hashtables.

Bloom Filters

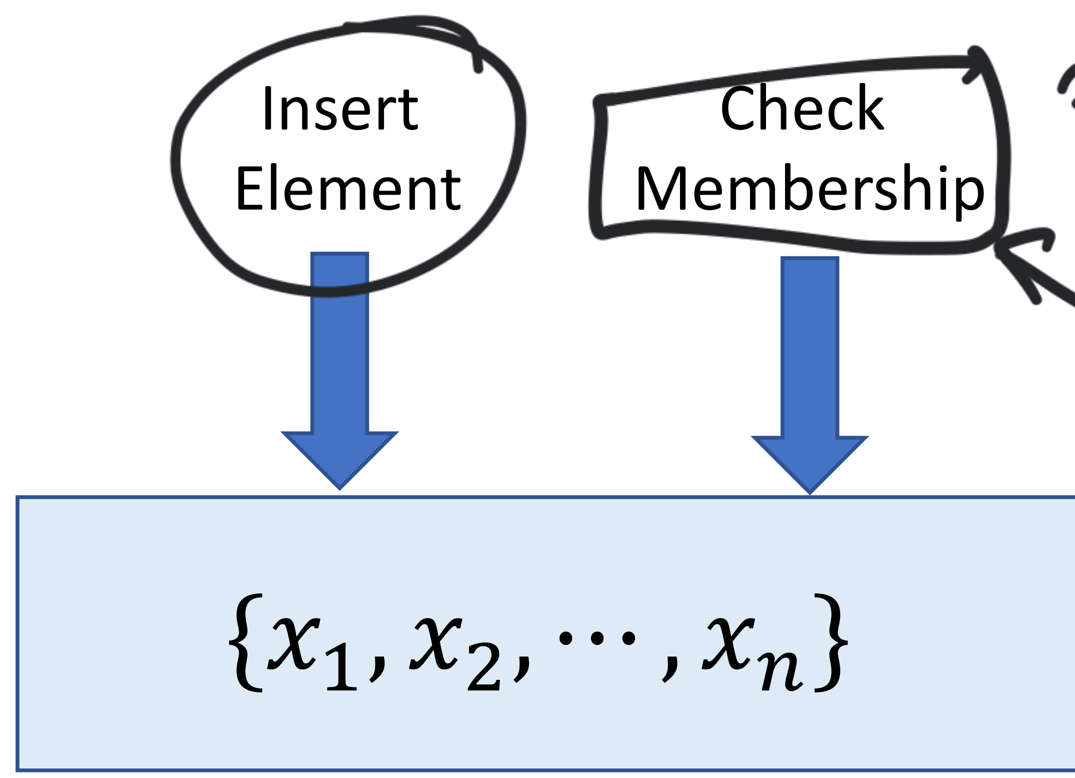
Sriram Sankaranarayanan

Data Structures and Algorithms

What is a Bloom Filter?

$\{x_1, \dots, x_n\}$

- A fast set data structure based on hashing.



x_{10}?
x_{200}?

true → may belong to / not

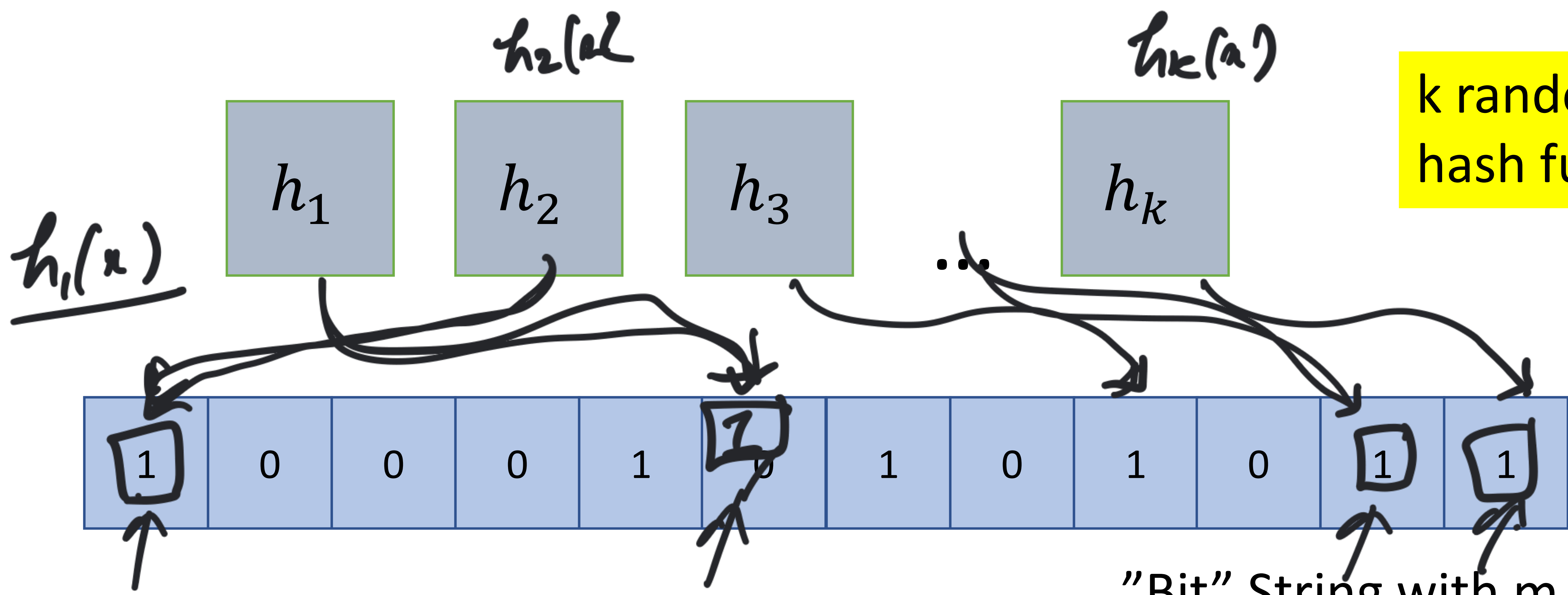
false → definitely does not belong

- Based on hash-tables.
- Approximate in nature: false positives possible.

true

False Positives.

Basic Idea



k randomly chosen hash functions

"Bit" String with m bits.

Insert element x: Set the bits $h_1(x)$, $\boxed{h_2(x)}$, …, $h_k(x)$

signature

Membership of element x: Are the bits $\boxed{h_1(x)}$, $\boxed{h_2(x)}$, … , $\boxed{h_k(x)}$ all set to 1?

Bloom Filter: Properties

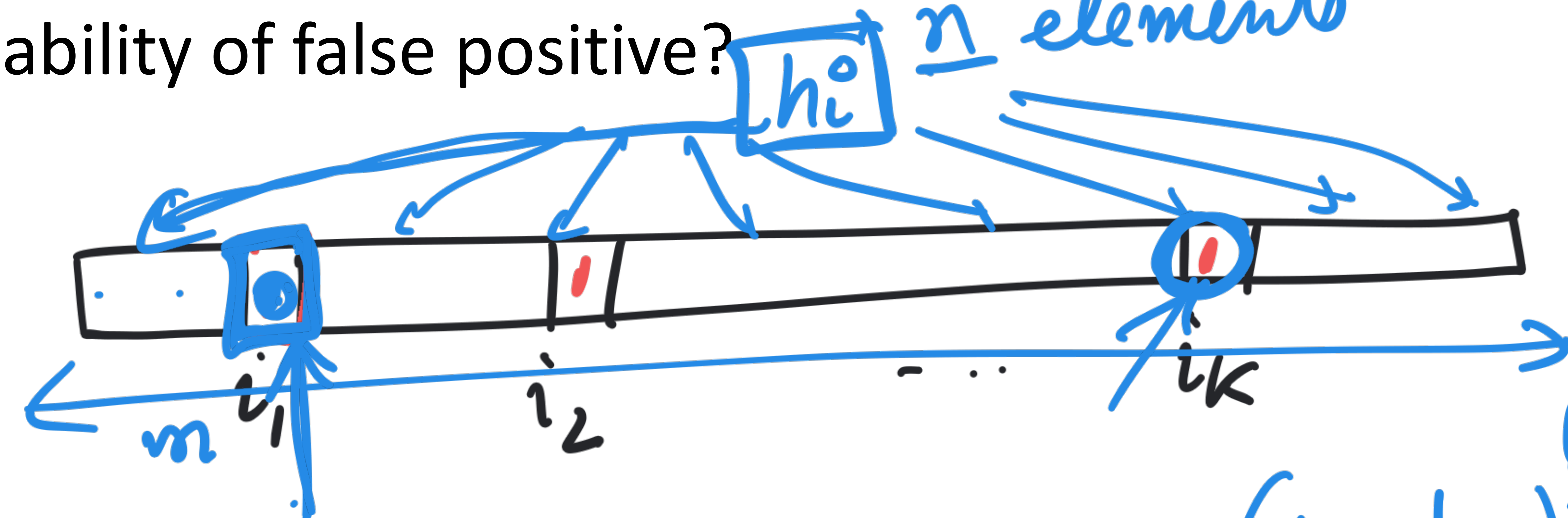
- Constant time insertion and membership check
  - More precisely $\boxed{\Theta(k)}$
- If element $\boxed{\text{was inserted}}$, membership query will return true.
- False positives possible.
  - Membership query may return true but element may not have been inserted.

Bloom Filter: Analysis

- Probability of false positive?

$\underline{k}$ functions

$\underline{n}$ elements.



kn

$\lim_{m \to \infty} (1 - \frac{1}{m} \ldots$

$\approx e \ldots$

$$\Pr(\text{bit } i_1 \text{ } \underline{\text{was not set}}) = \left(1 - \frac{1}{m}\right)^{kn} \approx \boxed{e^{-kn/m}}$$

$$\Pr(i_1 \text{ was set}) = \left(1 - e^{-kn/m}\right)$$

$$\Pr(\underline{i_1 \text{ was set}} \text{ and } \underline{i_2} \ldots \underline{i_k}) = \left(1 - e^{-kn/m}\right)^k \Leftarrow \text{False positive}$$

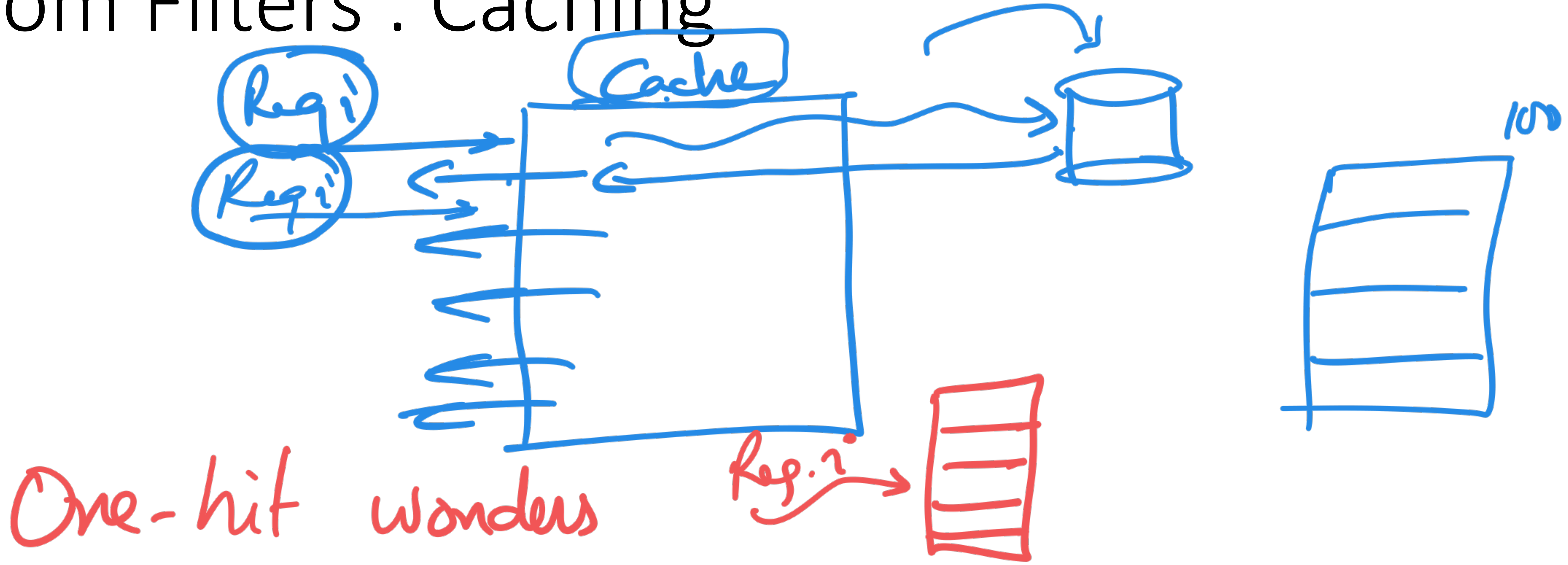
Bloom Filter By Numbers

- n = 5,000 strings (these could be long strings) inserted
- m = 25,000 bit vector size (5 bits/element)
- k = 3 hash functions.

- Probability of false positives is

$$\boxed{\left(1 - e^{-\frac{kn}{m}}\right)^k} = (1 - e^{-0.6})^3 = 0.09$$

Bloom Filters : Caching



[Maggs, Bruce M.](); [Sitaraman, Ramesh K.]() *(July 2015),* [*"Algorithmic nuggets in content delivery"*]() *(PDF), SIGCOMM Computer Communication Review,* ***45*** *(3): 52–66.*