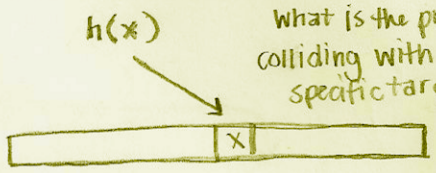
CS157 Lecture #11 : HASHING

HASH FUNCTIONS vs CRYPTOGRAPHIC HASH FUNCTIONS

* Where are hash functions useful?
  - ↳ hash tables [data structure w/ O(1) lookup]



What is the prob of colliding with a specific target?

$m \approx 10^9$ [to fit in memory]

no collisions in one trial : $1 - \frac{1}{10^9}$

collision in 10^9 trials : $1 - \left(1 - \frac{1}{10^9}\right)^{10^9}$

$\approx 1 - \left(e^{-\frac{1}{10^9}}\right)^{10^9}$

$\approx 1 - e^{-1}$

* how big should we make sure m is (being as lazy as possible)?
  - ↳ attacks, in time O(m) have probability $1 - \frac{1}{e}$ of succeeding, so want something that takes at least 10^{15} computations (at least 64 bits)

BIRTHDAY PARADOX

Given K people, what is the probability that 2 of them will have the same birthday?

$\sum_{i=1}^{K} \frac{(i-1)}{365} = \binom{K}{2}\frac{1}{365}$ → $\binom{K}{2} \cdot \frac{1}{m} \Rightarrow$ collision likely when $K \geq \sqrt{m}$

← what is the probability that any collision will occur?

* are hash functions (esp cryptographic) actually random? (no)

Ex Let's hash IP addresses: (x_1, x_2, x_3, x_4) s.t. $0 \leq x_i \leq 2^5$.

① pick coefficients c_1, c_2, c_3, c_4 at random, $0 \leq c_i \leq m-1$

② $h(x) = \sum_{i=1}^{4} x_i c_i \mod m$

To analyze, first analyze $x_1 c_1 \mod m$

↳ take $ax \equiv b \mod c$. Then there are either 0 solutions, ↳ or infinitely many. What if we only want solutions mod c?

↳ if $\gcd(a,c) = 1 \Rightarrow$ 1 solution

↳ Suppose $x_1 c_1 \mod m = b$. For how many choices of c_1 will this be true? $\gcd(x_1, m)$

↳ but if we take m to be prime, there is exactly 1 choice of coefficient for every target

* Let m = 257. A hash function family h is universal if $\forall x \neq y$ $P(h(x) = h(y)) \leq \frac{1}{m}$ (over all parameters). Is ours universal?

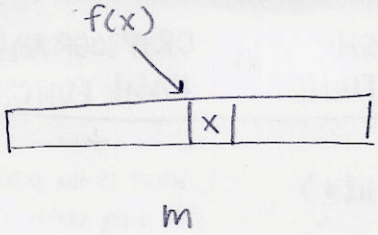
$P\left(\sum x_i c_i \bmod m = \sum y_i c_i \bmod m\right) = P\left((x_1 - y_1)c_1 + \sum_{i=2}^{4}(x_i - y_i)c_i \equiv 0\right)$

$= P\left((x_1 - y_1)c_1 \equiv -\sum (x_i - y_i)c_i\right)$ ← Pick c_2, c_3, c_4. Then the right hand side becomes a constant

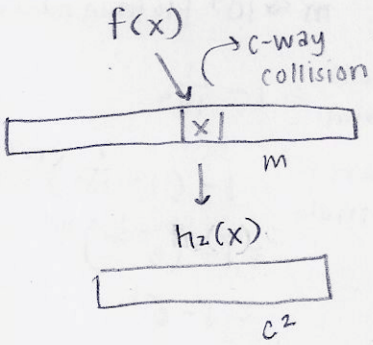
$= P\left((x_1 - y_1)c_1 \equiv K\right) = \boxed{\frac{1}{m}}$

↳ since $x_1 - y_1 \neq 0$, this has one solution.

$x \neq y \Rightarrow x_1 \neq y_1$ (WLOG)



* What do you do in the case of a collision?
  - ↳ linked list at each entry of the array
  - ↳ hash it again with a different function



Let c_i = # things hashed to i

$\Rightarrow$ storage $= m + \sum_i c_i^2 = m + \sum_i c_i + 2\binom{c_i}{2}$

$= m + n + \sum_i 2\binom{c_i}{2} \leq m + n + \frac{n^2}{m} \approx O(m)$ [when $n \approx m$]