Universal Hash functions

Theorem: For key k, $E[n_{h(k)}] \leq \{\alpha + 1 if k \in T, \alpha otherwise\}$

n= total # items in T m = size of T n/m = load factor = α n_i = # of items in table T bucket i.

Proof: Let $X_{kl} = \{1 if h(k) = h(l), 0 otherwise\}$ Let $Y_k =$ number of keys that hash to same slot as k.

$Y_k = \sum_{l \neq k, l \in T} X_{kl}$

$E[Y_k] = E[\sum_{l \neq k, l \in T} X_{kl}] = \sum E[X_{kl}] = \sum 1/m$

If $k \notin T$ then $n_{h(k)} = Y_k$and $|\{k \in T | l \neq k\}| = n \rightarrow E[Y - k] = n/m$

If $k \in T$ then $n_{h(k)} = Y_k + 1$ and $|\{k \in T | l \neq k\}| = n - 1 \rightarrow E[Y - k] = (n-1)/m$.

Example of universal set of hash functions

$h_{a,b}(k) = ((ak + b) \mod p) \mod m$

p is a prime bigger than any key (fixed)

Choose $a \in \{1, 2, \ldots, p-1\}$ and $b \in \{0, 1, \ldots, p-1\}$ to select hash function from set.

Cuckoo Hashing

Find operation takes O(1).

Delete operations takes O(1).

Insert operation takes O(1) amortized, expected

This is very weird, seems almost impossible. Occasionally, the table resizes.

Use two hash functions h_1 and h_2.

A key k with either be stored in $h_1(k)$ or in $h_2(k)$. Each slot in table contains one key.

If the slot is full, it tries the backup hashing function, if that also fails, it kicks that node out. If it cycles, it resizes to be larger.

Cuckoo graph for n items

Vertices = {0,1,…,m-1}

Edges $= \{(h_1(k), h_2(k)) | k \in T\}$

Assume $h_1(k)$ and $h_2(k)$ are uniform and independent random edges.

We add n random edges to the graph. We want to know if there's a cycle. Insertion will succeed if there's no cycle sn the cuckoo graph.

First consider paths

Lemma: For c > 1, and $m \geq 2cn$.

Pr[cuckoo graph has path of length l from i to j] $leq \frac{1}{mc^l}$.