# 字节换算

1字节=8位(1 byte = 8 bit)

bit只有0和1两种状态

# 布隆过滤器

### 使用场景 (黑名单问题)

黑名单功能哈希表占用空间过大,可以用布隆过滤器

会有无辜命中情况(数据不匹配时可能误认为匹配)

java里一个int四个字节,一个字节8bit

所有一个int变量可以表示32bit

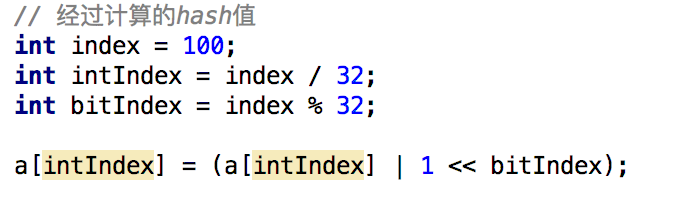
一个长度为100的int数组 可以表示(32\*100)bit

## 设计步骤

### 初始化黑名单数据

黑名单数据经过哈希计算将底层bit位数据修改,为了保证减少误差

需要同时准备多个哈希函数 同时修改对应bit位数据



### 过滤功能使用

一个URL经过N个哈希函数

如果是黑名单数据,对应bit位必然都为1

如果不是黑名单,对应bit位不为1 但是对应bit位可能会受到其他数据的哈希修改影响

即会产生误差

### 如何减少误差

底层数组长度越长,碰撞几率越小

哈希函数增多,合理设计,只要对应bit位有一位不为1,则URL不在黑名单中

# 一致性哈希

参考: <https://blog.csdn.net/bntX2jSQfEHy7/article/details/79549368>

# 哈希冲突