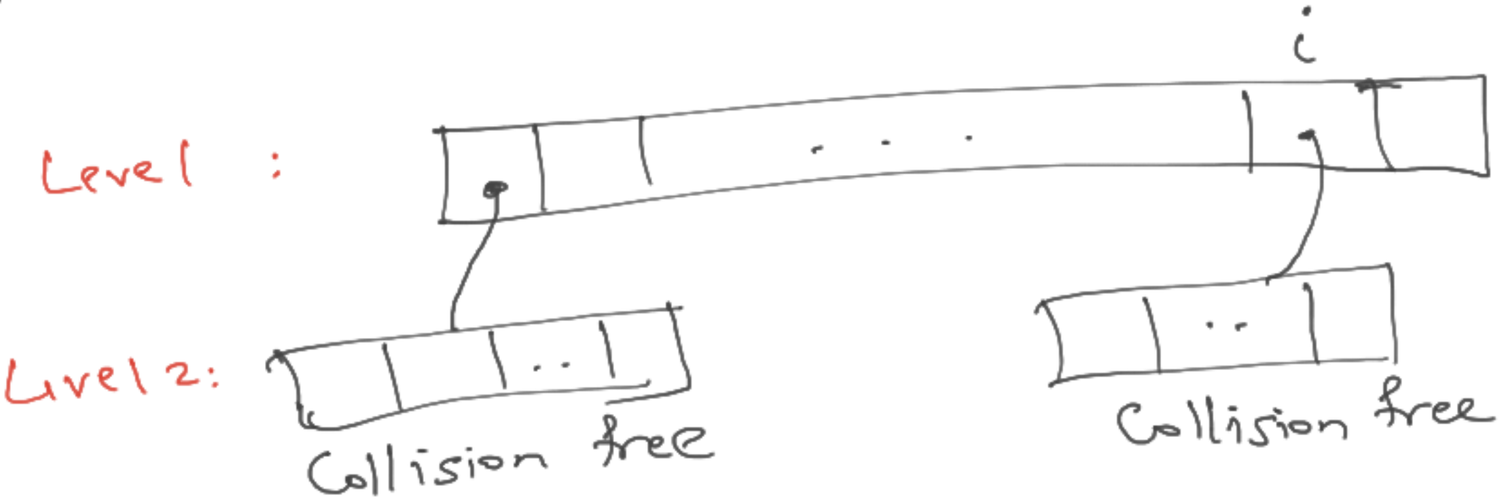
Hashing 2

Jan 27 2022

Two-level hashing ("perfect hashing")



Let $C(i)$ = num. of elements that gets mapped to location i (in the first level)

If Level 2 HT is $C(i)^2$ → Collision free for location i

Total table space for Level 2

$$= \sum_{i=1}^{M} c(i)^2$$

we know $E[C] = \binom{N}{2}\frac{1}{M}$

$$E\left[\sum_{i=1}^{M}\binom{c(i)}{2}\right] = \binom{N}{2}\frac{1}{M}$$

$$E\left[\sum c(i)^2 - \sum c(i)\right] = O(N) \quad \text{(since } M = O(N)\text{)}$$

$$\Rightarrow E\left[\sum c(i)^2\right] = O(N)$$

Collision-free and $O(N)$ table space!

More stronger properties:

k-wise independent hash functions

Defn: $H \quad U \longrightarrow [M]$ k-wise indep. if for any k distinct keys $x_1, \ldots, x_k$ and values $\alpha_1, \alpha_2 \ldots \alpha_k$

$$p\left(h(x_1)=\alpha_1 \cap h(x_2)=\alpha_2 \cap \ldots h(x_k)=\alpha_k\right) \leq \frac{1}{M^k}$$

$k=2$, pair-wise indep.

Properties: H is k-wise indep for $k \geq 2$. Then

1. H is also $(k-1)$-wise indep.
2. For any $x \in U$ and $a \in [M]$, $P[h(x) = a] \leq \frac{1}{M}$
3. H is universal

Pairwise indep (stronger!) vs. universal ?

$h(x) = Ax$

Consider $x = 0$

$h(0) = 0$

To turn this into pairwise indep : $h(x) = Ax + b$ (uniform random binary vec.)

Open addressing

- Single array
- No separate D.S.

- Linear probing
  - use step-size

Quadratic

Cuckoo Hashing

Two tables T_1 & T_2

both of size $M = O(N)$

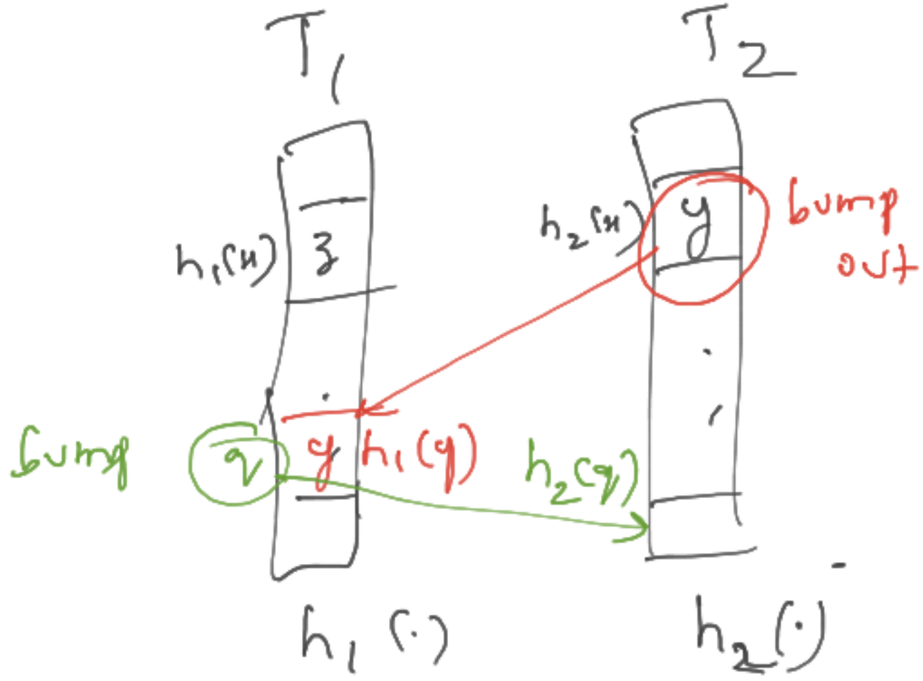
Two hash functions h_1 & $h_2 \in H$

assume fully random ($O(\log N)$-wise indep suffices)

Insertion:

- x goes into either $T_1[h_1(x)]$ or $T_2[h_2(x)]$

- stop when no more bumps or if more than $6 * \log N$ bumps & rehash



Query:

$O(1)$ only 2 locations.

Thm: The expected time to perform insert is $O(1)$ if $M \geq 4N$

Proof Sketch:

"Cuckoo graph" G.

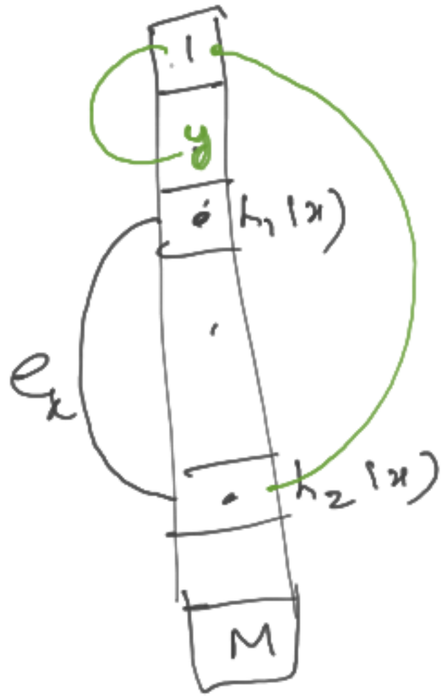
M vertices:
= hash table locations.

Edges: correspond to the items to be inserted.

$\forall x \in S$, $e_x = (h_1(x), h_2(x))$

↑ for all

Cuckoo graph



"y" can get bumped out when inserting a new element x if y falls in a path in the cuckoo graph starting from $h_1(x)$ or $h_2(x)$

Defn: Bucket of x

$B(x)$ = set of nodes of G reachable from $h_1(x)$ or $h_2(x)$

Random

= 'connected component' of G with edge e_x

$E[\text{insertion time of } x] = E[|B(x)|]$

size of $B(x)$

To show: $E[|B(x)|] \leq O(1)$

$$E[|B(x)|] = \sum_{\substack{y \in S \\ y \neq x}} P[e_y \in B(x)]$$

$$\leq N \, P[e_y \in B(x)]$$

union bound

Sufficient to show

$$P[e_y \in B(x)] \leq O\left(\frac{1}{M}\right)$$

$M \geq 4N$.

Lemma: For any $i, j \in [M]$

$$P[\text{there exists a path of length } \ell \text{ between } i \text{ and } j \text{ in the cuckoo graph}] \leq \frac{1}{2^\ell M}$$

Proof: (exercise)

For $\ell = 1$ . . .

Then use induction.

To show : $P[e_y \in B(x)] \leq \Theta\left(\frac{1}{M}\right)$

Rmk : Using the lemma

$$P(e_y \in B(x)] \leq \sum_{l \geq 1} \frac{1}{2^l M}$$

(from prev Lemma)

$$= O\left(\frac{1}{M}\right)$$

$M \geq 4N$

Application : Bloom Filter

Membership query

Room for mistakes :

Only false positives

but no false negatives

Useful for 'filter operations' - typically elements not in the set

Space efficient data structure for approximate membership queries

- Array T of M bits
- k hash function $h_1, h_2, \ldots h_k : U \rightarrow [M]$
  (Assume completely random)

Adding a key:

$x \in S$ set bits $T[h_1(x)], T[h_2(x)] \ldots T[h_k(x)]$ to 1

Membership query: check locations

$x : T[h_1(x)] \ldots T[h_k(x)]$

Let p = prob. that a bit in T is not set

$$p = \left(1 - \frac{1}{M}\right)^{kN} = \left(\left(1 - \frac{1}{M}\right)^{M}\right)^{\frac{kN}{M}}$$

$$\left(1 - \frac{1}{x}\right)^{x} \to \frac{1}{e}$$

$$= \left(e^{-1}\right)^{\frac{kN}{M}}$$

$$= e^{-\frac{kN}{M}}$$

Prob. of false positive = all k bits set

$$= (1-p)^{k} = \left(1 - e^{-\frac{kN}{M}}\right)^{k}$$

$$\frac{d}{dk}(\quad) = 0 \qquad \text{(Exercise)}$$

False positive prob minimized

$$k = \frac{M}{N} \ln 2$$

ϵ denotes prob. of false positive.

$$\epsilon = \left(\frac{1}{2}\right)^{\frac{M}{N} \ln 2}$$

$$M = 1.44 \log(1/\epsilon)$$

For 1% false positive.

$M \approx 10N$ bits

$k = 7$

$N * \log|U|$