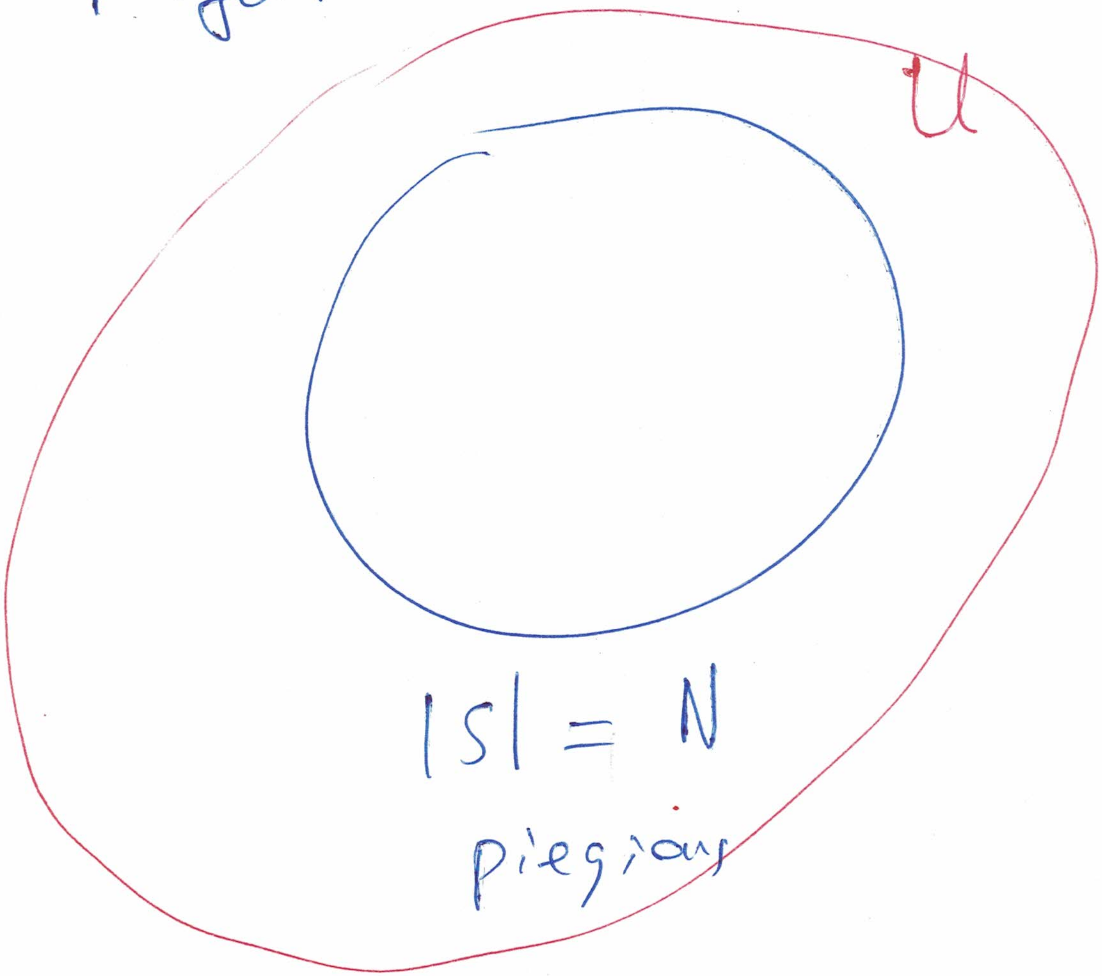
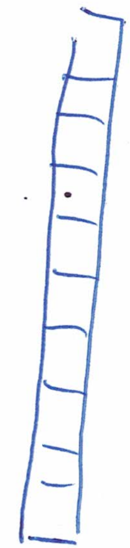
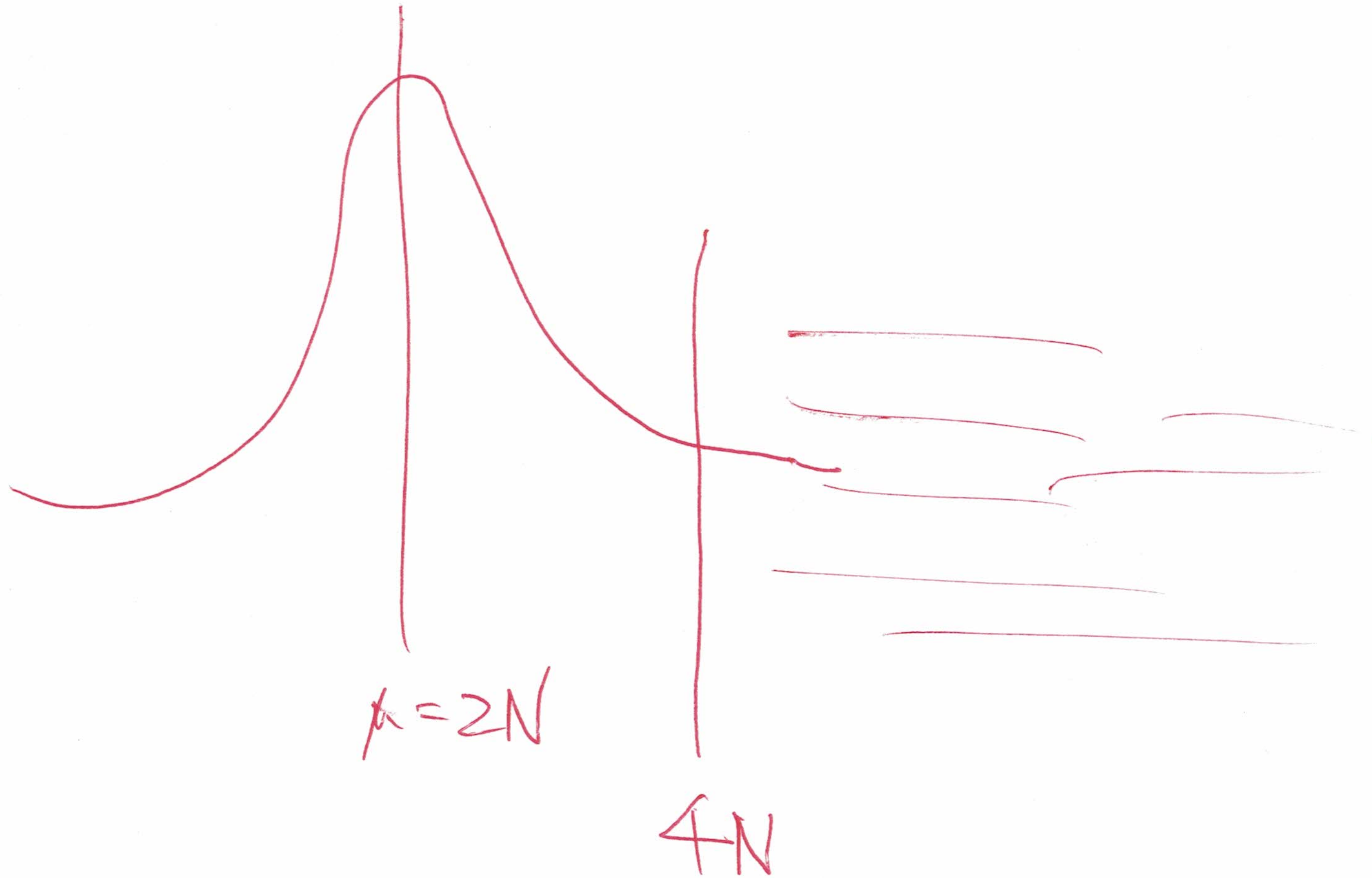
Cpts 515.   10/23/2020.

Perfect Hash. ⟵ no collision.

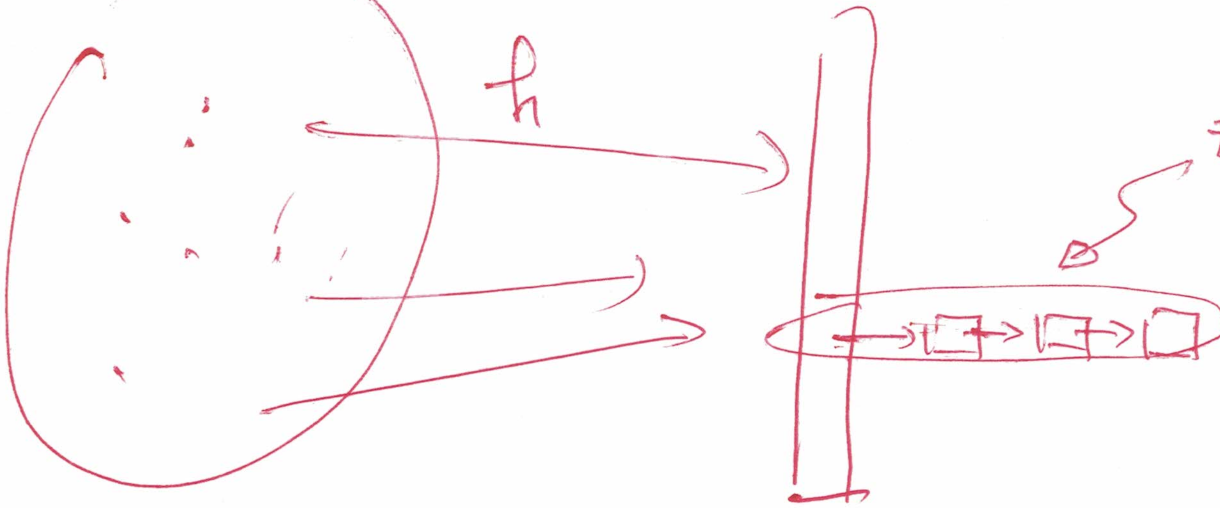




$O(N)$ holes.



$\mu = 2N$
4N



Each bin j contains n_j balls.

$|S| = N$ $\qquad$ $M = \cancel{O(N)} \, N$

Throw N balls to N bins. It's known:

$$\text{Exp}\left(\sum_{j=0}^{N-1} n_j^2 \right) \le 2N.$$

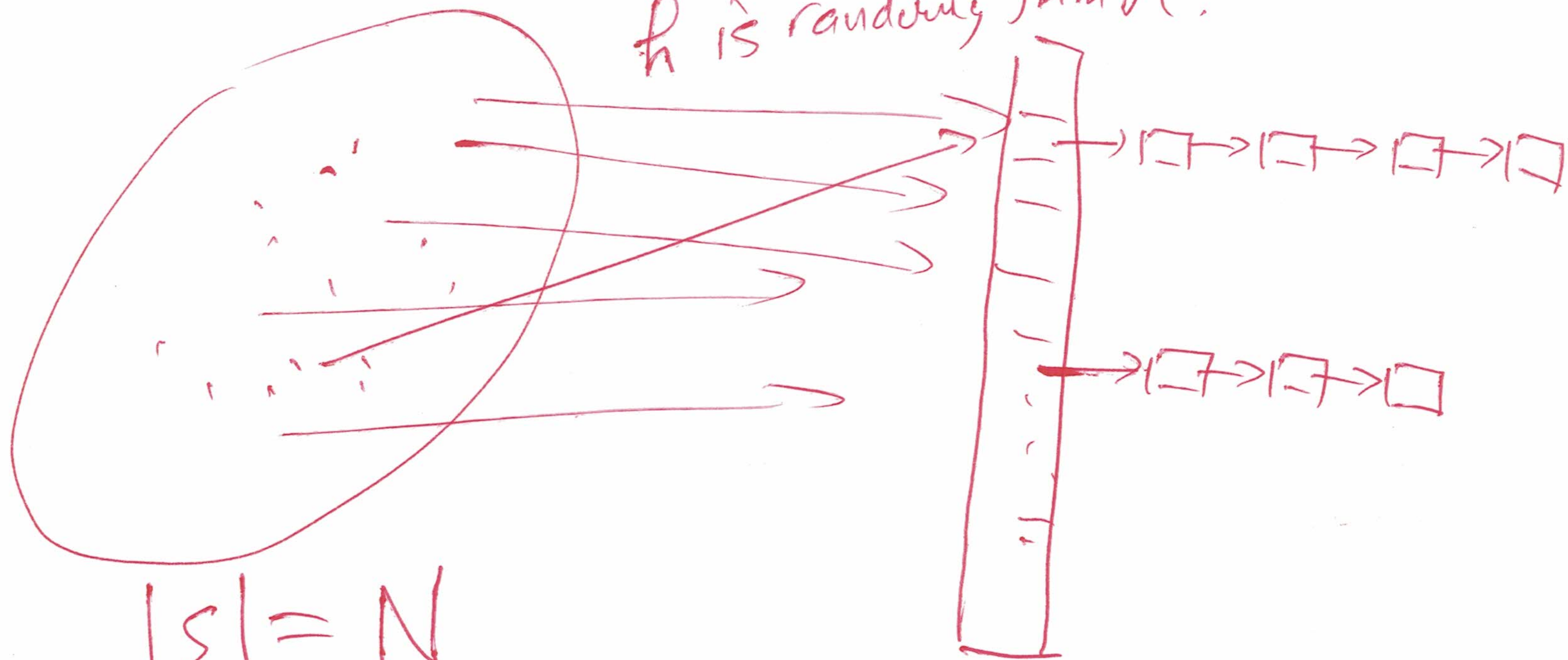
We also have:

$$\text{Prob}\left\{ \sum_{j=0}^{N-1} n_j^2 > 4N \right\} < 1/2.$$

Two steps. $\mathcal{H}$: Universal hash family.

(1).

h is randomly from $\mathcal{H}$.



$|S| = N$

$M = \cancel{O(N)}\, N$.

(2). hash the collisons again (using the $O(N^2)$ no collision method).

Perfect Hashing Alg:

(1). Repeat

select $h \in \mathcal{H}$

Until $\sum_{j=0}^{N-1} n_j^2 \leq 4N$.

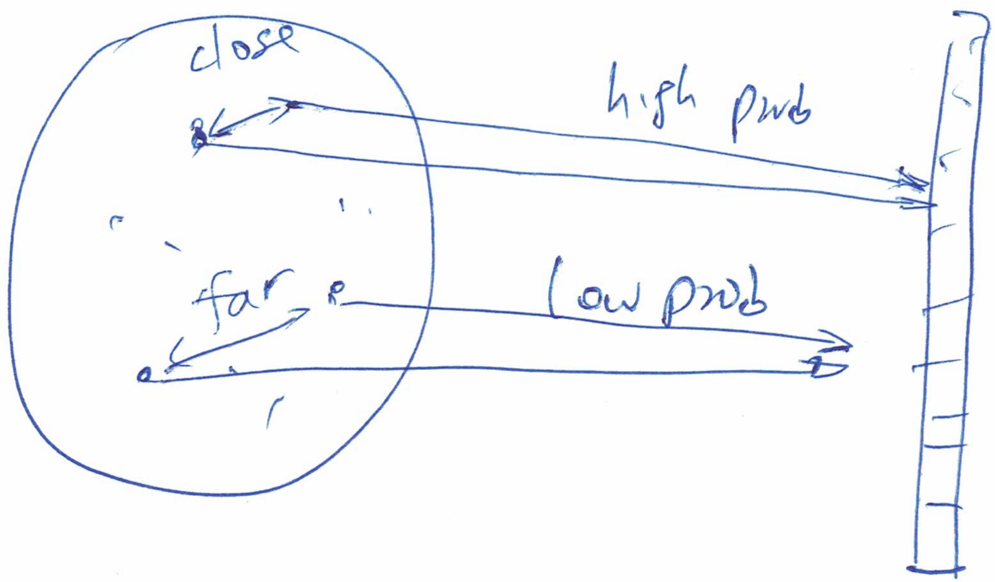
(2). For each j-th slot with n_j keys, ($n_j > 1$). // collison case

select h_j such that the n_j keys are hashed into n_j^2 slots with no collision.

---

total space: $\leq 5N$

Locality Sensitive Hashing.



$\mathcal{H}$: hash family $\mathcal{U} \longrightarrow \mathbb{N}$. where $\mathcal{U}$ is a metric space. Let h be random on $\mathcal{H}$ and

(1).

(1). if $d(q_1, q_2) \leq r_1$ Then

$$\text{Prob}\{h(q_1) = h(q_2)\} \geq p_1;$$

(2). if $d(q_1, q_2) \geq r_2$ then

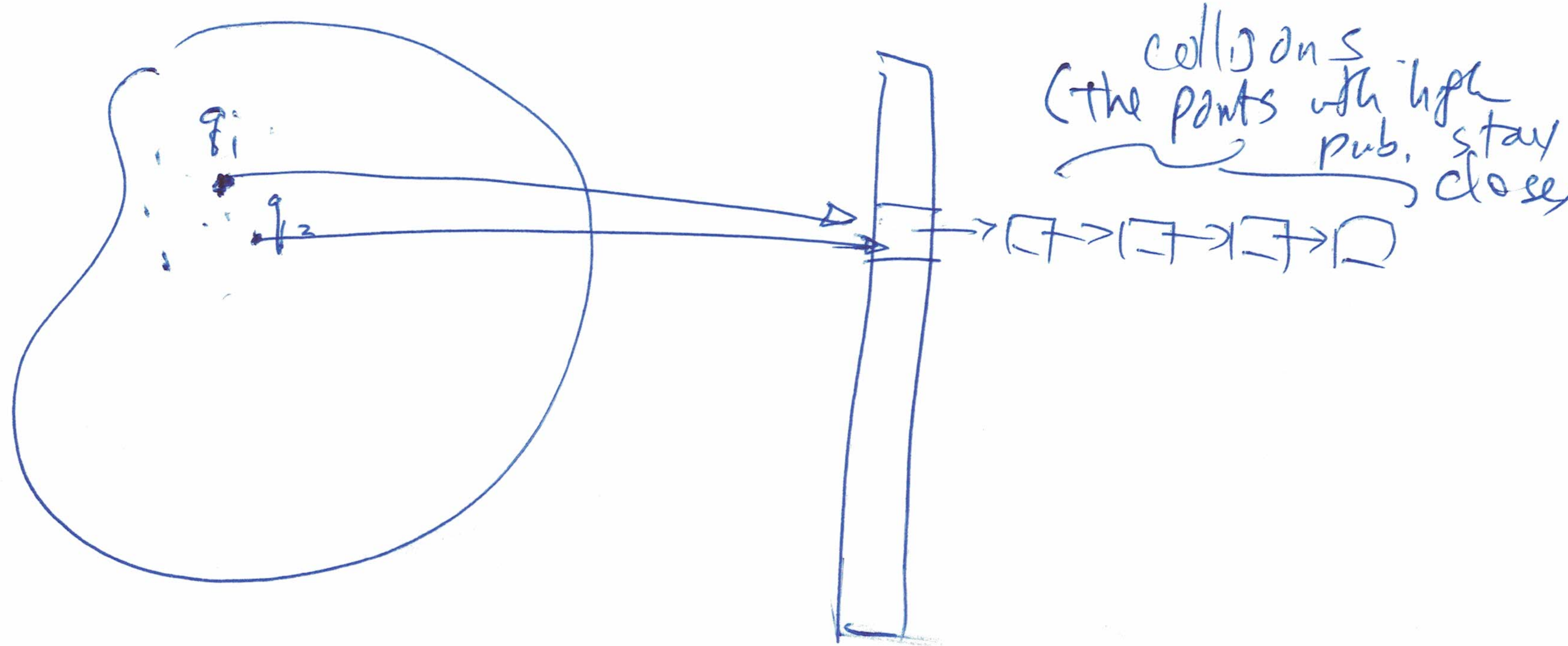
$$\text{Prob}\{h(q_1) = h(q_2)\} \leq p_2,$$

where $q_1, q_2 \in \mathcal{U}$, $p_1 > p_2$, and, $r_1 < r_2$.

Then, $\mathcal{H}$ is called (r_1, r_2, p_1, p_2)-sensitive.

Given $q_1, \ldots, q_n$ where n is large

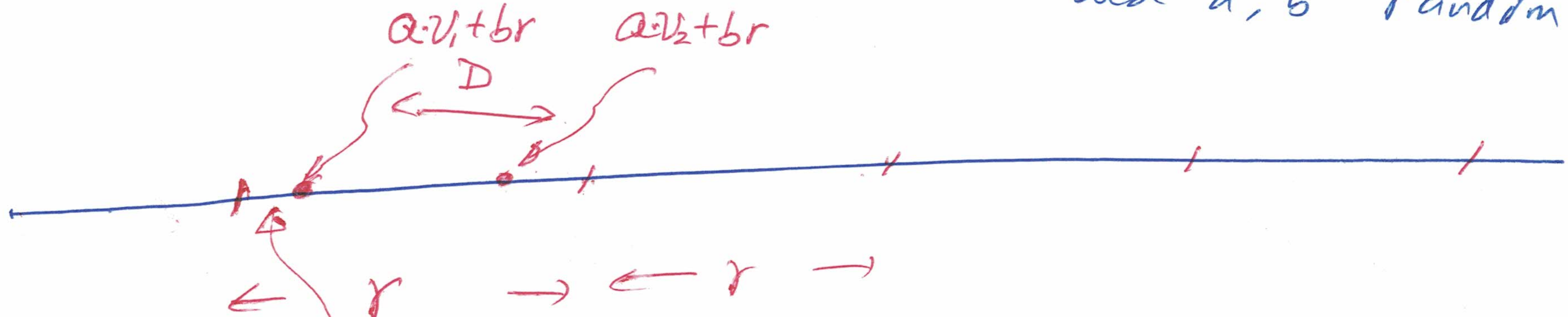
find these (q_i, q_j) pairs with $d(q_i, q_j)$ small. How?


q_1
q_2
collisions
(the points with high prob. stay close)

Intuition.

$$h_{a,b}(v_1) = h_{a,b}(v_2) \quad \text{with } v_1, v_2 \text{ fixed}$$

and a, b random.



iff (1). $\dfrac{a \cdot (v_1 - v_2)}{r} \in [0, 1)$ // $D \in [0, r)$.

(2). $br \in [0, r - D]$.

Read SCG'04 with the $p(c)$ derived from (1) and (2).

(Datar etc, SCG'04).

Let $\mathcal{U} = \mathbb{R}^d$. $\mathcal{H}$: The family of all

$h_{a,b}(\cdot)$, where

1. $a \in \mathbb{R}^d$ with each component a_i is chosen randomly from normal distribution (with $\mu = 0$, $\sigma = 1$).
2. b is a random number in $[0, 1]$.
3. r is some positive number.
4. $h_{a,b}(v) = \left\lfloor \frac{a \cdot v}{r} + b \right\rfloor$

// $v \in \mathbb{R}^d$, a vector

Bloom filter.

(a data structure to rep. a set.).

Define

$$h(a, x) = \left(\begin{array}{l}\text{rotate } x \text{ to the}\\ \text{left by } a \text{ bits}\end{array}\right) \bmod 16.$$

Example. $x = 10011000010$.

$a = 3$.

$$11000\underbrace{010100} \bmod 16$$

$$h(a, x) = 0100 = 4.$$

We are given:

$$h_1 = h(2, x)$$

$$h_2 = h(4, x)$$

$$h_3 = h(3, x).$$

Consider a set $\{x_1, x_2\} = \{11001000, 0100010011\}$.

$h_1(x_1) = 3$	$h_1(x_2) = 13$
$h_2(x_1) = 12$	$h_2(x_2) = 4$
$h_3(x_1) = 6.$	$h_3(x_2) = 10.$

I add x_1 to Bloom filter;

0	0	0	1	1	0	1	0	0	0	1	0	1	1	0	0	0
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

I add x_2 to Bloomfilter:

Given: x

Query: $x \in$ the set?

Compute $h_1(x)$, $h_2(x)$, $h_3(x)$.
and see if the $h_1(x)$-bit, $h_2(x)$-bit and $h_3(x)$-bit are all set to 1. in the Bloom filter. if yes, set $x \in$ the set.
set $x \notin$ the set.

False positive: when query x for $h_1(x), \ldots, h_k(x)$, all the bit $= 1$ at positions $h_1(x), \ldots, h_k(x)$, the probability of false positive:

$$\left(1 - \left(1 - \frac{1}{m}\right)^{kn}\right)^k \approx \left(1 - e^{-\frac{kn}{m}}\right)^k.$$

In practice, how to choose the k?

$$k \approx \frac{m}{n} \cdot \ln 2.$$