Count Min Algorithm : like a Bloom filter

$\{1, \ldots, n\}$

Stream $S = \{a_1, \ldots, a_m\}$, $a_j \in \{1, \ldots, n\} = [n]$

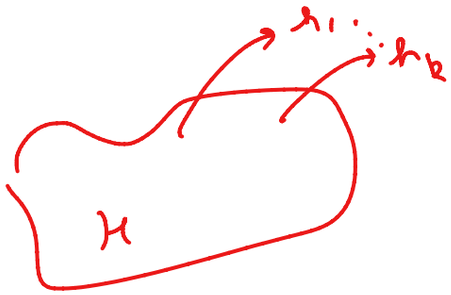
for $\delta > 0$, $\varepsilon > 0$ (small values)

Set $t = \log_2\left(\frac{1}{\delta}\right)$, $k = \frac{2}{\varepsilon}$

choose t independent random hash functions $h_1, \ldots, h_t$; $h_i : [n] \to [k]$

as maps they are deterministic

they map things far away from each other

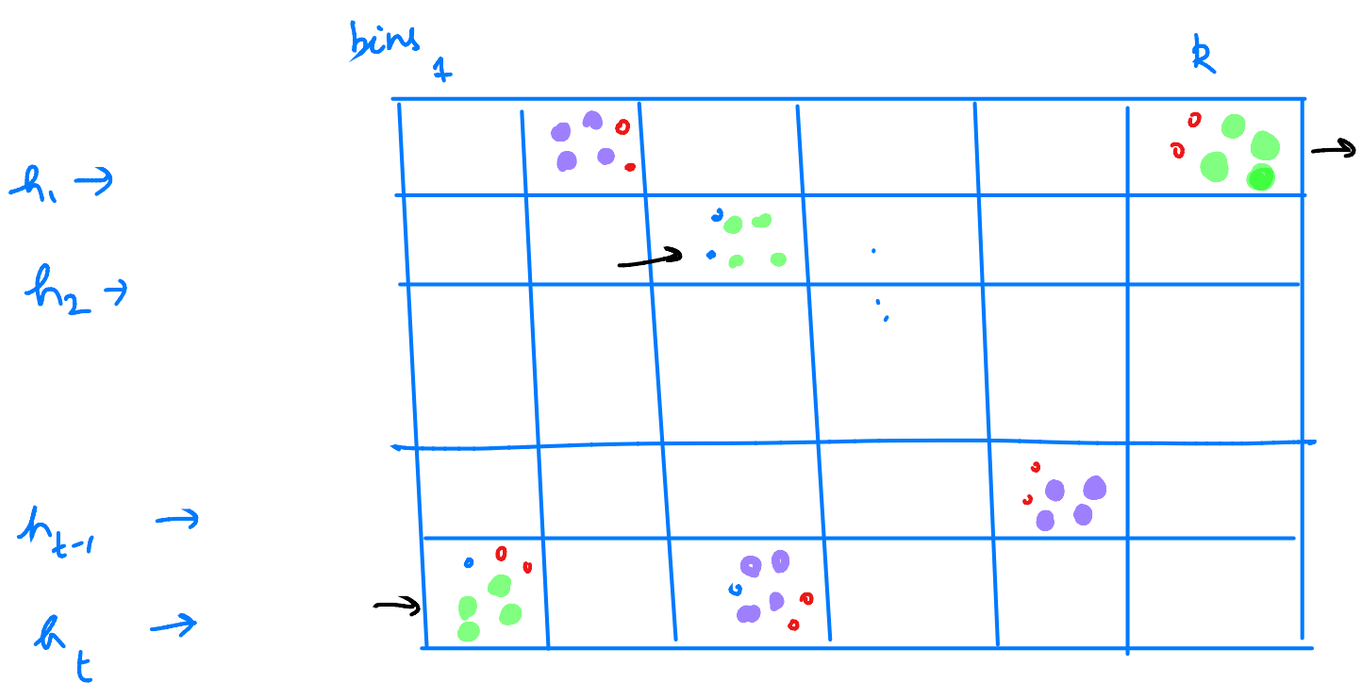


Initialise $t \times k$ counter matrix C s.t $C_{i,j} = 0$

for $i \in [m]$ (sweep token wise)

do for $j \in [t]$ (sweep hash function wise)

do $C_{j, h_j(a_i)} = C_{j, h_j(a_i)} + 1$



Say q =

for a query $q \in [n]$, $\hat{f}_q = \min_{j \in [t]} C_{j, h_j(q)}$

Count Algorithm : like a Bloom filter

$\{1, \ldots, n\}$

Stream $S = \{a_1, \ldots, a_m\}$, $a_j \in \{1, \ldots, n\} = [n]$

for $\delta > 0$, $\varepsilon > 0$ (small values) Set $t = \log_2\left(\frac{1}{\delta}\right)$, $k = \frac{2}{\varepsilon}$

choose t independent random hash functions $h_1, \ldots, h_t$; $h_i : [n] \to [k]$

" $g_1, \ldots, g_t$; $g_i : [n] \to \{+1, -1\}$

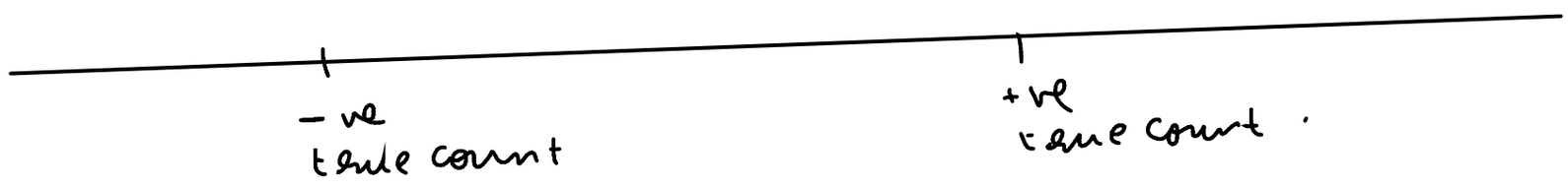
Choose t

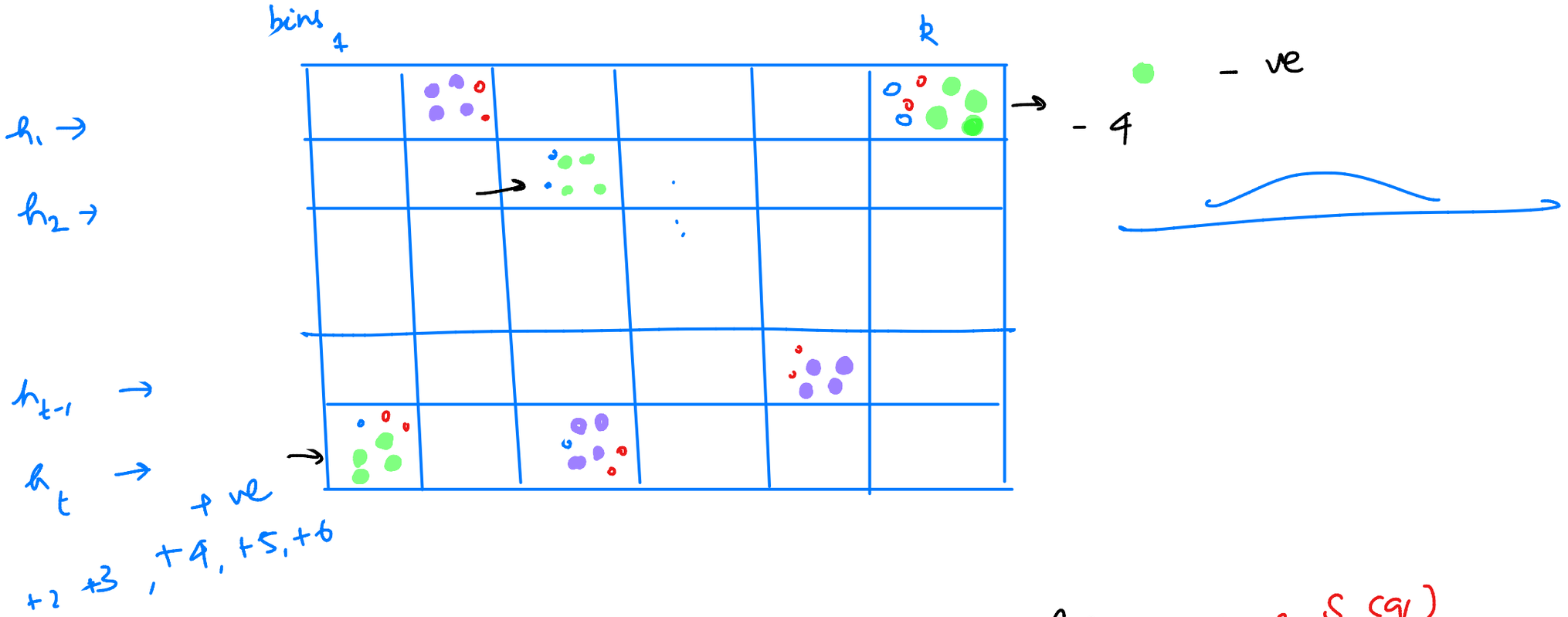
Initialise $t \times k$ counter matrix C s.t $C_{i,j} = 0$

for $i \in [m]$ (sweep token wise)

do for $j \in [t]$ (sweep hash function wise)

do $C_{j, h_j(a_i)} = C_{j, h_j(a_i)} + g_j(a_i)$





for a query $a \in [n]$, $\hat{f}_a$ = median$_j$ $C_{j,h_j(a)} \cdot S_j(a)$

Idea: frequent items = signal = same sign

Non-frequent items = +/- = cancels out each other

Actual = True ± Spread

Estimating Number of distinct elements

- $d = |\{ j : f_j > 0 \}|$

$1, 2, 2, 2, 1, 3, 4, 1, 2,$

- $\text{Zeros}(p) = \max \{ i : 2^i \text{ divides } p \}$

$\text{zeros}(h) = 2, \underbrace{100}$

Tidemark Algorithm

- choose a random hash function

$$h : [n] \to [n]$$

- $z \leftarrow 0$

- Process (token j)

if zeros $(h(j)) > z$ then $z \leftarrow$ zeros $(h(j))$

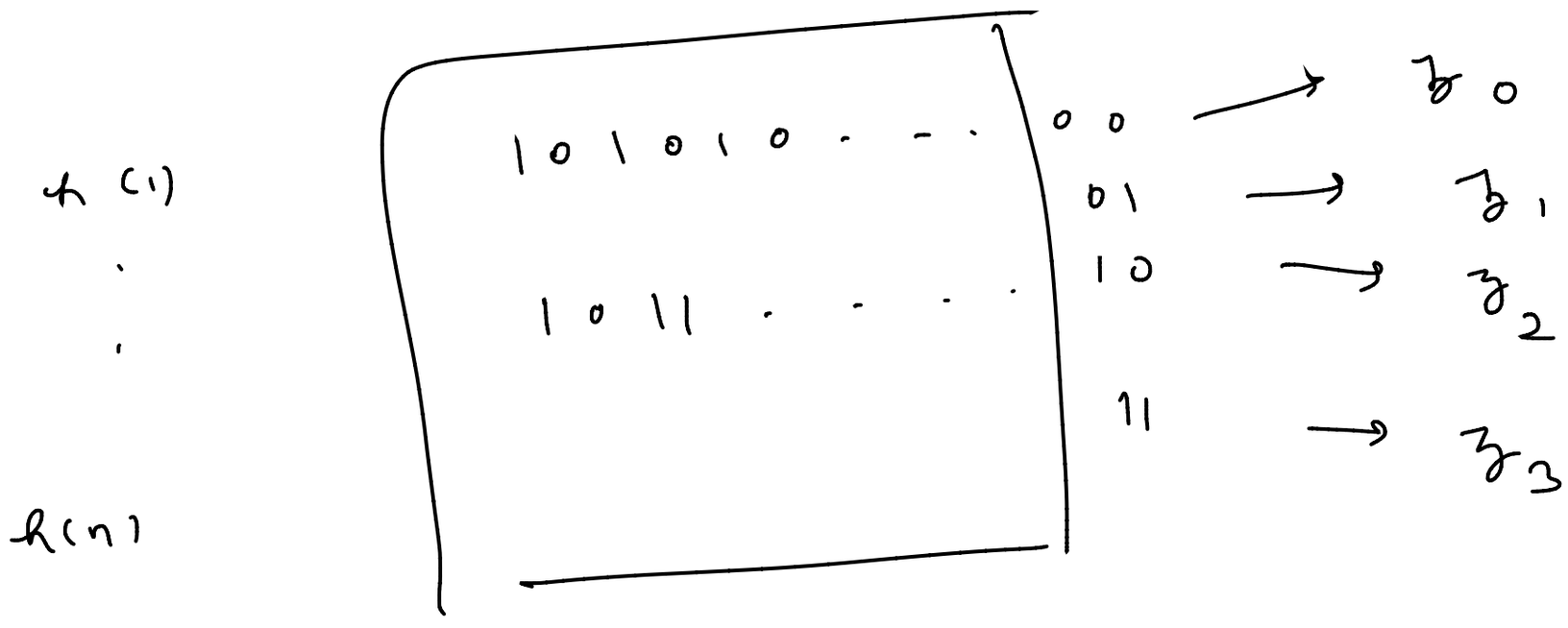
- output $2^{z+\frac{1}{2}}$

$\neq 2^{z+1}$

log Log

★ $z_1, \ldots, z_m$: counters

★ each item is directed to one of the 'm' counters.



$(z_1 + \cdots + z_m)/m$

★ estimate = $m \cdot \sqrt{2}$ 2

Super log

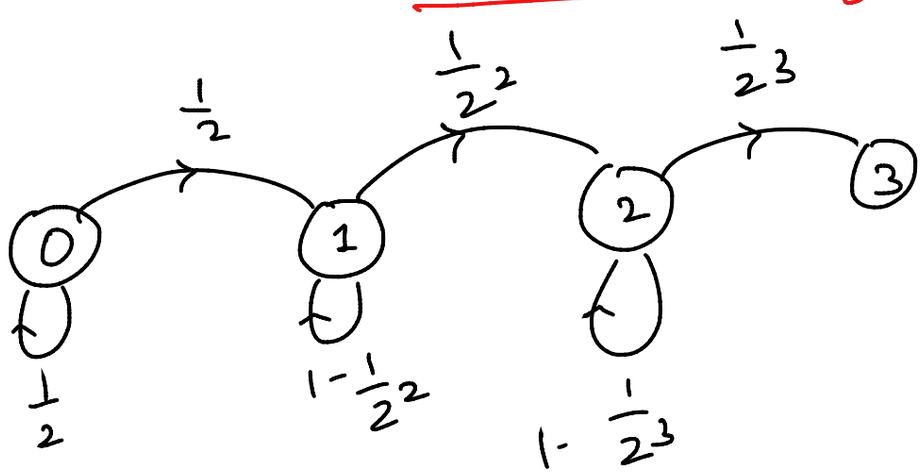
* same as log log

* estimate = $0.7 \cdot m \cdot \sqrt{2} \cdot 2^{\text{Avg (Smallest 70\% of } y_1 \cdots y_m)}$

Hyper log log

* estimate = $m \cdot \sqrt{2} \cdot$ Harmonic mean (2^{y_i})

Idea: Simulate coin toss $\longrightarrow$ p = prob of getting 0



- Probability of getting k consecutive heads with a fair coin is $\frac{1}{2^k}$

- to see one such sample we need 2^k trials.

count = 2 $\rightarrow$ count = 3

3 consecutive 0s

$h(x)$ $\longrightarrow$ one coin toss sequence $\longrightarrow$

1 1 0 1 1 0 1 0 ✗

. . . 1 1 0 0 ✗

- - . 1 0 1 ✗

- . 0 0 0 ✓

•

$h(1) \rightarrow$ 1 0 0 0 1 0 1

$h(2) \rightarrow$ 1 0 1 1 1 0 1 0

⋮

$h(n) \rightarrow$

- to move from $j-1$ to j, the counter has to see 2^j elements (on an average)

- At j total distinct elements

$$2 + 4 + \cdots + 2^j = 2(1 + \cdots + 2^{j-1})$$

$$= 2(2^j - 1) = 2^{j+1} \times \frac{1}{\sqrt{2}}$$