曝光过滤 & Bloom Filter

王树森

ShusenWang@xiaohongshu.com

http://wangshusen.github.io/

小红书

曝光过滤问题

- 如果用户看过某个物品，则不再把该物品曝光给该用户。
- 对于每个用户，记录已经曝光给他的物品。（小红书只召回1个月以内的笔记，因此只需要记录每个用户最近1个月的曝光历史。）
- 对于每个召回的物品，判断它是否已经给该用户曝光过，排除掉曾经曝光过的物品。
- 一位用户看过 n 个物品，本次召回 r 个物品，如果暴力对比，需要 $O(nr)$ 的时间。

Bloom Filter

- Bloom filter 判断一个物品 ID 是否在已曝光的物品集合中。
- 如果判断为 no，那么该物品一定不在集合中。
- 如果判断为 yes，那么该物品很可能在集合中。（可能误伤，错误判断未曝光物品为已曝光，将其过滤掉。）

参考文献：

- Burton H. Bloom. Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM*, 1970.

Bloom Filter

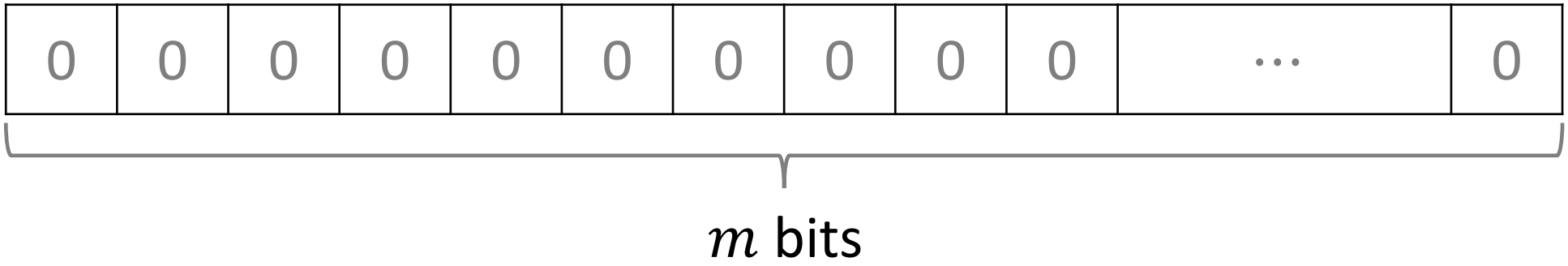
- Bloom filter 把物品集合表征为一个 m 维二进制向量。
- 每个用户有一个曝光物品的集合，表征为一个向量，需要 m bit 的存储。
- Bloom filter 有 k 个哈希函数，每个哈希函数把物品 ID 映射成介于 0 和 $m-1$ 之间的整数。

参考文献：

- Burton H. Bloom. Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM*, 1970.

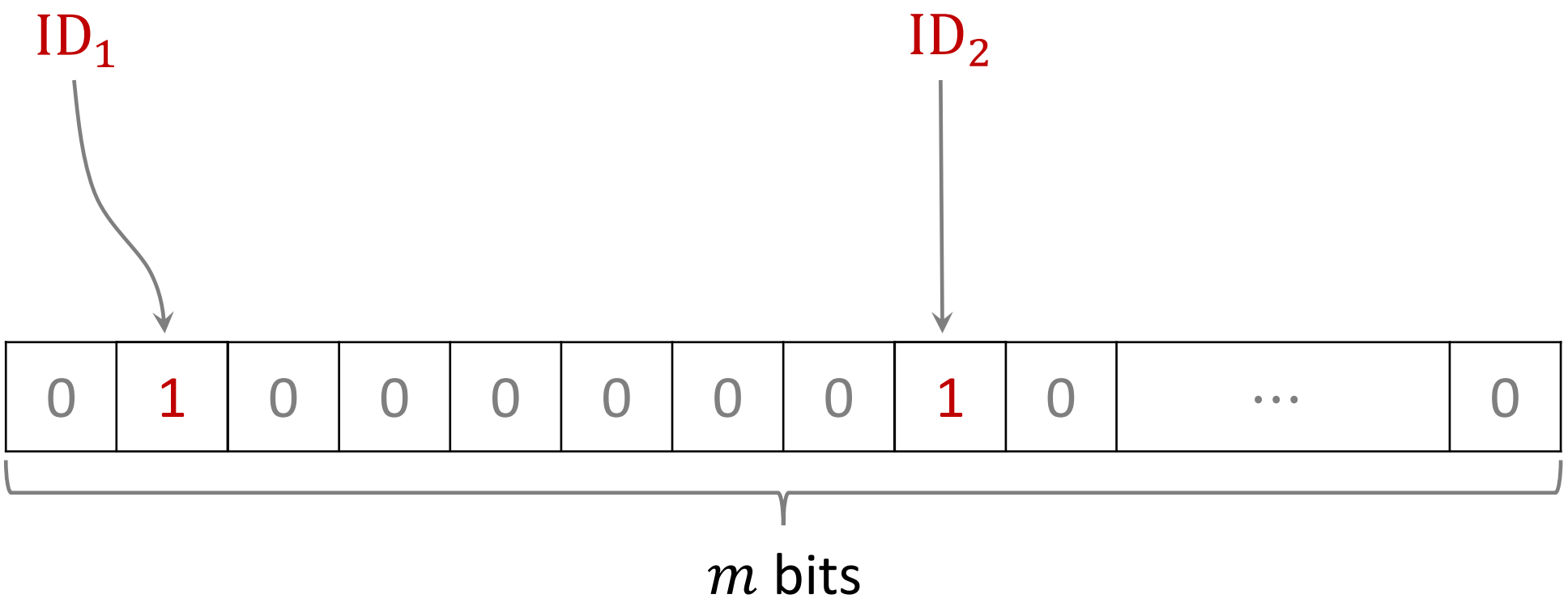
Bloom Filter $(k = 1)$

二进制向量：



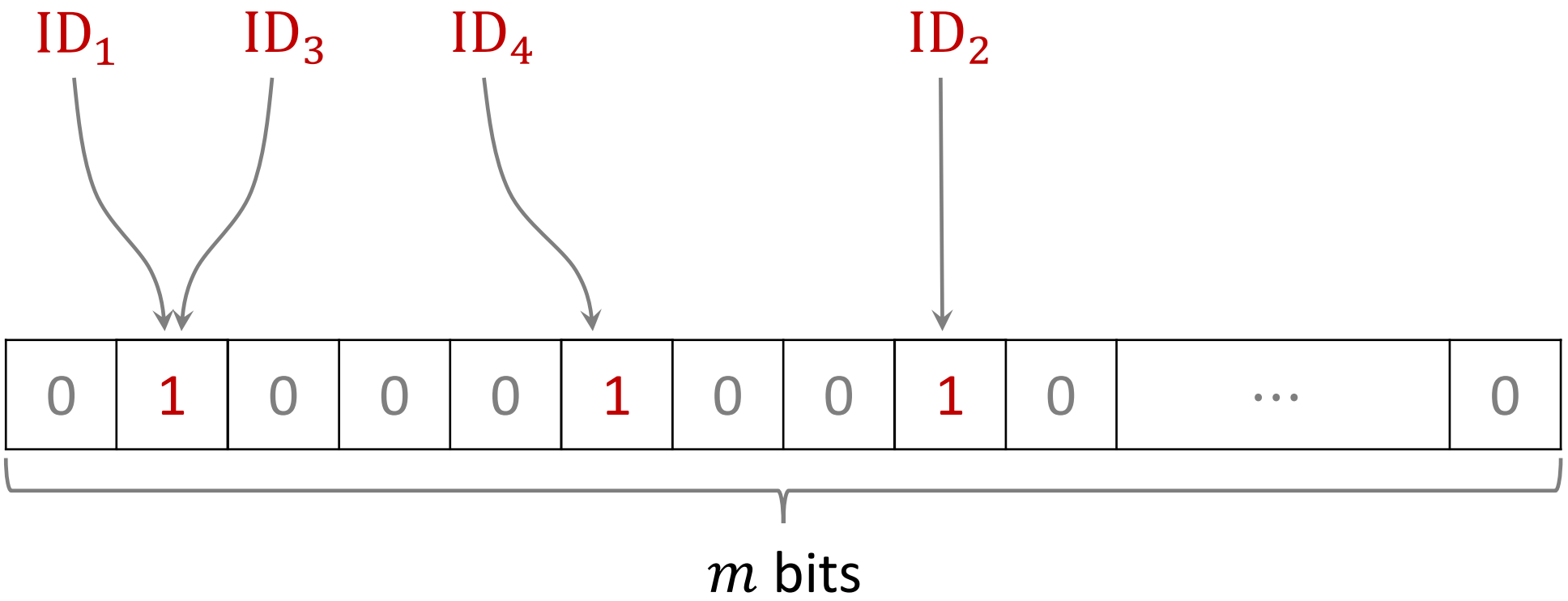
Bloom Filter $(k = 1)$

已曝光物品：



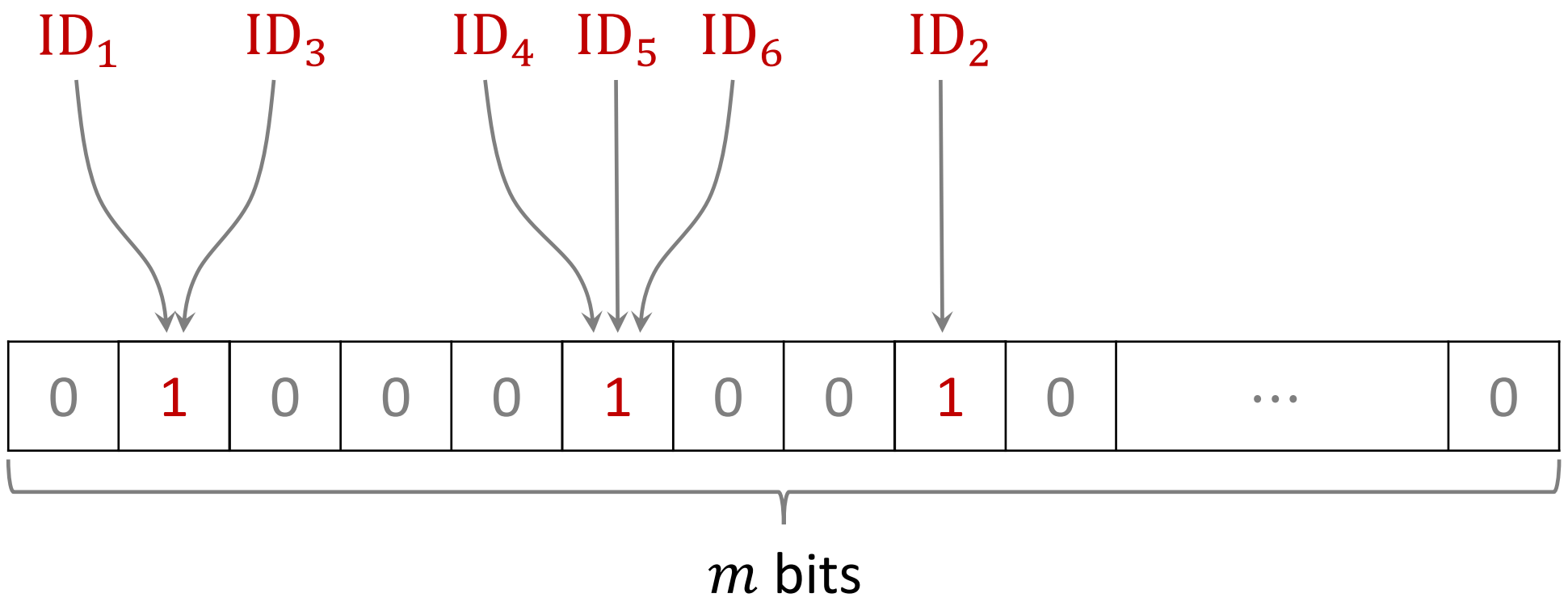
Bloom Filter $(k = 1)$

已曝光物品：



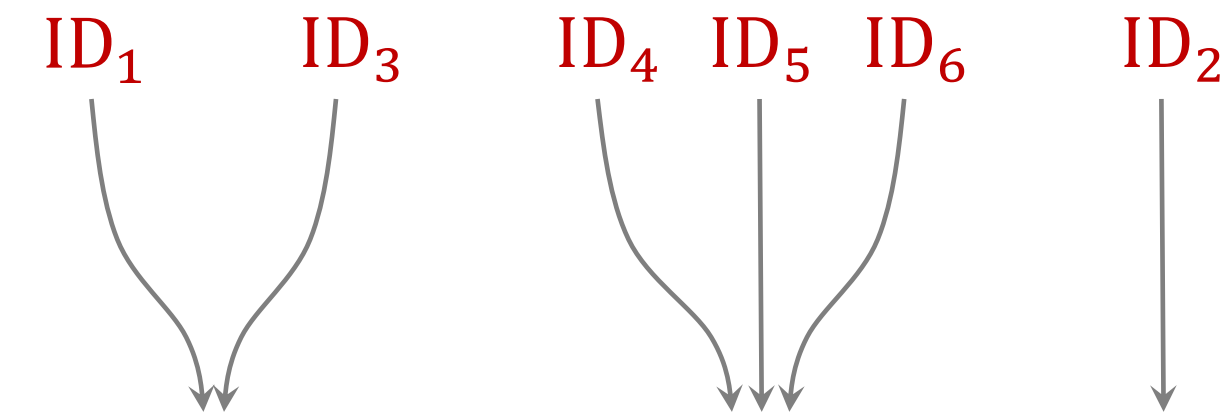
Bloom Filter $(k = 1)$

已曝光物品：



Bloom Filter $(k = 1)$

已曝光物品：



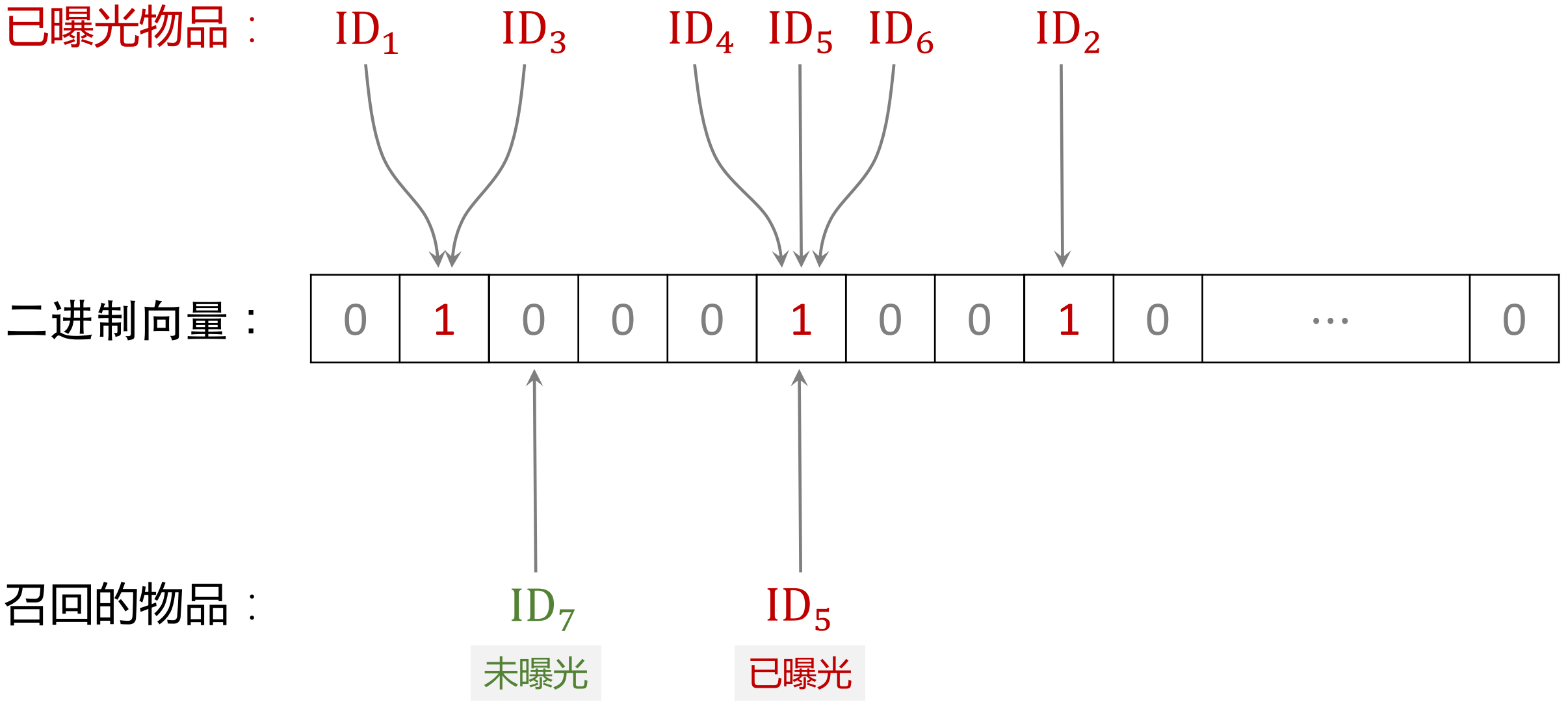
二进制向量：

0	1	0	0	0	1	0	0	1	0	…	0

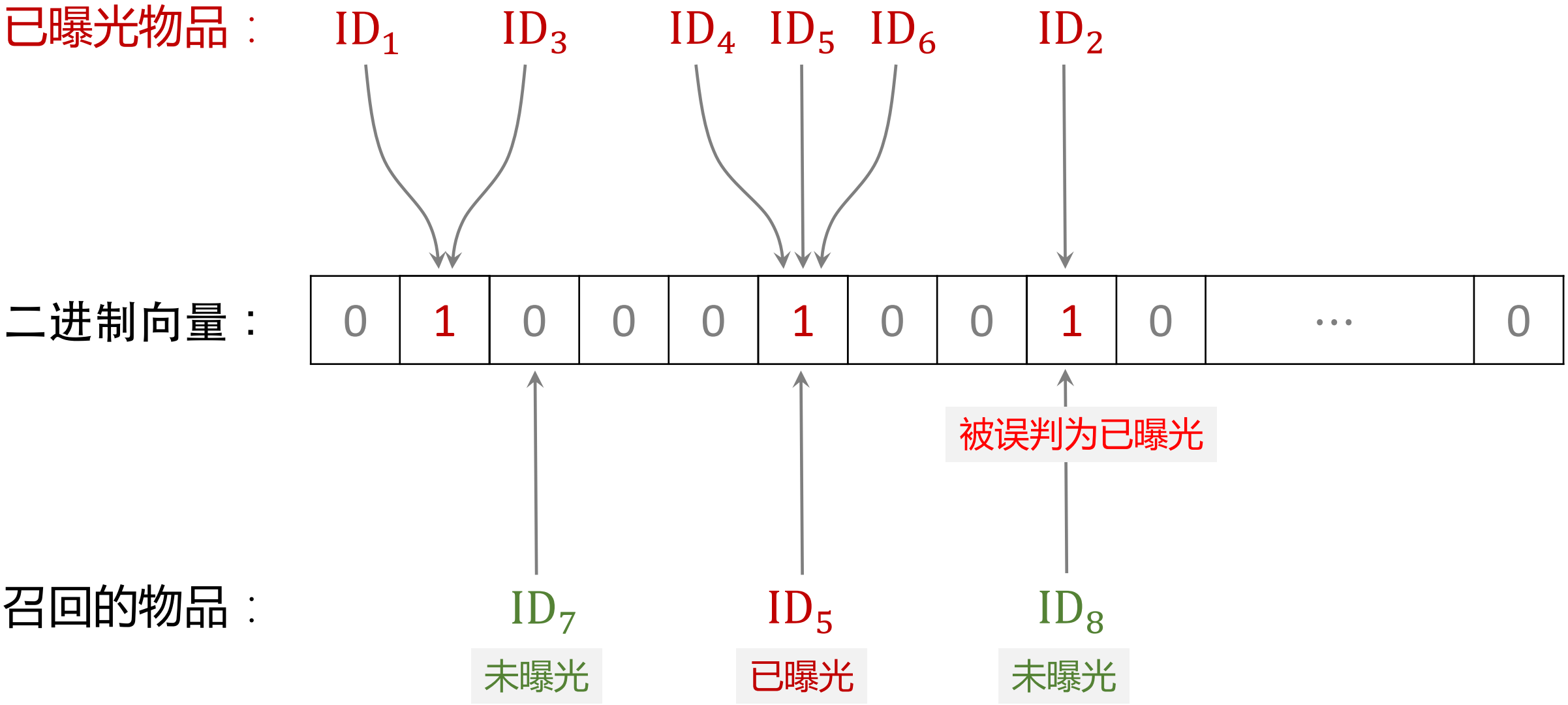
召回的物品：

未曝光

Bloom Filter $(k = 1)$



Bloom Filter $(k = 1)$



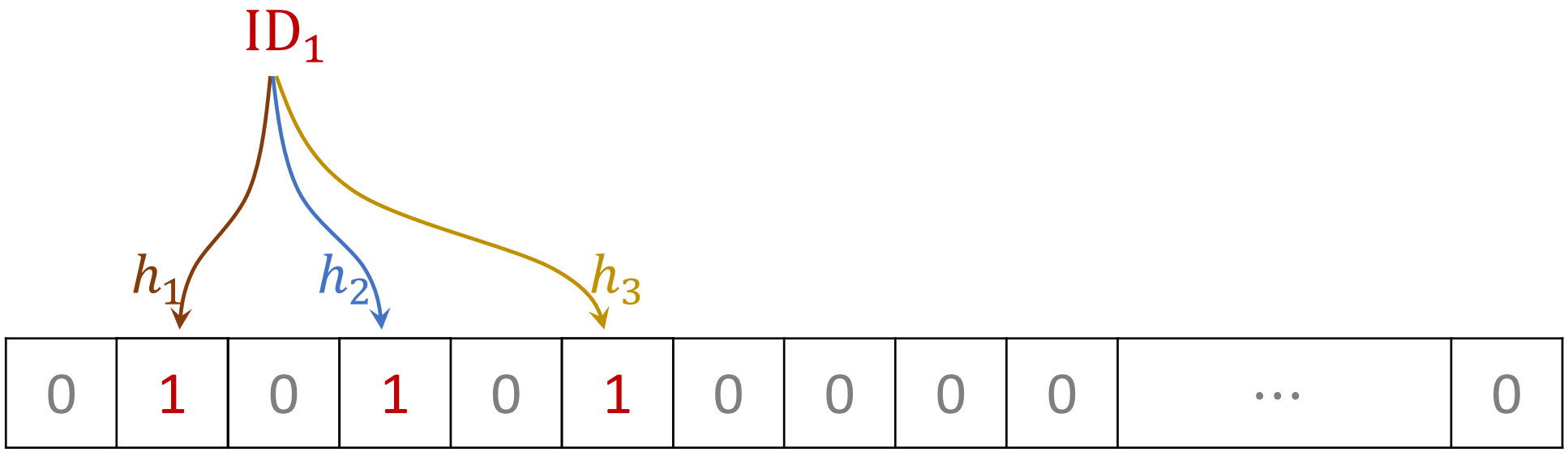
Bloom Filter $(k = 3)$

二进制向量：

0	0	0	0	0	0	0	0	0	0	…	0

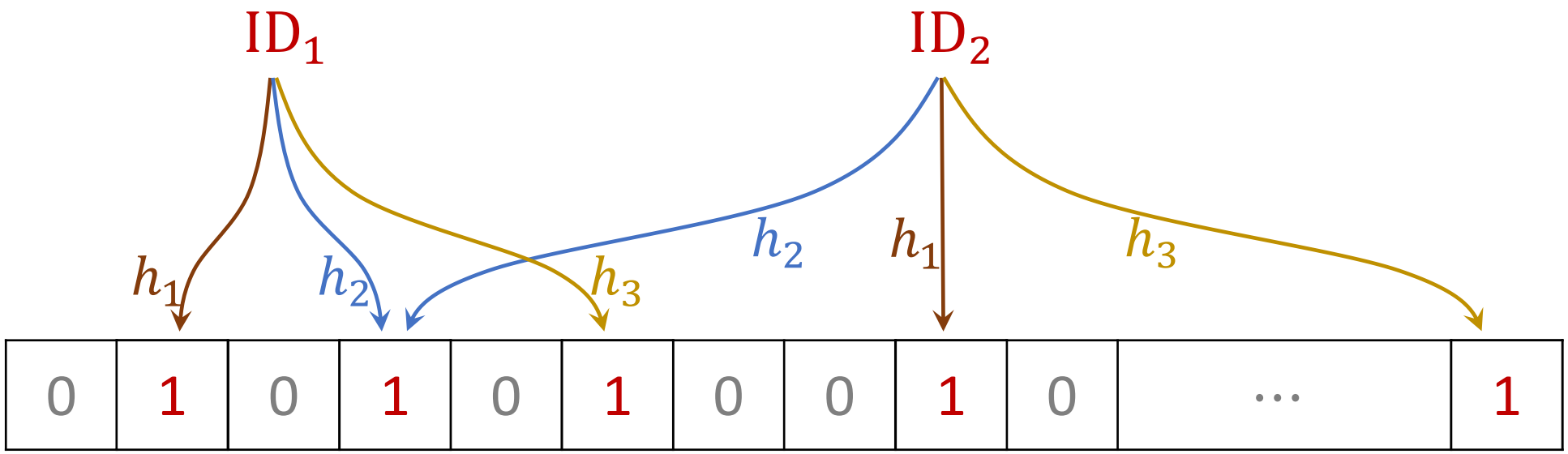
Bloom Filter $(k = 3)$

已曝光物品：



Bloom Filter $(k = 3)$

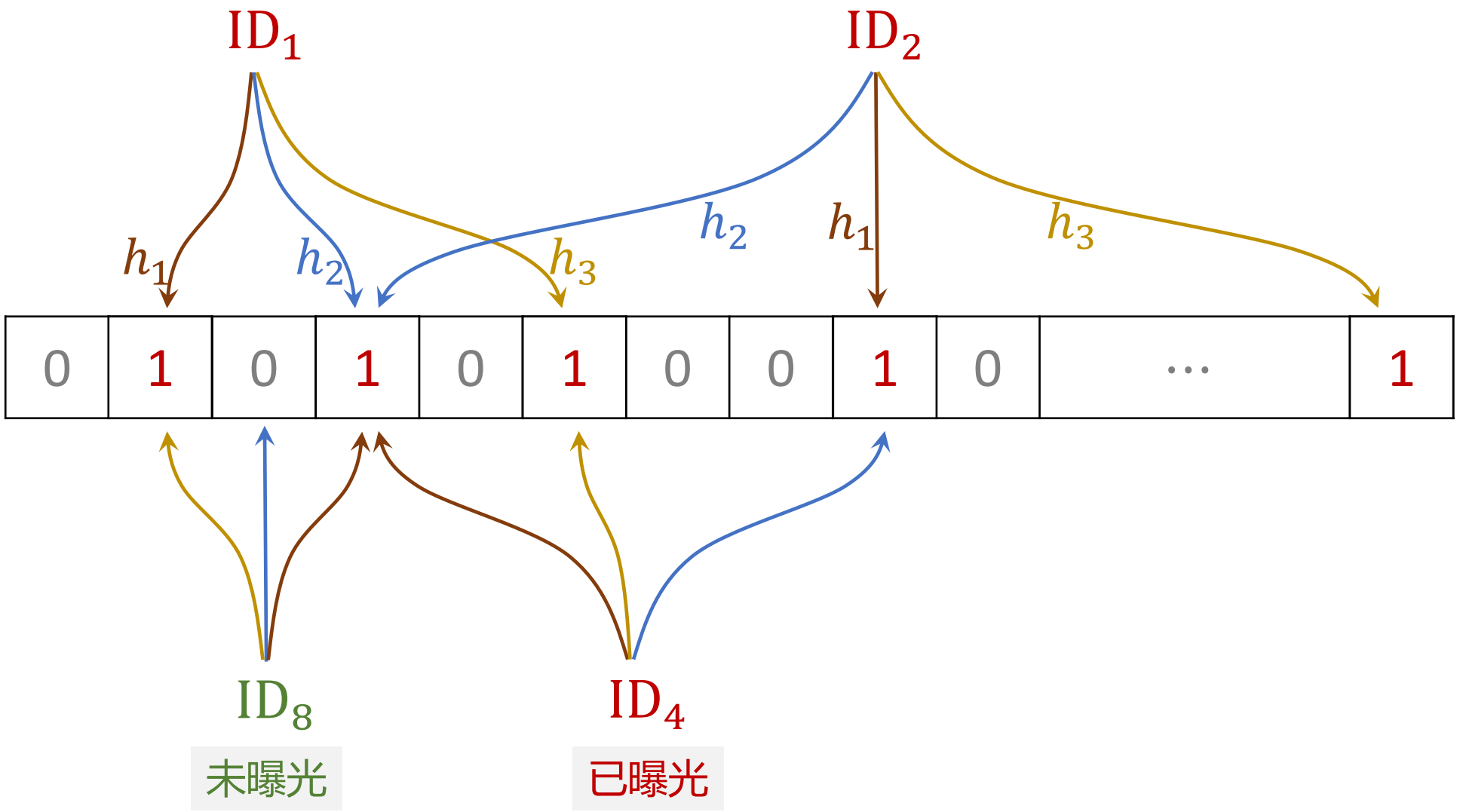
已曝光物品：



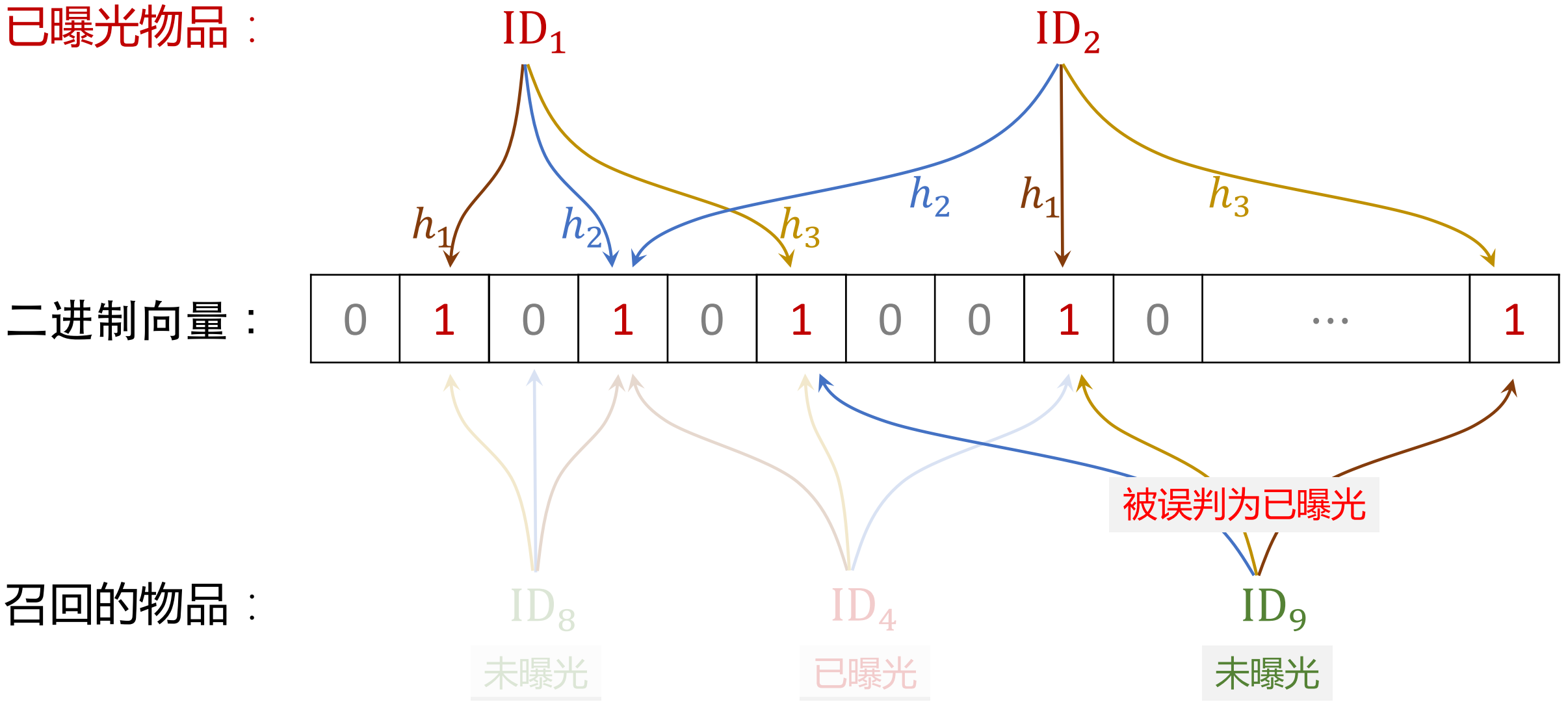
二进制向量：

Bloom Filter $(k = 3)$

已曝光物品：



Bloom Filter $(k = 3)$



Bloom Filter

- 曝光物品集合大小为 n，二进制向量维度为 m，使用 k 个哈希函数。
- Bloom filter 误伤的概率为 $\delta \approx \left(1-\exp\left(-\frac{kn}{m}\right)\right)^k$。
  - n 越大，向量中的 1 越多，误伤概率越大。（未曝光物品的 k 个位置恰好都是 1 的概率大。）
  - m 越大，向量越长，越不容易发生哈希碰撞。
  - k 太大、太小都不好，k 有最优取值。

Bloom Filter

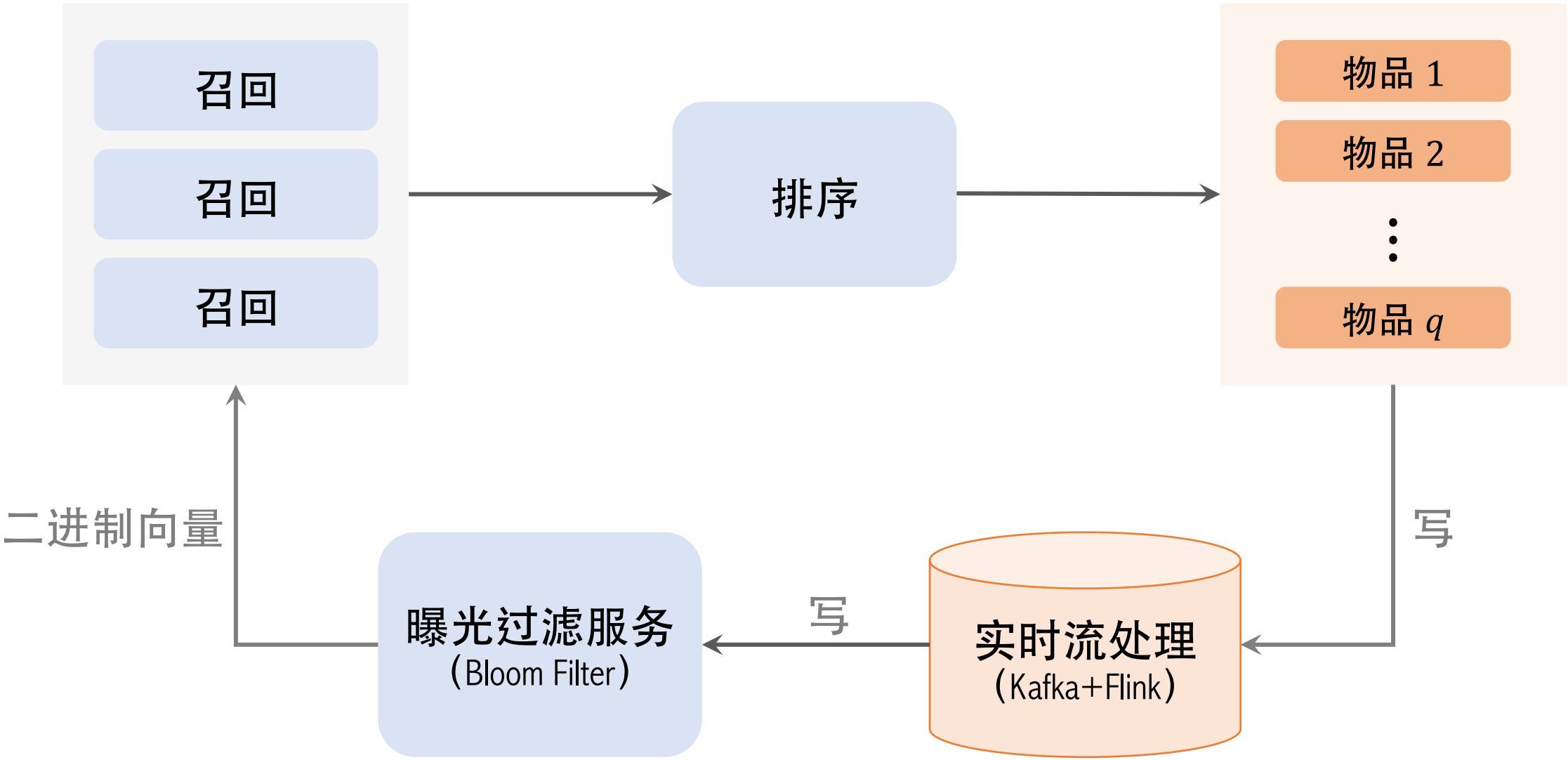
- 曝光物品集合大小为 n，二进制向量维度为 m，使用 k 个哈希函数。
- Bloom filter 误伤的概率为 $\delta \approx \left(1 - \exp\left(-\frac{kn}{m}\right)\right)^k$。
- 设定可容忍的误伤概率为 δ

Bloom Filter

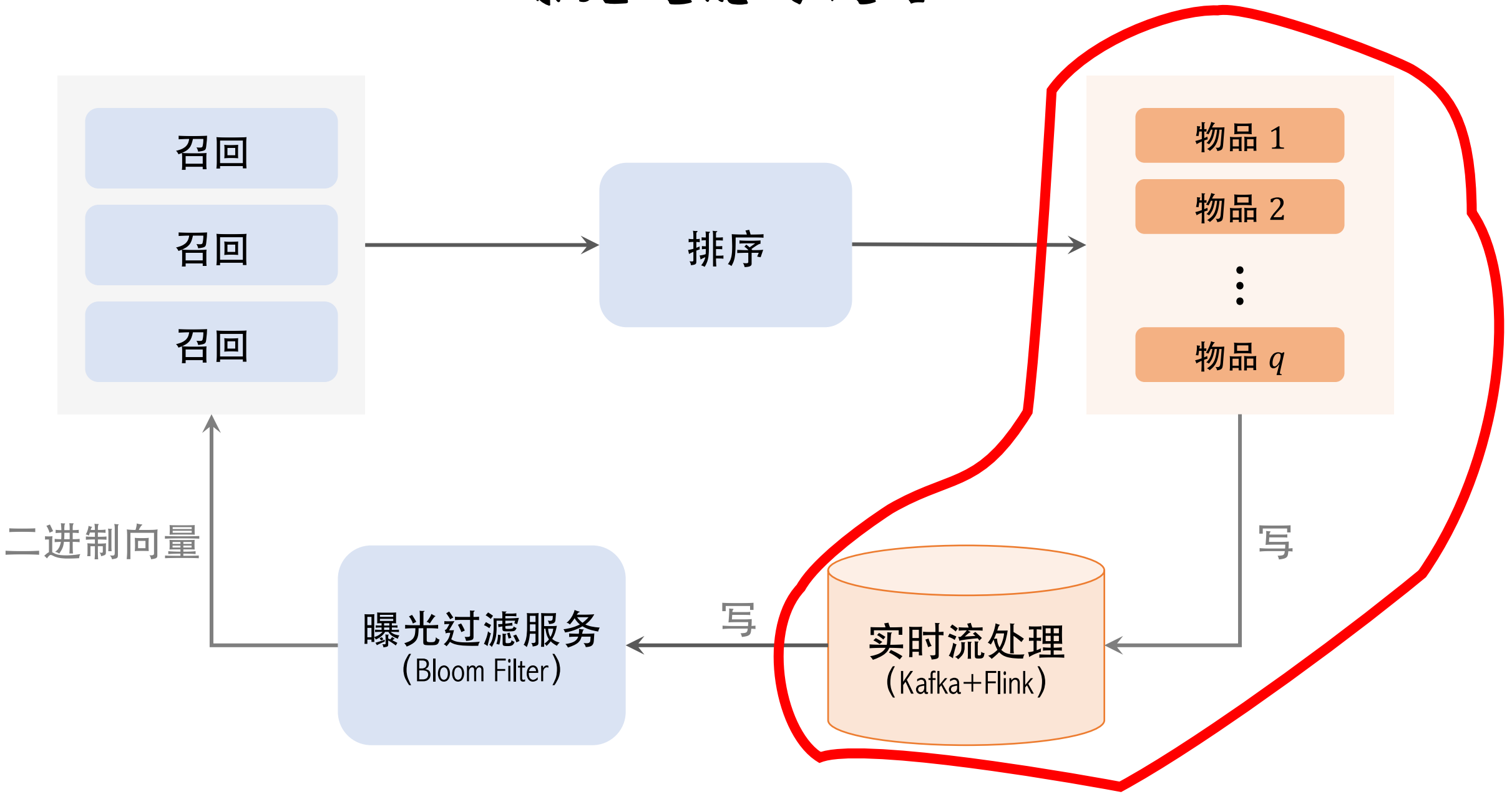
- 曝光物品集合大小为 n，二进制向量维度为 m，使用 k 个哈希函数。
- Bloom filter 误伤的概率为 $\delta \approx \left(1-\exp\left(-\frac{kn}{m}\right)\right)^k$。
- 设定可容忍的误伤概率为 δ，那么最优参数为：

$$k = 1.44 \cdot \ln\left(\frac{1}{\delta}\right), \qquad m = 2n \cdot \ln\left(\frac{1}{\delta}\right).$$

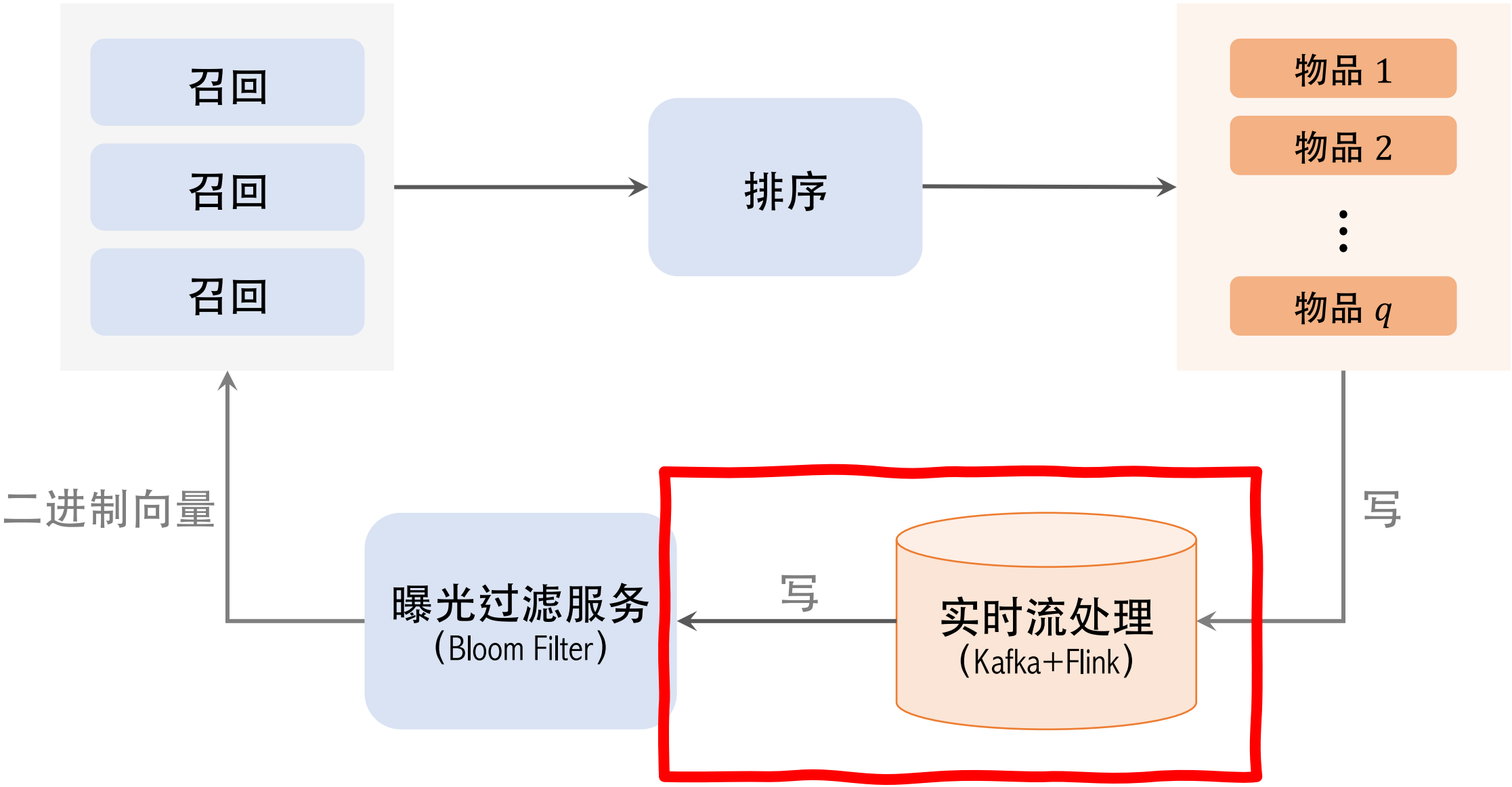
曝光过滤的链路


召回
召回
召回
排序
物品 1
物品 2
物品 q
二进制向量
曝光过滤服务
(Bloom Filter)
写
实时流处理
(Kafka+Flink)
写

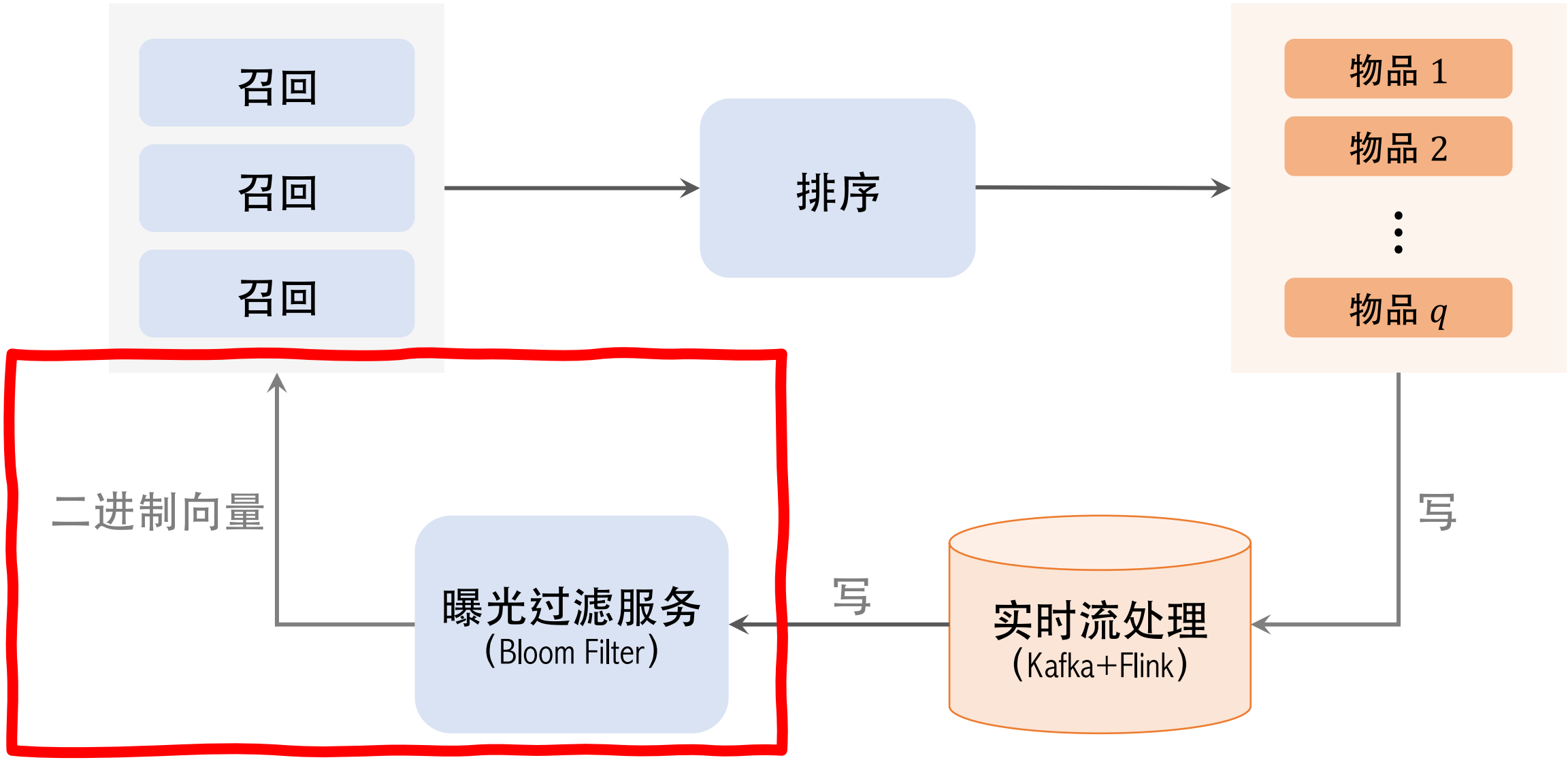
曝光过滤的链路


召回
召回
召回
排序
物品 1
物品 2
物品 q
二进制向量
曝光过滤服务
(Bloom Filter)
写
实时流处理
(Kafka+Flink)
写

曝光过滤的链路


召回
召回
召回
排序
物品 1
物品 2
物品 q
二进制向量
曝光过滤服务
(Bloom Filter)
写
实时流处理
(Kafka+Flink)
写

曝光过滤的链路


召回
召回
召回
排序
物品 1
物品 2
物品 q
二进制向量
曝光过滤服务
(Bloom Filter)
写
实时流处理
(Kafka+Flink)
写

Bloom Filter的缺点

- Bloom filter 把物品的集合表示成一个二进制向量。
- 每往集合中添加一个物品，只需要把向量 k 个位置的元素置为 1。（如果原本就是 1，则不变。）
- Bloom filter 只支持添加物品，不支持删除物品。从集合中移除物品，无法消除它对向量的影响。
- 每天都需要从物品集合中移除年龄大于 1 个月的物品。（超龄物品不可能被召回，没必要把它们记录在 Bloom filter，降低 n 可以降低误伤率。）

Thank You!

http://wangshusen.github.io/