1

ADT : DICTIONARY

↳ KEYS

- STACKS / QUEUES

POP → MOST RECENT

LEAST RECENT

LIFO / FIFO

OPERATIONS:

INSERT (K, V)

KEY NAME

INFORMATIONS RELATED TO K

RETRIEVE (K) → NOT EXISTS (UNSUCCESS)

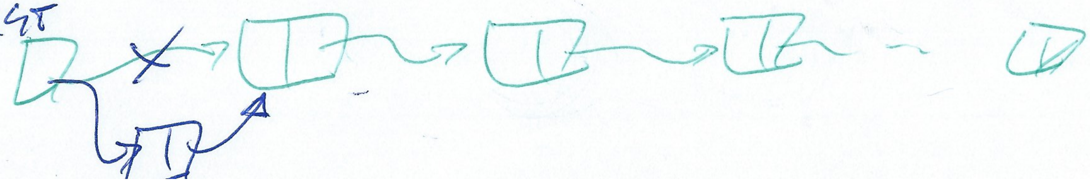
→ INFORM. REL. TO K (SUCCESS)

IMPLEMENT BY A LIST : INSERT AT FIRST TOP $O(1)$

RETRIEVE SCAN WHOLE LIST $O(n)$ → ESPECIALLY UNSUCCESS

OF ELEMENTS

LIST



2) MODEL OF COMPUTATION

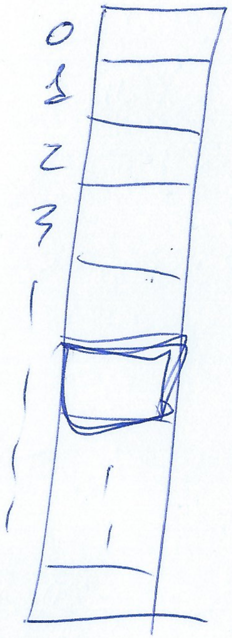
- POINTER MACHINE → ACCESS INFO VIA <u>POINTERS</u>

- RANDOM ACCESS MACHINE → ACCESS INFO VIA <u>ADDRESSES</u>

(RAM)

MEMORY

REAL COMPUTERS

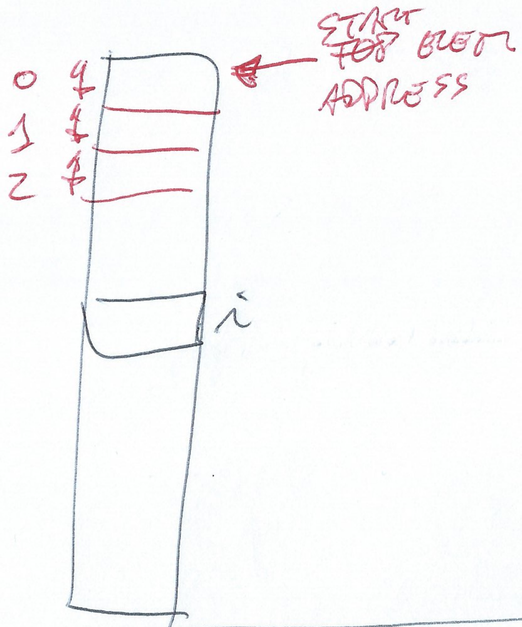


<u>CAN COMPUTE ADDRESSES</u>

$a_i + 27$

$\frac{a_i + a_j}{2}$

3) MONODIMENSIONAL ARRAY X



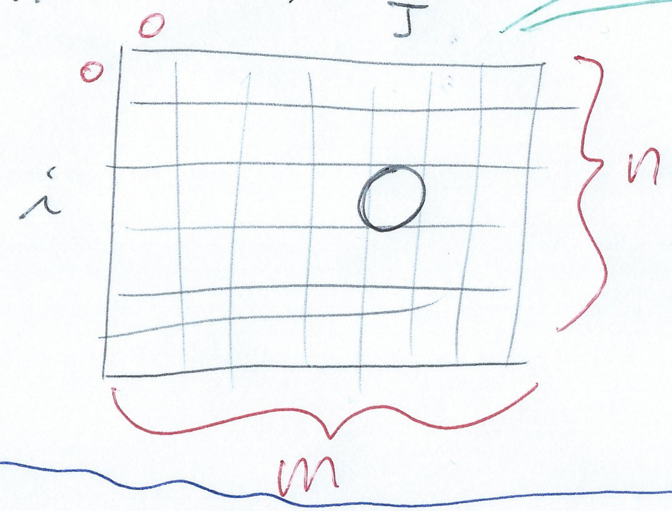
$X[i]$ COMPUTE ADDRESS

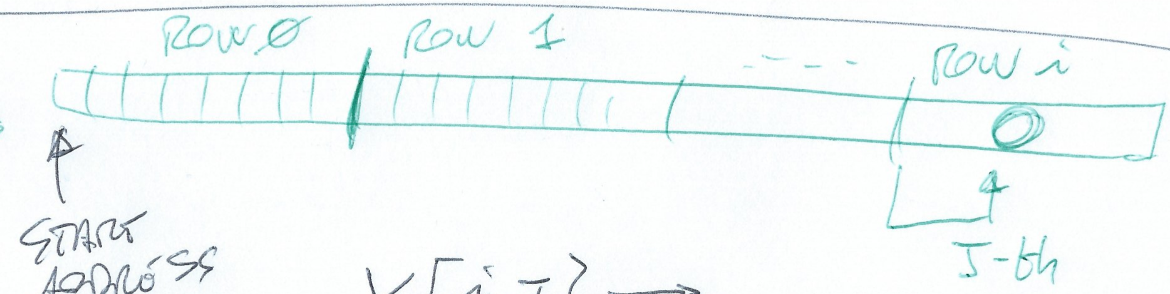
$= a_0 + i \times$ ITEM SIZE

START ADDRESS

MAP FUNCTION

BIDIM. ARRAY





$X[i,J] \longrightarrow$

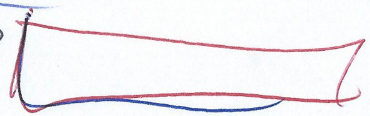
$ADDR_{i,J} = a_0 + \text{ITEM SIZE} \times (\underbrace{i \times m}_{\text{FULL ROWS}} + J)$

MAP FUNCTION

K-DIMENS. ARRAY

$n_0 \times n_1 \times n_2 \times \dots \times n_{k-1}$

$X[i_0, i_2, i_3, \dots, i_{k-1}] \longrightarrow$??



??

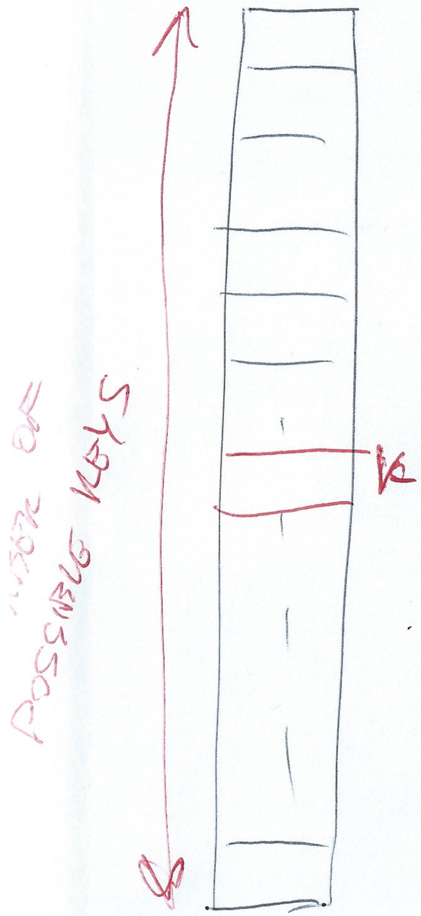
4

HASH TABLE

DIRECT ACCESS

	TIME	SPACE
RETRIEVE(K)	O(1)	∞ → SIZE OF UNIVERSE (KEY SPACE)
INSERT(K,V)	O(1)	
DELETE(K)		

K IS A NUMBER



DIRECT ~~HASH~~ ACCESS

FOR EACH DAY OF THE YEAR HOW MANY WORKED HOURS

INFO

KEYS

MATRICOLA → ANAGRAPHIC

$\approx 2 \cdot 10^8$ VALUES POSSIBLE

ONLY 10^7 PEOPLE

SURNAMES 25^{30} VALUES

THERE CAN BE WASTED SPACE

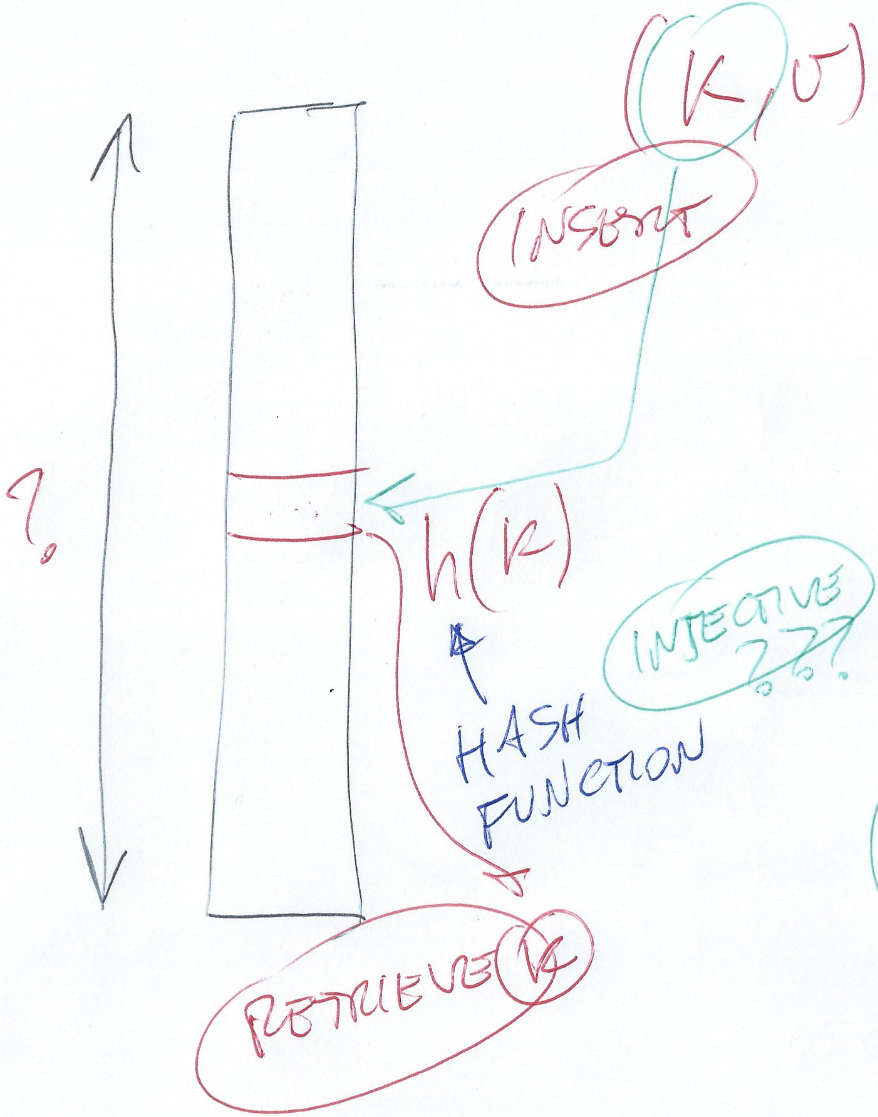
GOOD

5 HASH FUNCTIONS

n : NUMBER OF ITEMS (APPROX)

m : SIZE OF HASH TABLE $\quad m > n$

$m = n + 30\%$



$h : K \rightarrow \underbrace{[0, 1, \ldots\, m-1]}_{m \text{ addresses}}$

ANY POSSIBLE FUNCTION

MIX UP BITS

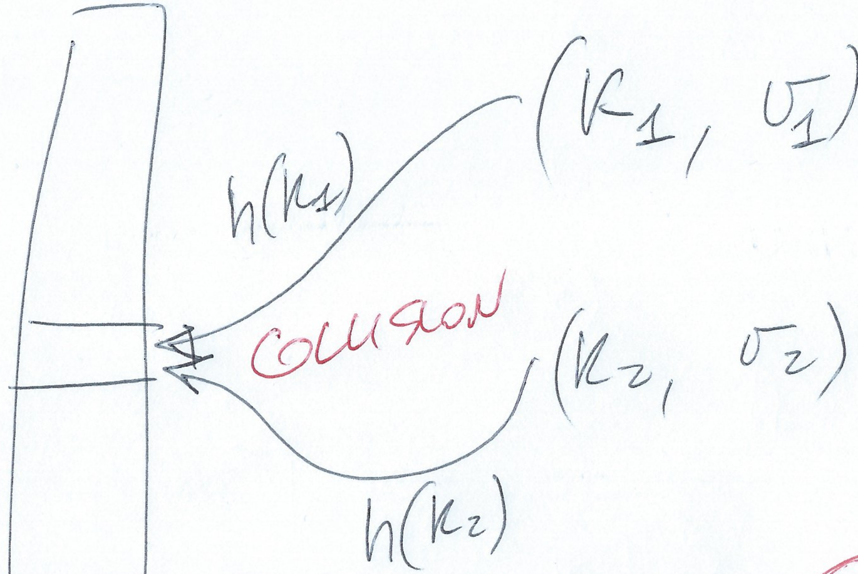
"RANDOM RESULT" ← DETERMINISTIC

K → NATURAL NUMBER % m

REMAINDER

$\in [0, 1, \ldots, m-1]$

6) COLLISIONS



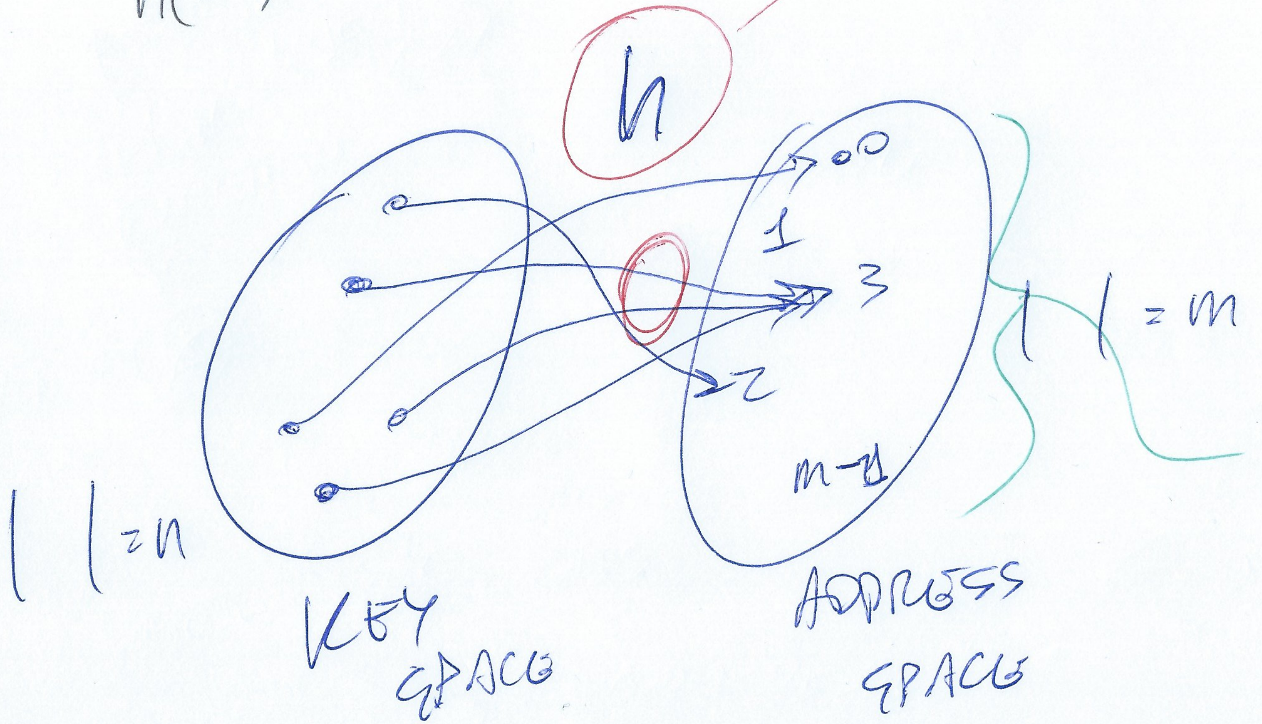
~~PERFECT HASHING (NO COLLISIONS) POSSIBLE FOR STATIC SETS OF KEYS~~

CRYPTOGRAPHIC FUNCTIONS

$\boxed{\text{MD5}}$

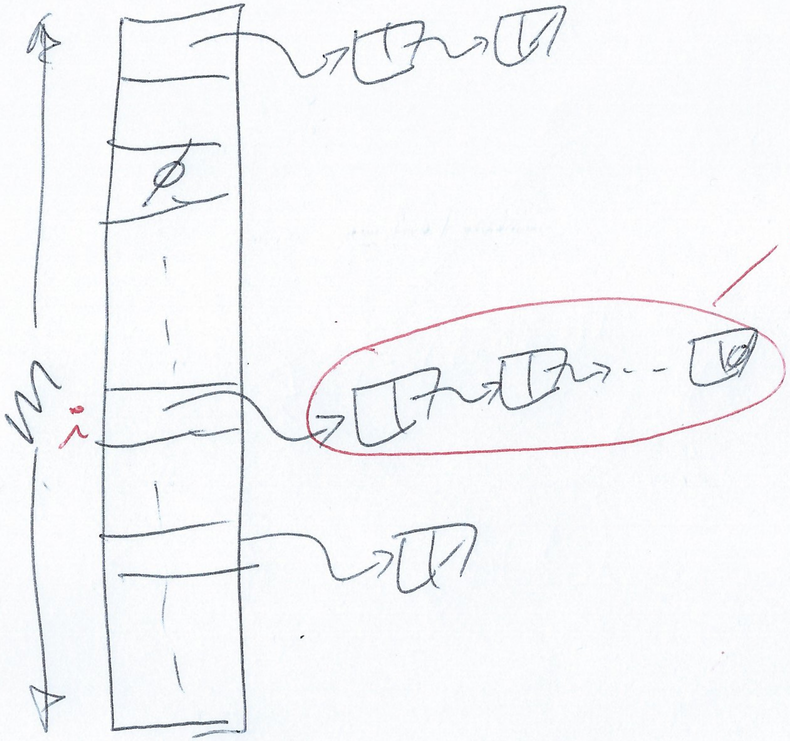
$\boxed{\text{SHA xxx}}$

- FINGERPRINTS
- MESSAGE DIGESTS



AS UNIFORM AS POSSIBLE

7 DEALING WITH COLLISIONS: CHAINING



CHAIN CONTAINS ALL KEYS (INSERTED) S.T. $h(k) = i$

- INSERT(k, v)
  - COMPUTE $h(k)$ — $O(1)$
  - ADD k, v TO LIST $h(k)$

  $O(1)$

- RETRIEVE(k)
  - COMPUTE $h(k)$ — $O(1)$
  - SCAN THE $h(k)$ LIST → v / $\nexists$

  $O(?)$

WORST CASE $\Theta(n)$

UNLIKELY

EXPECTED $\Theta(1)$ IF $m \approx n$

COMPUTE EXPECTED LENGTH OF CHAINS $= \dfrac{\sum \text{LENGTHS}}{m} = \dfrac{n}{m}$

- DELETE(k)

  AS RETRIEVE FOR

(8) COLLISIONS : OPEN ADDRESS (INTERNAL)

NO CHAINS: DATA IN
INSIDE THE TABLE

INSERT(k_1, v_1)

$h(k_1)$

BUSY?
TRY SOMEWHERE ELSE

PROBE
SEQUENCE

INSERT(k_2, v_2)

INSERT(k_3, v_3)

LINEAR:

$h(k), h(k)+1, h(k)+2, \ldots$

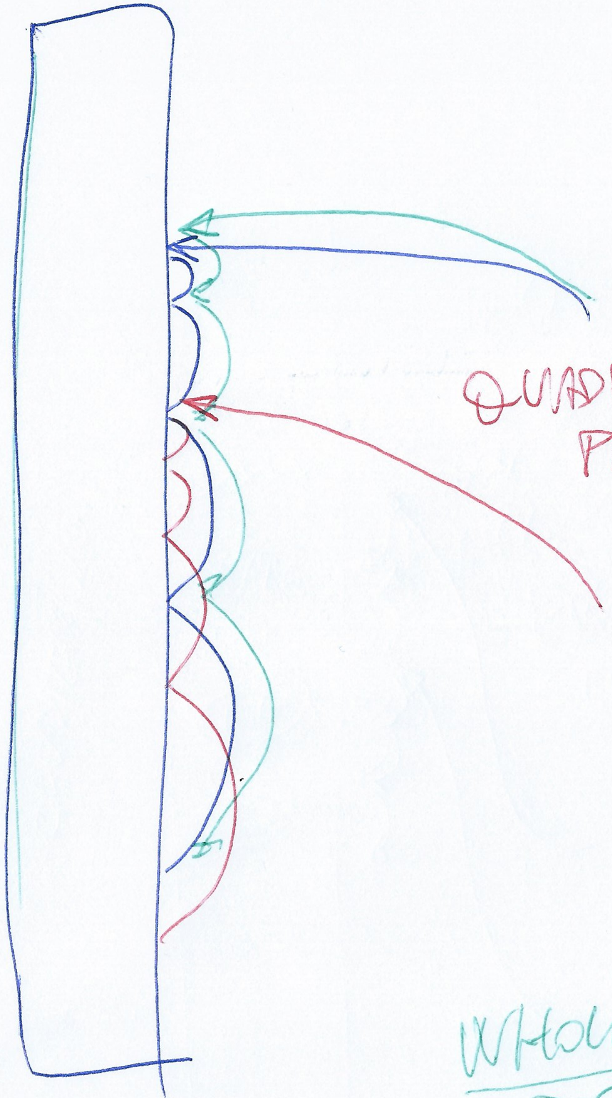
mod m

- PRIMARY CLUSTERING
  SAME PROBE SEQUENCE
- SECONDARY CLUSTERING
  SAME PORTION OF PROBE SEQUENCE

QUADRATIC:

$h(k), h(k)+1^2, \ldots, h(k)+i^2, \ldots$

(9)



QUADRATIC PROBING

NO SECONDARY CLUSTERING

DOUBLE HASHING

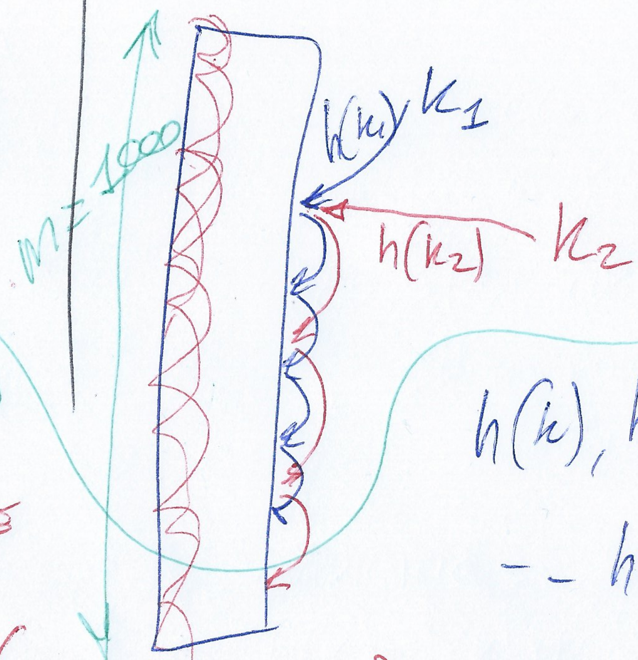
PROBE SEQUENCE:

$h(k_1),\ h(k_1)+h_2(k_1),\ \dots$

$\neq$ mod m

$h(k_2),\ h(k_2)+h_2(k_2),\ \dots$

CHOOSE PROBE STEP BY ANOTHER HASHING FUNCTION



$h(k),\ h(k)+h_2(k),\ \dots$

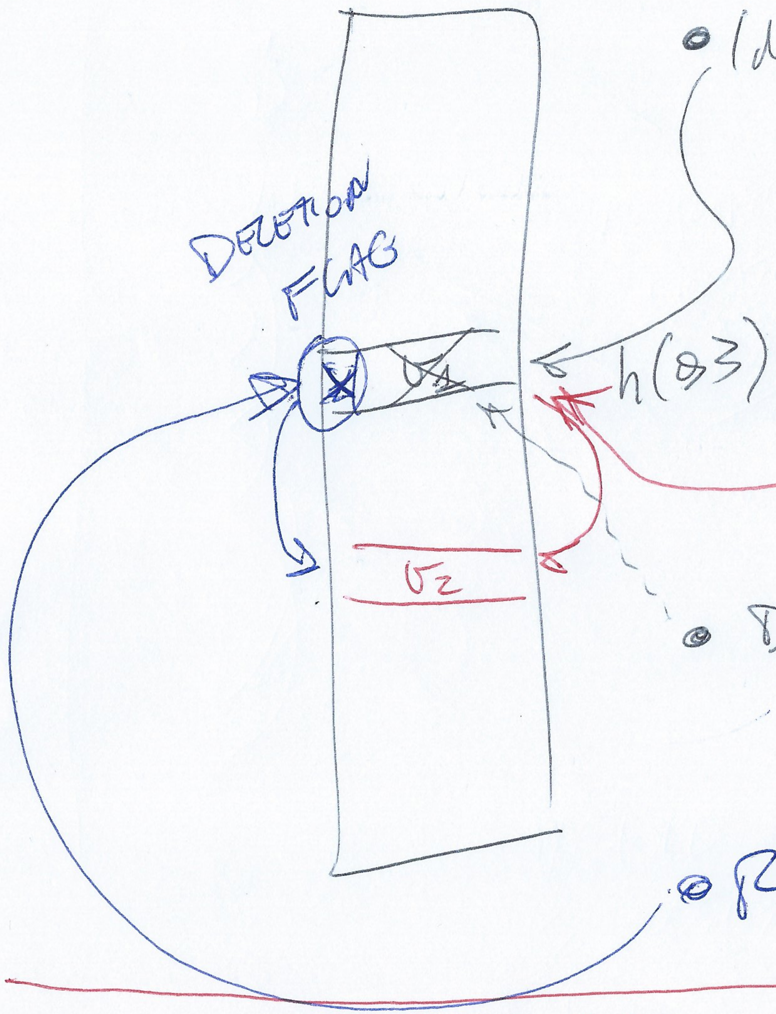
$\dots\ h(k)+i\,h_2(k)$

mod m

WHOLE TABLE IS POSSIBLY PROBED

TRUE IF m AND PROBE STEP HAVE MCD = 1 (RELATIVE PRIME)

DELETIONS (INTERNAL PROBING)



- INSERT(83, v_1)
- INSERT(321, v_2)

h(321)

- DELETE(83)
- RETRIEVE(321) → WRONG

$\Theta(1)$

VERY DYNAMIC SITUATION:

$m \neq n$ CHANGES

HOW TO CHANGE ?

INTERNAL PROBING

$$\text{RETRIEVE/INSERT} = \Theta\left(\text{EXPECTED LONGHT OF PROB}\right) \leq \frac{1}{1-\frac{n}{m}}$$