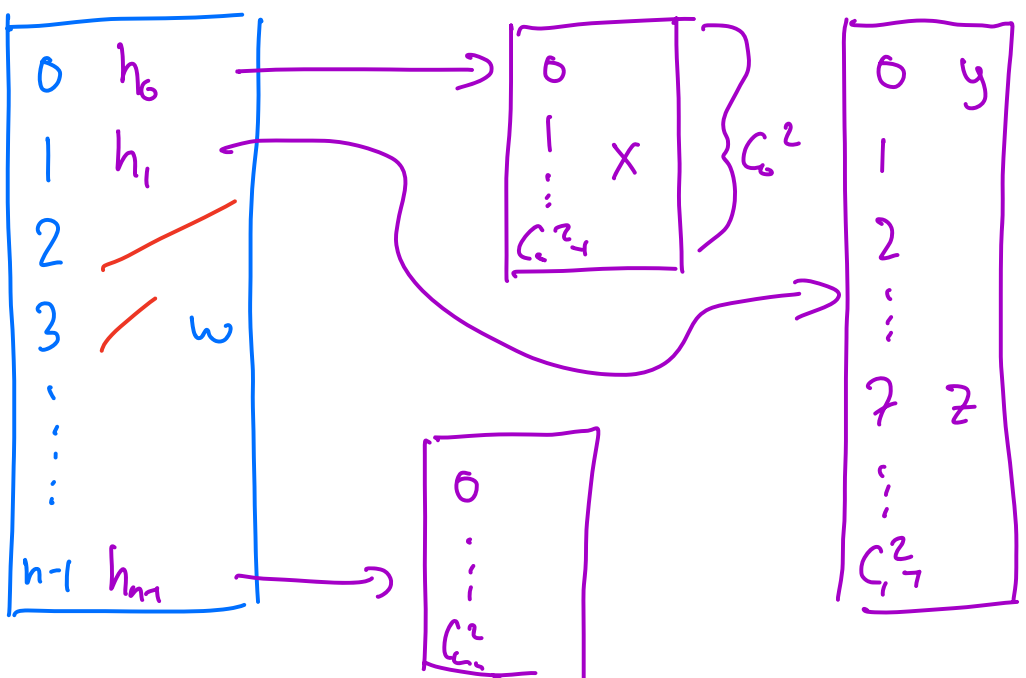
By Markov's inequality,

$$\Pr\{\#\text{ of colliding pairs} \geq 1\} \leq E\{\#\text{ of coll. pairs}\} \leq \frac{1}{2}.$$

FKS Perfect Hashing

Level 1: n items hash into n buckets

Level 2: Each bucket is collision-free hash table.



(c_i is the # of items in bucket i)

$c_1^2 \le 7$

FKS Perfect Hashing

Static hash table:

Construct(array & n items)

Query(key)

How to query:

Hash to level 1 slot

Hash to level 2 w/ h_ϵ.

If found, return

Else return "not found"

How to construct:

Loop through all the items and put them into lists,

Loop through all the lists:

For list t, choose a random h_t.

Try to build a hash table for the items in list t.

While there's a collision, choose a different h_ϵ, try again.

FKS Perfect Hashing

How to construct:

Loop through all the items and put them into lists, $-O(n)$

Loop through all the lists:

$O(1)$ times through the loop:

- For list t, choose a random h_t. $-O(1)$
- Try to build a hash table for the items in list t. $-O(c_t)$
- While there's a collision, choose a different h_t, try again.
  - Repeat w/ prob $\leq \frac{1}{2}$.

Σ the b_i loop completes in $\sum O(c_t) = O(n)$

$E\{\#\text{ of trials}\} \leq \frac{1}{2} + \frac{1}{2}(1 + E\{\#\text{ of trials}\})$

$\frac{1}{2}E\{\#\text{ of trials}\} \leq 1 \Rightarrow E\{\#\text{ of trials}\} \leq 2.$

= total expected construction time of $O(n)$.

FKS Perfect Hashing

How to query:

Hash to level 1 slot — $O(1)$

Hash to level 2 w/ h_{ϵ}. — $O(1)$

If found, return
Else return "not found" } $O(1)$

$O(1)$ queries deterministically!

FKS Perfect Hashing

What about the space?

Level 1 uses $O(n)$ space.

Level 2 use $\sum_t c_t^2$ space

	1	2	3	4	5
1	·	·	·	·	·
2	·	·	·	·	·
3	·	·	·	·	·
4	·	·	·	·	·
5	·	·	·	·	·

1 2 3 4 5 are in bucket t.

$$E\{c_t^2\} = \sum_{i,j} \Pr\{h(x_i) = t \wedge h(x_j) = t\}$$

$$E\left[\sum C_t^2\right] = \sum_{i,j,t} \Pr\{h(x_i) = t \wedge h(x_j) = t\}$$

$$= \sum_{i,j} \Pr\{h(x_i) = h(x_j)\}$$

$$\leq \sum_{i,j} \frac{1}{m} \quad \text{(by universality)}$$

$$= n^2/m$$

$$= n \quad \text{b/c } m = n.$$

$\Rightarrow$ Level 2 uses $O(n)$ space.

Open Address Hash Tables

Linear Probing

To insert x, try slot $h(x)$

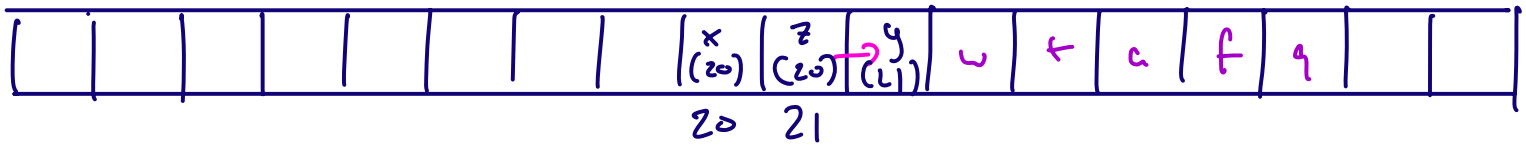
if full try slot $h(x)+1$

" " " " $h(x)+2$

⋮

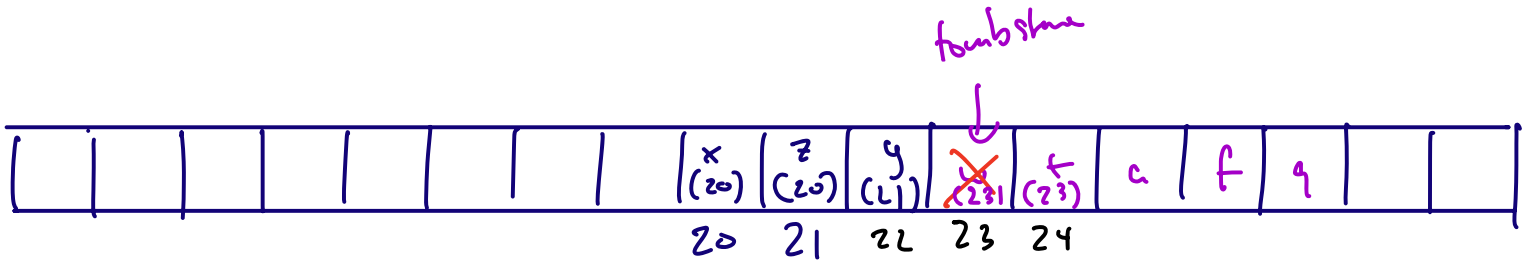
Ordered Linear Probing

Insert z w/ $h(z) = 20$



Query for b w/ $h(b) = 20$

Keeping hashes ordered with runs lowers query cost.



Delete w.

Can't delete from the middle of a run

Two techniques:

1. Rebuild the run without the hole (expensive)
2. Use a tombstone