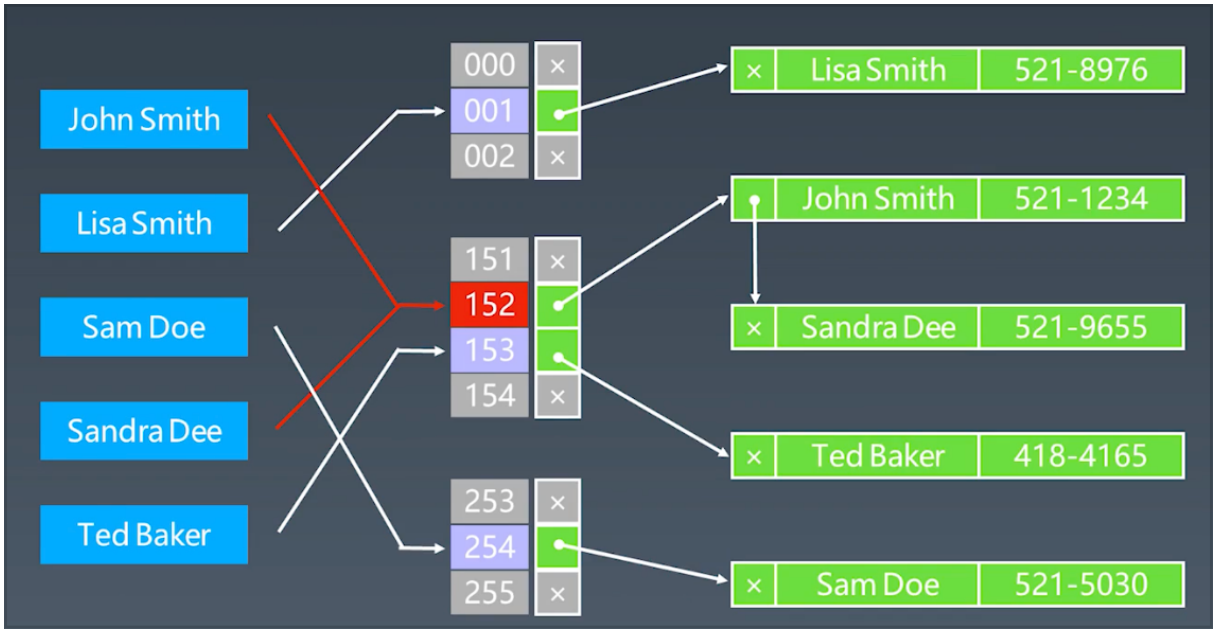
21. 布隆过滤器 Bloom Filter

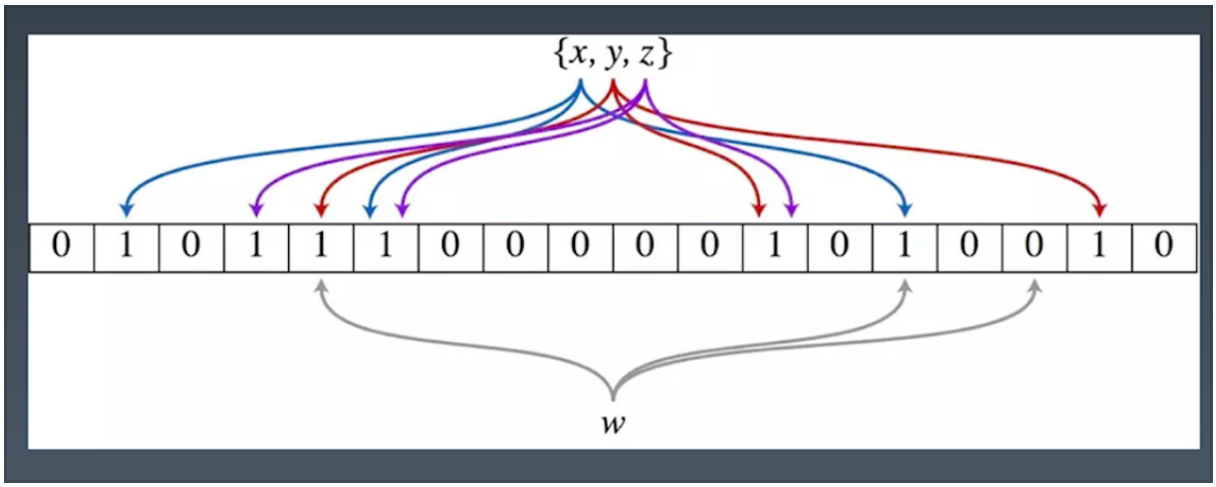
HashTable + 拉链存储重复元素



Bloom Filter

- 布隆过滤器的原理和实现
- 一个很长的 二进制 向量 和 一系列 随机映射函数 ；
- 布隆过滤器可以用于检索一个元素是否在一个集合中：
  - 如果检索的二进制位均为1，则可能存在布隆过滤器中（检索新插入元素时候，可能是之前已经分配过元素的二进为1位，此时，并不能判断新插入元素就是在布隆过滤器中）；
  - 如果检索的二进制位存在一个 0 ，则一定不存在布隆过滤器中.
- 优点：空间效率和查询时间都远远超过一般算法；
- 缺点：有一定的误识别率和删除困难 .
- 总结：布隆过滤器只是放在外面来当一个缓存使用的，即，当一个很快速的判断使用.当被检索元素在布隆过滤器中被查到后，会继续在这台机器上的数据库中去查，如果没有查找到的话，说明元素不存在于布隆过滤器中，就会将被检索元素插入到布隆过滤器中；当被检索元素，没有在布隆过滤器中被查到后，直接将被检索元素插入布隆过滤器中。

布隆过滤器示意图



x (1,1,1) ➜ x 存在于 布隆过滤器中,

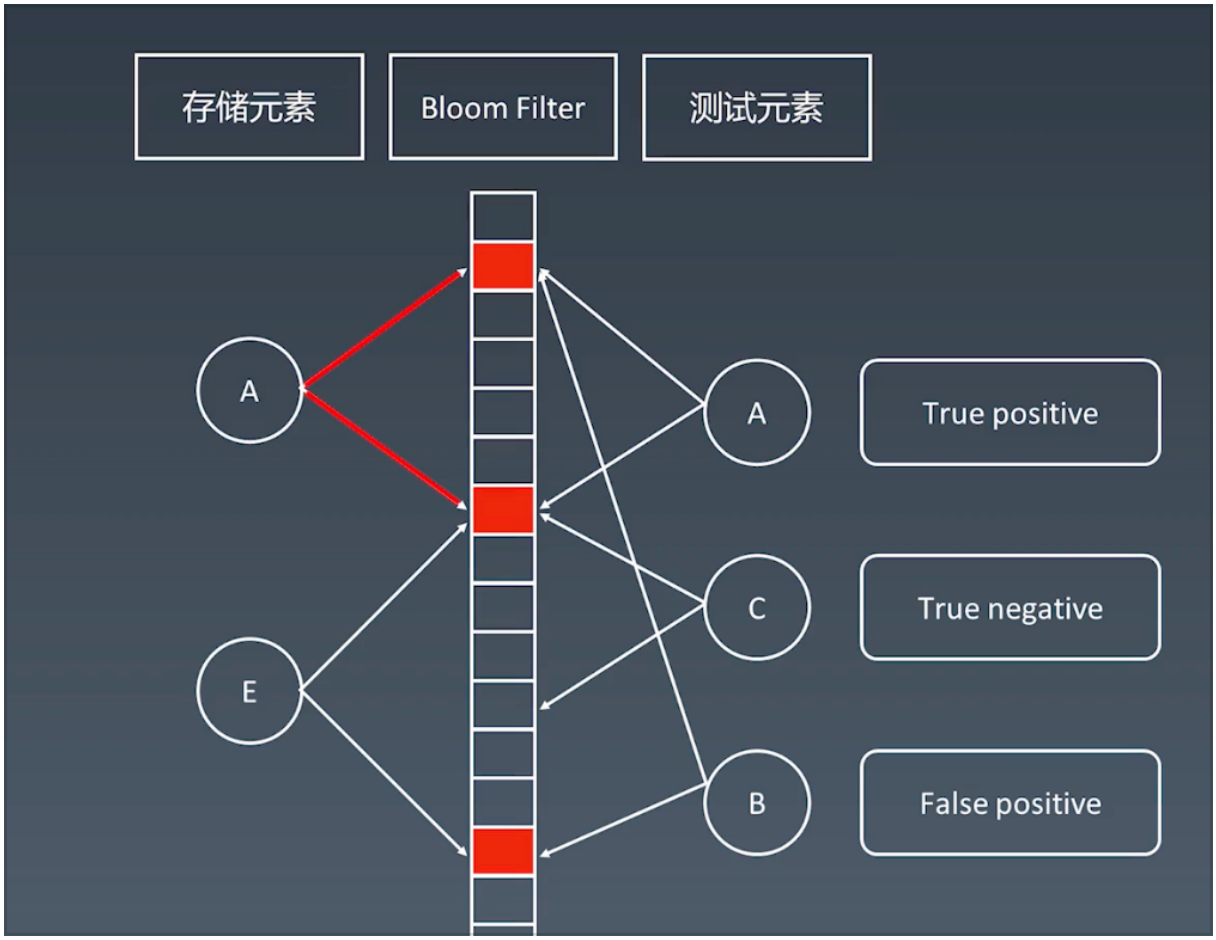
y (1,1,1) ➜ y 存在于 布隆过滤器中,

w (1,1,0) ➜ w 不存在于 布隆过滤器中.

列子

C 测试元素，二进制位不全为1，不在布隆过滤器中；

B 测试元素，虽然二进制位全为1，但是不在数据库（存储元素中），所以也不在布隆过滤器中.



案例

1. 比特币网络
2. 分布式系统（Map - Reduce）– Hadoop、Search Engine
3. Redis Cache
4. 垃圾邮件、品论等的过滤
5. 集合判重

使用布隆过滤器解决缓存击穿、垃圾邮件识别、集合判重

代码实现

Python

布隆过滤器 Python 实现例子 1

布隆过滤器 Python 实现例子 2

高性能布隆过滤器 Python 实现例子

Java

布隆过滤器 Java 实现例子 1

布隆过滤器 Java 实现例子 2

C/C++

#Algorithm/Part II : Theory/Data Structure#