$$\Rightarrow e^{-\frac{kn}{m}} = \frac{1}{2}$$

$$\Rightarrow \frac{kn}{m} = \ln 2$$

$$\Rightarrow k = \ln 2 \times \frac{m}{n} = 0.7 \times \frac{m}{n}$$

至此，我们得到了如何根据 m 与 n 的值得到最合适的哈希函数数量 k 的公式，把这个公式带回失误率公式，就得到了如何根据失误率 p 和样本数 n 来确定布隆过滤器大小 m 的公式。

布隆过滤器会有误报，对已经发现的误报样本可以通过建立白名单来防止误报。比如，已经发现“aaaaaa5”这个样本不在布隆过滤器中，但是每次计算后的结果都显示其在布隆过滤器中，那么就可以把这个样本加入到白名单中，以后就可以知道这个样本确实不在布隆过滤器中。

在此特别感谢本篇文章参考网文的作者 Allen Sun（http://www.cnblogs.com/allensun/archive/2011/02/16/1956532.html）。

只用 2GB 内存在 20 亿个整数中找到出现次数最多的数

【题目】

有一个包含 20 亿个全是 32 位整数的大文件，在其中找到出现次数最多的数。

【要求】

内存限制为 2GB。

【难度】

士 ★☆☆☆

【解答】

想要在很多整数中找到出现次数最多的数，通常的做法是使用哈希表对出现的每一个数做词频统计，哈希表的 key 是某一个整数，value 是这个数出现的次数。就本题来说，一共有 20 亿个数，哪怕只是一个数出现了 20 亿次，用 32 位的整数也可以表示其出现的次数

而不会产生溢出，所以哈希表的 key 需要占用 4B，value 也是 4B。那么哈希表的一条记录(key,value)需要占用 8B，当哈希表记录数为 2 亿个时，需要至少 1.6GB 的内存。

但如果 20 亿个数中不同的数超过 2 亿种，最极端的情况是 20 亿个数都不同，那么在哈希表中可能需要产生 20 亿条记录，这样内存会不够用，所以一次性用哈希表统计 20 亿个数的办法是有很大风险的。

解决办法是把包含 20 亿个数的大文件用哈希函数分成 16 个小文件，根据哈希函数的性质，同一种数不可能被哈希到不同的小文件上，同时每个小文件中不同的数一定不会大于 2 亿种，假设哈希函数足够好。然后对每一个小文件用哈希表来统计其中每种数出现的次数，这样我们就得到了 16 个小文件中各自出现次数最多的数，还有各自的次数统计。接下来只要选出这 16 个小文件各自的第一名中谁出现的次数最多即可。

把一个大的集合通过哈希函数分配到多台机器中，或者分配到多个文件里，这种技巧是处理大数据面试题时最常用的技巧之一。但是到底分配到多少台机器、分配到多少文件，在解题时一定要确定下来。可能是在与面试官沟通的过程中由面试官指定，也可能是根据具体的限制来确定，比如本题确定分成 16 个文件，就是根据内存限制 2GB 的条件来确定的。

40 亿个非负整数中找到没出现的数

【题目】

32 位无符号整数的范围是 0～4294967295，现在有一个正好包含 40 亿个无符号整数的文件，所以在整个范围中必然有没出现过的数。可以使用最多 1GB 的内存，怎么找到所有没出现过的数？

进阶：内存限制为 10MB，但是只用找到一个没出现过的数即可。

【难度】

尉 ★★☆☆

【解答】

原问题。如果用哈希表来保存出现过的数，那么如果 40 亿个数都不同，则哈希表的记录数为 40 亿条，存一个 32 位整数需要 4B，所以最差情况下需要 40 亿×4B=160 亿字节，大约需要 16GB 的空间，这是不符合要求的。