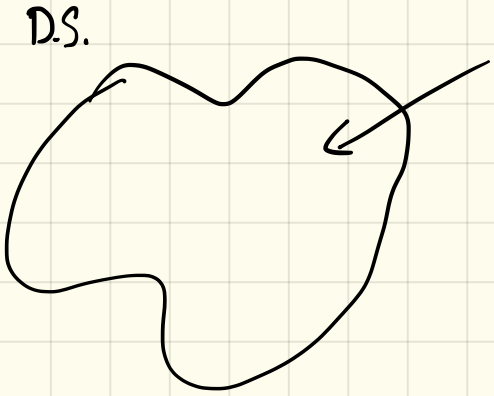
20.2.17:

Topics:
- Dictionaries / Hashing
- L(A, RMQ, ...)
- Suffix Trees/Arrays
- Persistence
- D.S. - geometric
  - for integers

Exam: at the end

Lecture notes: none

Dictionaries & Hashing



Binary Search Tree: $O(\log n)$ (slow)

Target: $O(1)$

As a tool: Hashing $h: [U] \to [m]$

ideally : h is totally random (family of fnc.)

Problem: size $U \log n$

$\Longrightarrow$ family $\mathcal{H}$ of hash function of small size which is at least universal, i.e.,

$$\forall x \neq y \quad \underset{h \in \mathcal{H}}{P}[h(x) = h(y)] = \underbrace{O(1/m)}_{\text{some times } \leq 1/m}$$

E.g.) $h_a(x) = ((ax) \bmod p) \bmod m$

where $p > m$ prime, $0 < a < p$

is universal.

Problem For $x_1, x_2 \in [U]$ $h(x_1), h(x_2)$ are may be correlated.

Def) A hash function $\mathcal{H}$ is

k-independent if

$$\forall_{x_1,\dots,x_k} \; P\Big[\bigwedge_i h(x_i) = t_i\Big] = O(1/m^k)$$

$\uparrow$ $x_i \neq x_j$ for $i \neq j$

E.g.) $((ax+b) \bmod p) \bmod m$ 2-ind.

E.g.) $\left(\left(\sum_{i=0}^{k-1} a_i x^i\right) \bmod p\right) \bmod m$ k-ind.

cf. Shamir's secret sharing

E.g.) Tabulation hashing (TH)

$u = 2^{c \cdot \ell}$ $\quad m = 2^{\ell'}$

$x = x_1 x_2 \cdots x_c \quad 0 \le x_i \le 2^{\ell}$

$\underbrace{T_1, T_2, \ldots, T_c}_{\text{totally random hash functions}} : [2^{\ell}] \to [2^{\ell'}]$

$h(x) := T_1(x_1) \oplus \cdots \oplus T_c(x_c)$

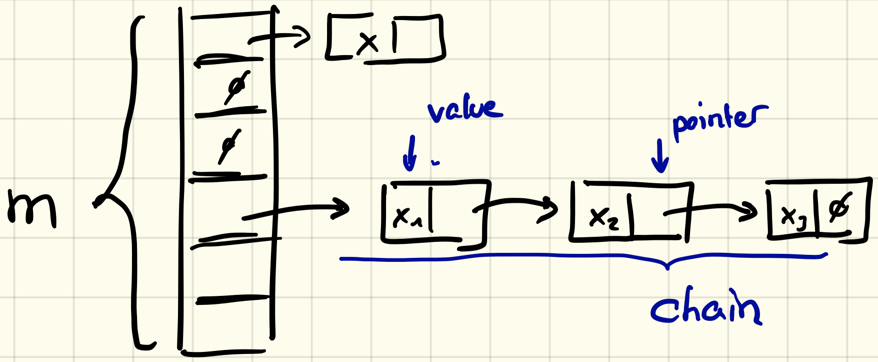
Storage space:

$$\mathcal{O}(c \cdot \underbrace{\ell'}_{\log(2^{\ell'})} \cdot 2^{\ell}) = \mathcal{O}(c \cdot \log m \cdot u^{1/c})$$

Lemma: TH is 3-independent, but not 4-independent ($c > 1$)

Chaining

hash fnc. $h: [u] \to [m]$



n elements, m table size

C_t length of the t-th chain.

$$E[C_t] = \sum_i P[h(x_i) = t]$$

universal h: $E[C_t] = O(n/m) \overset{\text{if } n = O(m)}{=} O(1)$

$\leadsto$ cost is quadratic in C_t.

$\to$ interested in $E[C_t^2]$.

$$E[C_t^2] = 1/m \sum_{s \in [m]} E[C_t^2]$$

$$= 1/m \sum_{i \neq j} P[h(x_i) = h(x_j)]$$

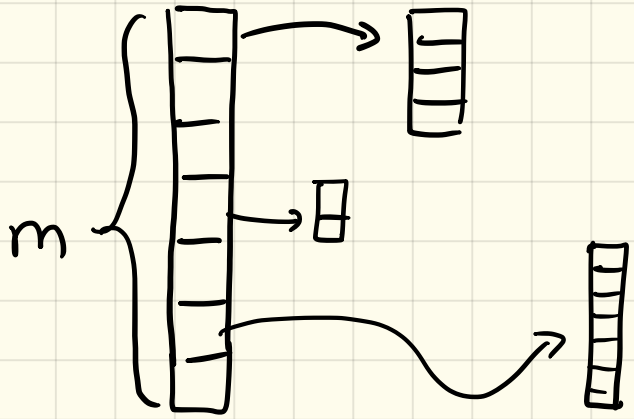
if at least 2-wise ind. ↓

$$= O(n^2/m^2) =^* O(1) \quad * \; n \in O(m)$$

w.h.p. all $C_t = O(\log n / \log \log n)$ | h. totally random Chernoff bounds.

also holds for $O(\log n / \log \log n)$ - ind. h and for tabulation hashing.

FKS hashing (perfect hashing)



chain of length C_t
$\Rightarrow$ hash table of size $O(C_t^2)$

$n = C_t$, $m = C_t^2$

adjust size acc.

$$E[\text{collisions in } C_t] = O(C_t^2/C_t^2) = O(1) \leq 1/2$$

$$P[\text{no collisions in } C_t] \geq 1/2$$

$$E\left[\sum_t C_t^2\right] = O(n)$$

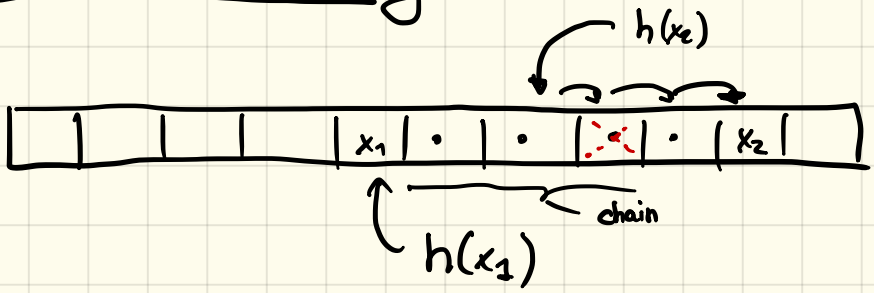
Complexity
amortized: $O(1)$
space: $O(n)$ expected

dynamic FKS

- rebuild 'bucket table' if there is a collision.
- if n gets too large double m. and rebuild all.
- if bucket gets too filled double its size and rebuild it.

Linear Probing

$m \geq (1+\varepsilon)n$



Insert, Query ✓

Delete: Only mark as deleted

↝ otherwise Insert, Query fail...

very fast (in proxis) if chains are small

totally random h: $O(1/\varepsilon^2)$

log n - ind. h: $O(1)$

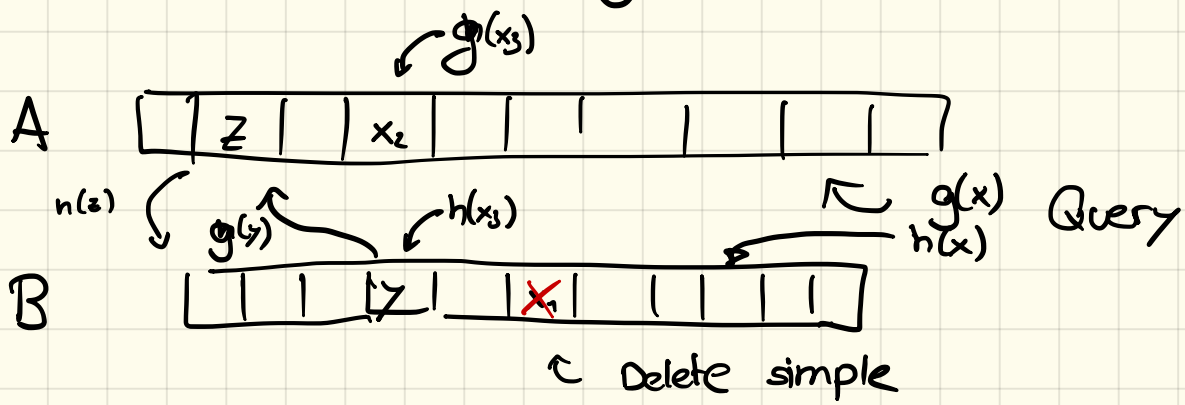
5 - wise h: $O(1)$

4 - wise h: it might not be $O(1)$

Tabulation hashing: $O(1/\varepsilon^2)$

Cuckoo hashing

$m \geq (1+\varepsilon)n$



$g: U \to A$, $h: U \to B$

$P[\,|\text{Chain of } op| = k\,] = 1/2^k$

?