CSL 630 Lecture 19 Sept 29

Universal Hash function

A set H of hash functions that satisfy the following property for all pairs $x, y \in \mathcal{U}$ (universe)

$$\sum_{h \in H} \delta_h(x,y) \leq \frac{c|H|}{m}$$ for some constant c

$$\delta_h(x,y) = \begin{cases} 1 & \text{if } h(x) = h(y) \\ 0 & \text{otherwise} \end{cases}$$ $x \neq y$

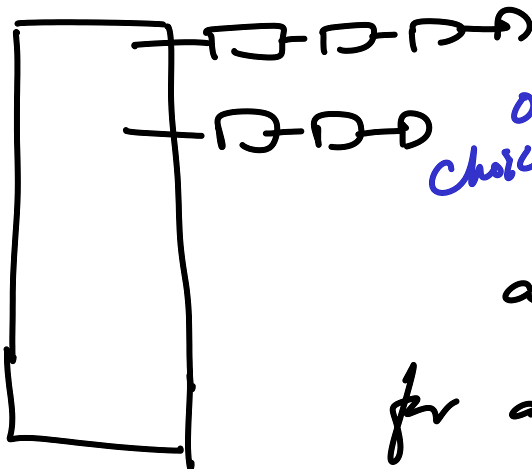
collision function

m = size of hash Table locations $\{0, 1, 2, \ldots m-1\}$

$\boxed{\text{What is the probability that } x, y \text{ collide for a randomly chosen } h?}$

$\leq \frac{c}{m}$

For any arbitrary subset $S \subset U$
$|S| = n$, and we choose a random hash function from H, what is the "expected" performance?



over choice of h

Suppose the <u>expected</u> length of a chain is ℓ. Then for a sequence of operations on S involving $\{$search, insert, del$\}$ the expected time for any operation is ℓ $\Rightarrow$ for T operations, the expected time is $T \cdot \ell$ (linearity of Expectation)

Given $x \in S$, we want to bound the expected no. of elements $y \in S$ that collide with x.

E [No. of elements $y \in S$ that collide with x] =

$$\underset{\text{choice of } h}{E}\left[\sum_{y \in S} \delta_h(x,y)\right]$$

$$\frac{1}{|H|}\left[\sum_{h \in H} \sum_{y \in S} \delta_h(x,y)\right]$$

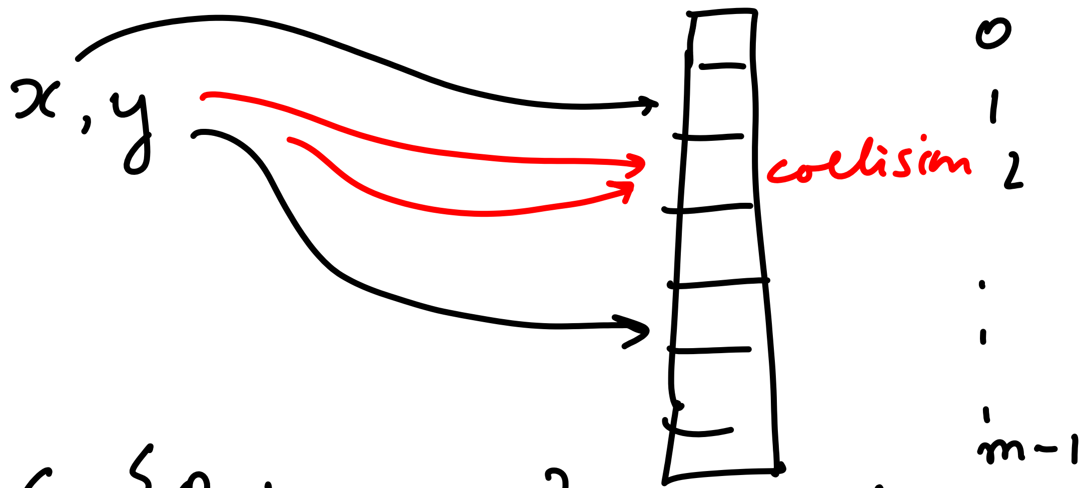
$$= \frac{1}{|H|} \sum_{y \in S} \sum_{h \in H} \delta_h(x,y)$$

$$= \sum_{y \in S} \underbrace{\frac{1}{|H|} \sum_{h \in H} \delta_h(x,y)}_{\frac{c}{m}\ \text{from the defn of Univ hash fn}} \leq \sum_{y \in S} \frac{c}{m}$$

$$\leq n \cdot \frac{c}{m} \qquad \frac{n}{m} : \text{loading factor}$$

$$= O(1) \text{ for } n = m$$

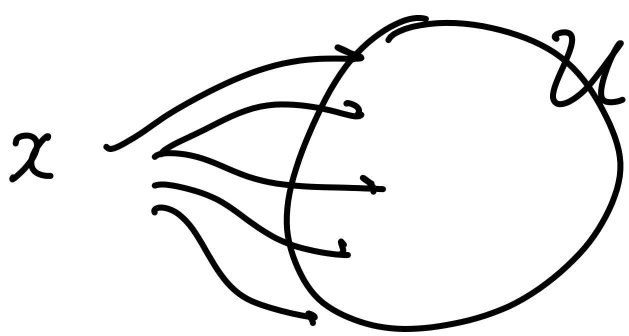
Existence of Universal Hash functions?



$x, y \in \{0, 1, \ldots N-1\}$ $\quad |\mathcal{U}| = N$

$\longrightarrow \{0, \ldots m-1\}$

If x can be mapped to the location $\{0 \ldots m-1\}$ randomly, then the property of universal function can be satisfied



Add a random no. a to x

$a \in \{0, 1, \ldots N-1\}$

$$x \longrightarrow (x+a) \bmod N \bmod m$$

$$\longrightarrow \{0, 1, \ldots m-1\}$$

$$h_a(x) = (x+a) \bmod N \bmod m$$

$$|H| = N$$

$\forall\, x, y \qquad \sum_{h \in H} \delta_h(x,y) \overset{?}{\leq} \frac{c\,|H|}{m} = \frac{cN}{m}$

for some constant c

For how many a's $\delta_a(x) = \delta_a(y)$

$$\big[\underbrace{(x+a) \bmod N}_{x'}\big] \bmod m \equiv \big[\underbrace{(y+a) \bmod N}_{y'}\big] \bmod m$$

$$x' \equiv_m y' \Rightarrow x' - y' \equiv_m 0$$

$x', y' \in \{0, 1, \ldots N-1\}$

$$x' - y' \in \left\{0, \pm m, \pm 2m, \ldots \pm \left\lfloor \frac{N}{m} \right\rfloor m\right\}$$

$x' - y' = km \Rightarrow x + a \equiv_N y + a + km$

$\Rightarrow x \equiv_N y + km$ so x and y will collide for all choices of a, i.e. it is not universal if $(x - y = km) \bmod N$

Let us choose instead $h_a(x) = (a \cdot x \bmod N) \bmod m$

for $a \neq 0$, N is prime

Then $x' \equiv_m y' \Rightarrow x \cdot a \equiv ya + km \bmod N$

$\Rightarrow (x-y)\,a \equiv_N km \Rightarrow a = (x-y)^{-1} \cdot km$

For each k, there is a unique solution a since $x-y \neq 0$ and $(x-y)^{-1}$ exists (since N is chosen prime)

So for $2 \cdot \lceil \frac{N}{m} \rceil$ choices of k, there are $\leq 2 \lceil \frac{N}{m} \rceil$ choices of a

i.e. x and y collide for at most $2\frac{N}{m}$ hash functions out of $N-1$ possible function $(a \neq 0)$. So

Since $\frac{2N}{m} \leq 2\left(1+\frac{1}{N-1}\right) \cdot \frac{N-1}{m}$

it is $2\left(1+\frac{1}{N-1}\right)$ universal. $\sim$ 2 universal since N is very large

Fact: (Bertrand's postulate) There is at least one prime between k and $2k$ for any integer k. So N can be chosen

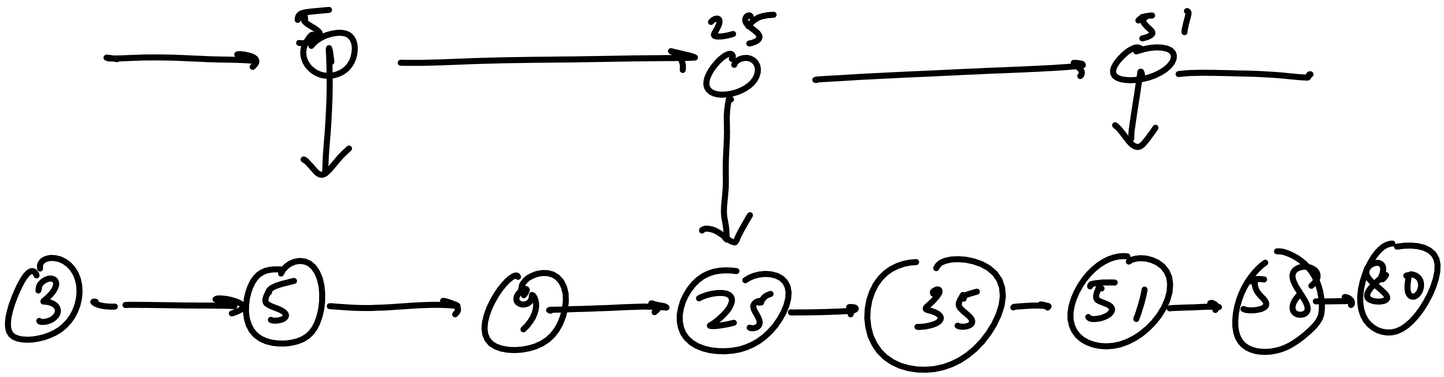
For any subset S, the expected time for T operations is $\leq cT$ (from universal hash function)

$\Rightarrow$ The prob that time will exceed $2 \cdot c \cdot T \leq \frac{1}{2}$

(Markov's ineq)

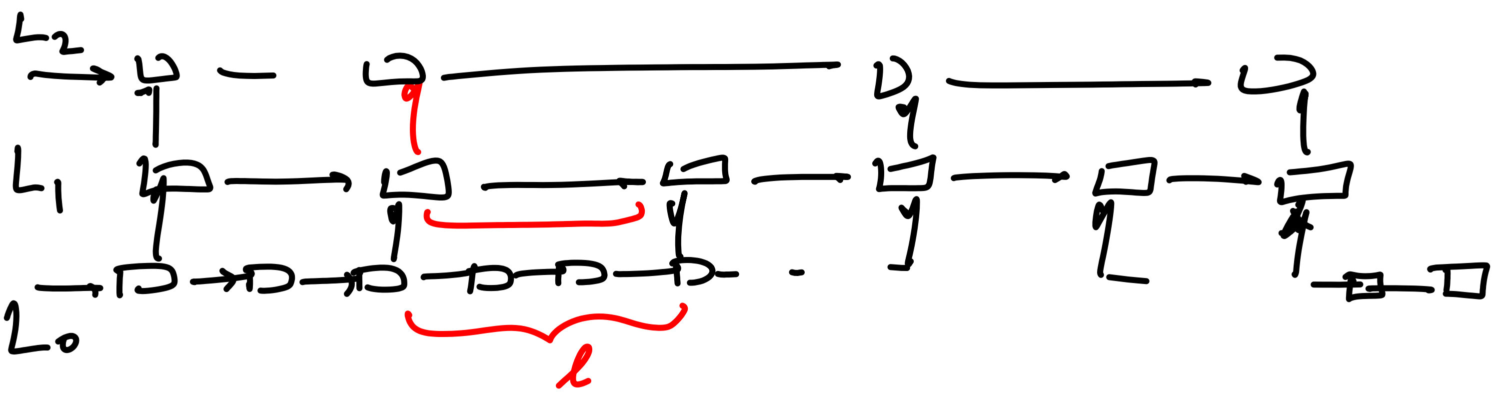
$\Rightarrow$ At least half the functions will behave well for S.

Skip Lists : an alternative dynamic dictionary data str.



20 ?

Insert / Search / Delete would take similar time in an ordered list



$L_0 \supset L_1 \supset L_2 \ldots L_i \supset L_{i+1} \ldots \supset L_k$

$|L_k|$ is relative small, we can do normal linked list search

Time to search $= |L_k| + \sum_i l_i$

$\leq O(k)$ if $|l_i|$ is $O(1)$