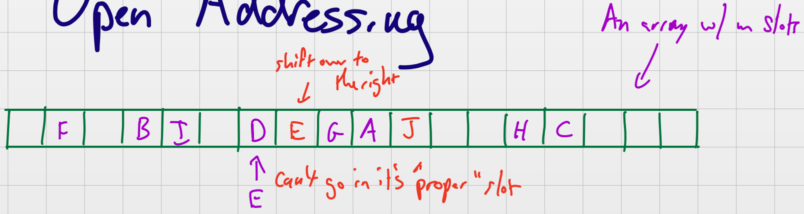
CS 5112 Hash Tables

Perfect Hashing

Open Addressing

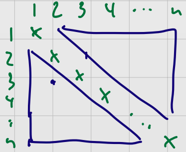


Insert (A) : Take $h(A) \% m$, and put A into that slot
Insert (B) :
Insert (C)
Insert (D)
Insert (E) : Suppose $h(E) \% m = h(D) \% m$

Can we have a hash table w/ no collisions?

We could hash into a really big array.
Suppose we have n items and m slots.

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



$$\frac{n^2-n}{2} = \frac{n(n-1)}{2}$$

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

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

Markov's Inequality

Suppose X is a R.V. w/ $X \geq 0$.

Then $\Pr(X \geq a) \leq E\{X\}/a$ for $a > 0$.

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

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

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

$\square$

Can we have a hash table w/ no collisions?

We could hash into a really big array.
Suppose we have n items and m slots.

$$E\{\#\text{ of colliding pairs}\} = \sum_{i<j} \Pr(h(x_i) = h(x_j))$$
$$= \sum_{i<j} \frac{1}{m} = \frac{n(n-1)}{2} \cdot \frac{1}{m} = \frac{n(n-1)}{2m}$$

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

$E\{\#\text{ colliding pairs}\} \leq \frac{1}{2}$.

$\Pr(\#\text{ of colliding pairs} \geq 1) \leq \frac{1}{2}$ by Markov's.

1	$\frac{1}{2}$
2	$\frac{1}{4}$
3	$\frac{1}{8}$

$$\text{Expectation} = \frac{1}{2} + 2 \cdot \frac{1}{4} + 3\,\frac{1}{8} + \cdots$$
$$= \sum \frac{i}{2^i}$$

Static Hash Tables.

Two operations: Construct (array of n items)
Query (key)

Really big hash table:

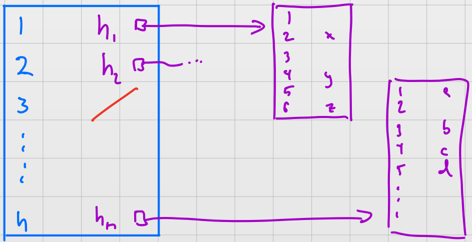
Expected 2 times.

Construct: 1. Pick a hash function $O(1)$
2. Try to put all items into the array $O(n)$
3. If 2 fails, retry 1.

Query: 1. Look at $h(k)$.

FKS Perfect Hashing

Level 1:



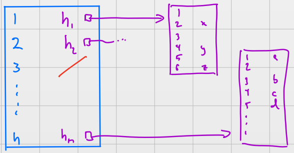
Level 2: Each bucket is a really big hash table (i.e. collision free)

Let C_t be the # of items in bucket t.
The level 2 table for t has C_t^2 slots.

FKS Perfect Hashing

Level 1:

Level 2: Each bucket is a really big hash table (i.e. collision free)



Let C_t be the # of items in bucket t.
The level 2 table for t has C_t^2 slots.

$O(n)$

Construct: 1. Loop through all items and hash them to buckets (lists)

2. For each bucket: a. Choose a random h_t. $O(1)$

b. Try to assign items to level 2 slots using h_t.

c. If there's a collision, back to a.

$O(C_t)$

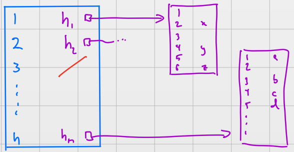
Expected $O(1)$

actually 2

FKS Perfect Hashing

Level 1:

Level 2: Each bucket is a really big hash table (i.e. collision free)



Let c_t be the # of items in bucket t.

The level 2 table for t has c_t^2 slots.

expected

$$O(n) + O(1) \cdot \sum_{t=1}^{n} c_t$$

$$= O(n) + O(n)$$

in expectation

$O(n)$

Construct: 1. Loop through all items and hash them to buckets (lists)

Expected $O(1)$ actually 2

2. For each bucket: a. Choose a random h_t. $O(1)$

$O(c_t)$

b. Try to assign items to level 2 slots using h_t.

c. If there's a collision, go back to a.

FKS Perfect Hashing

What about space?

Level 1: Uses $O(n)$

Level 2: $\sum_{t=1}^{n} C_t^2$ space

$E[C_t^2]$