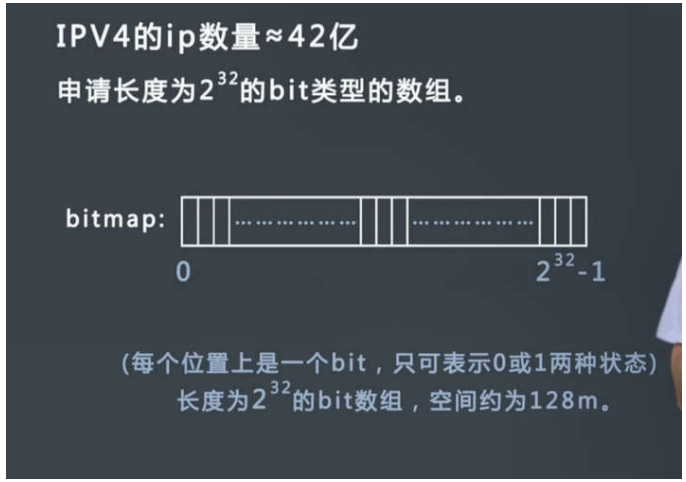
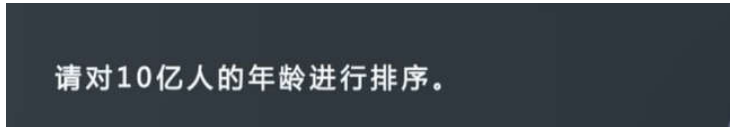
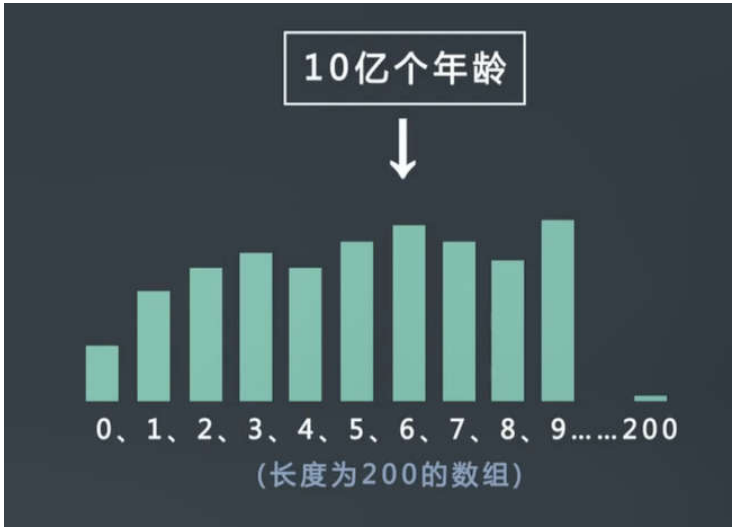
大数处理


常见海量处理题目解题关键
1、分而治之。通过哈希函数将大任务分流到机器，或分流成小文件。
2、常用的hashMap或bitmap。
难点：通讯、时间和空间的估算。

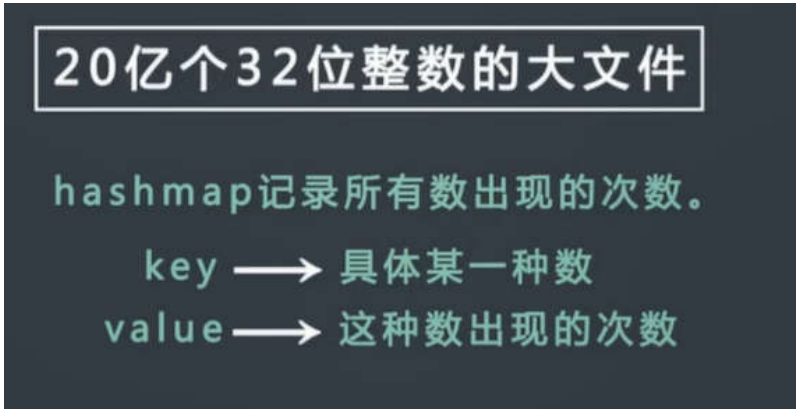

请对10亿个IPV4的ip地址进行排序，每个ip只会出现一次。

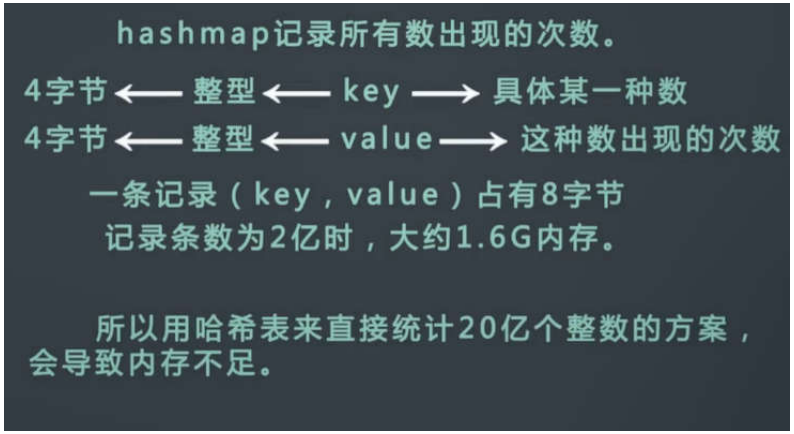

IPV4的ip数量≈42亿
申请长度为2^{32}的bit类型的数组。
bitmap:
0
2^{32}-1
(每个位置上是一个bit，只可表示0或1两种状态)
长度为2^{32}的bit数组，空间约为128m。

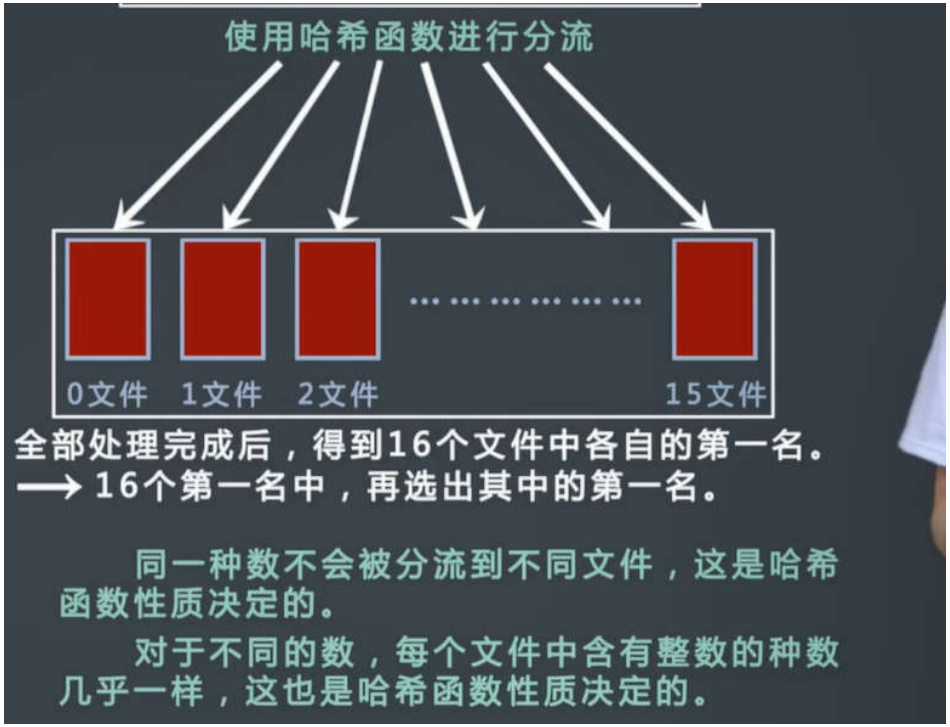

请对10亿人的年龄进行排序。

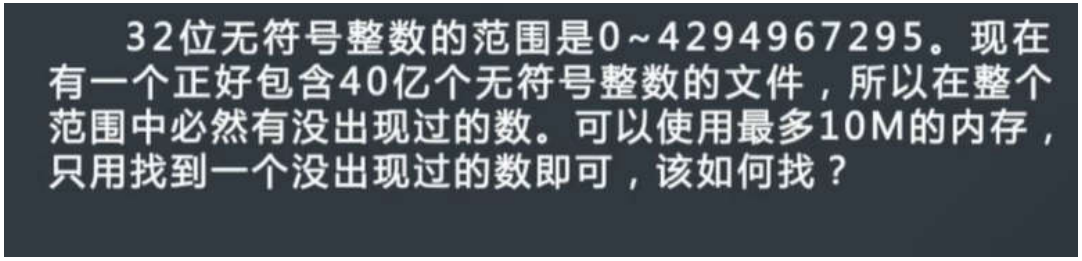


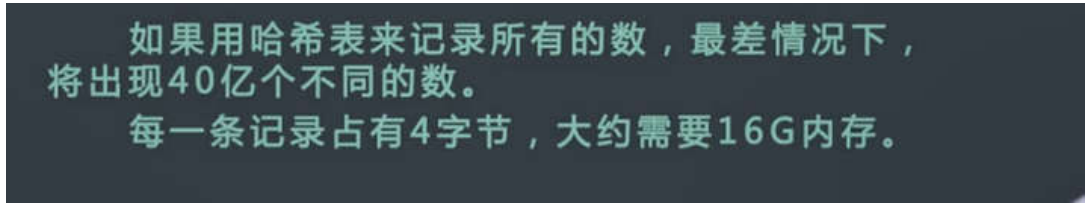
有一个包含20亿个全是32位整数的大文件，在其中找到出现次数最多的数。但是内存限制只有2G。


20亿个32位整数的大文件
hashmap记录所有数出现的次数。
key ⟶ 具体某一种数
value ⟶ 这种数出现的次数

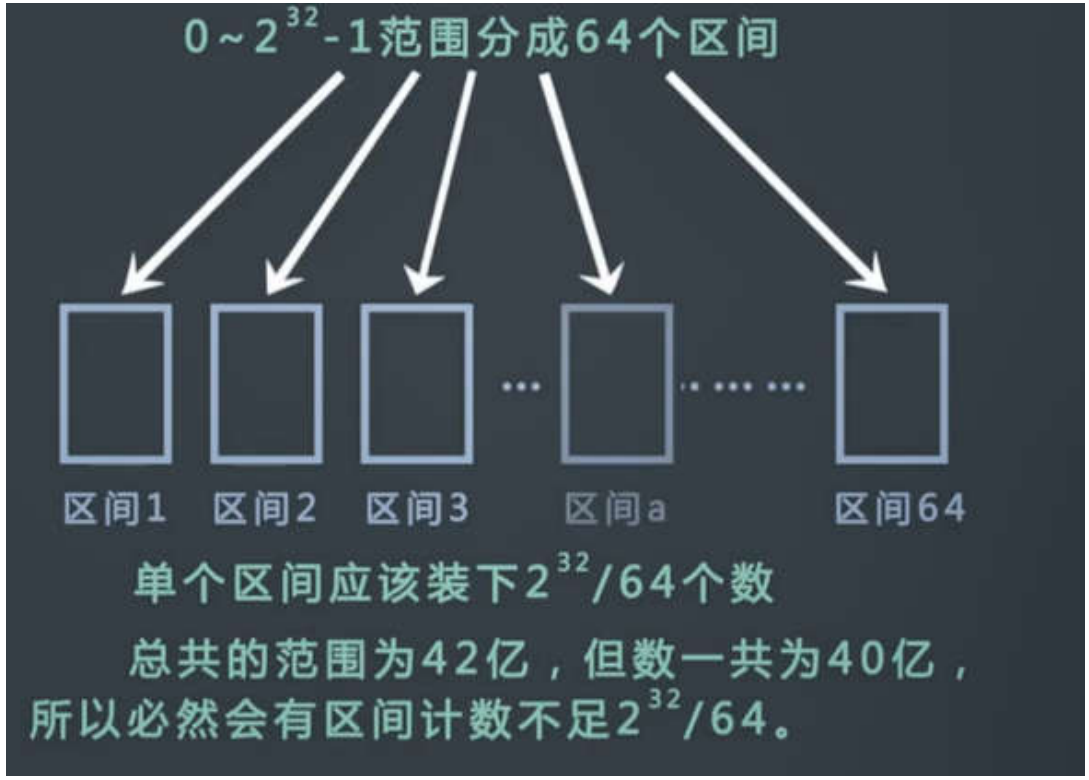

hashmap记录所有数出现的次数。
4字节 ⟵ 整型 ⟵ key ⟶ 具体某一种数
4字节 ⟵ 整型 ⟵ value ⟶ 这种数出现的次数
一条记录（key，value）占有8字节
记录条数为2亿时，大约1.6G内存。
所以用哈希表来直接统计20亿个整数的方案，
会导致内存不足。

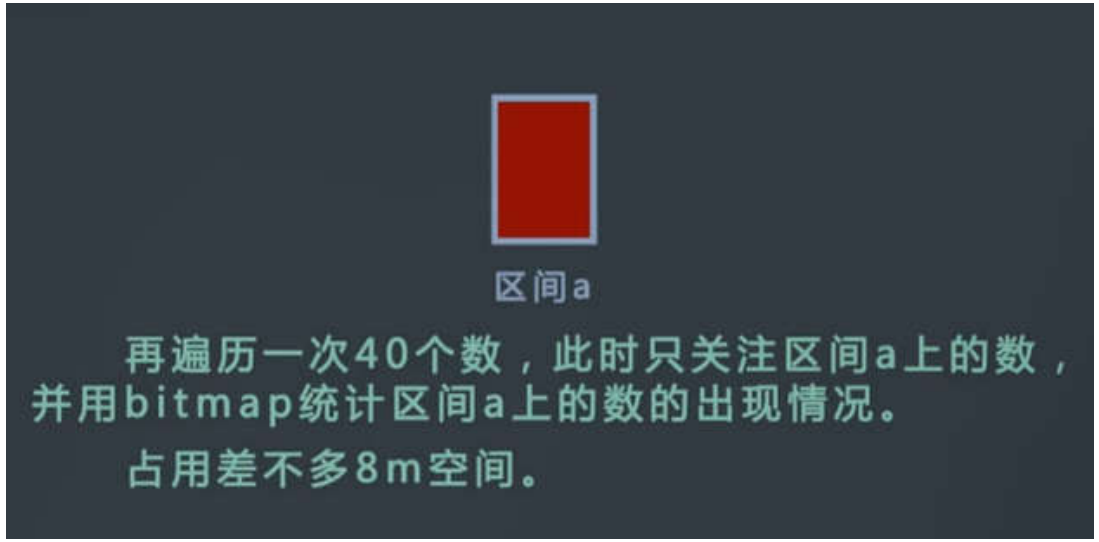

使用哈希函数进行分流
0文件
1文件
2文件
15文件
全部处理完成后，得到16个文件中各自的第一名。
⟶ 16个第一名中，再选出其中的第一名。
同一种数不会被分流到不同文件，这是哈希函数性质决定的。
对于不同的数，每个文件中含有整数的种数几乎一样，这也是哈希函数性质决定的。

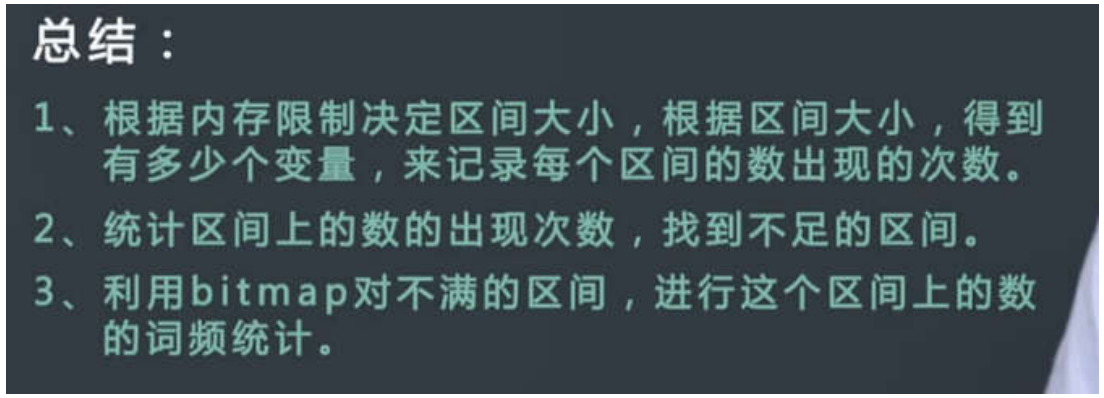

32位无符号整数的范围是0~4294967295。现在有一个正好包含40亿个无符号整数的文件，所以在整个范围中必然有没出现过的数。可以使用最多10M的内存，只用找到一个没出现过的数即可，该如何找？


如果用哈希表来记录所有的数，最差情况下，
将出现40亿个不同的数。
每一条记录占有4字节，大约需要16G内存。

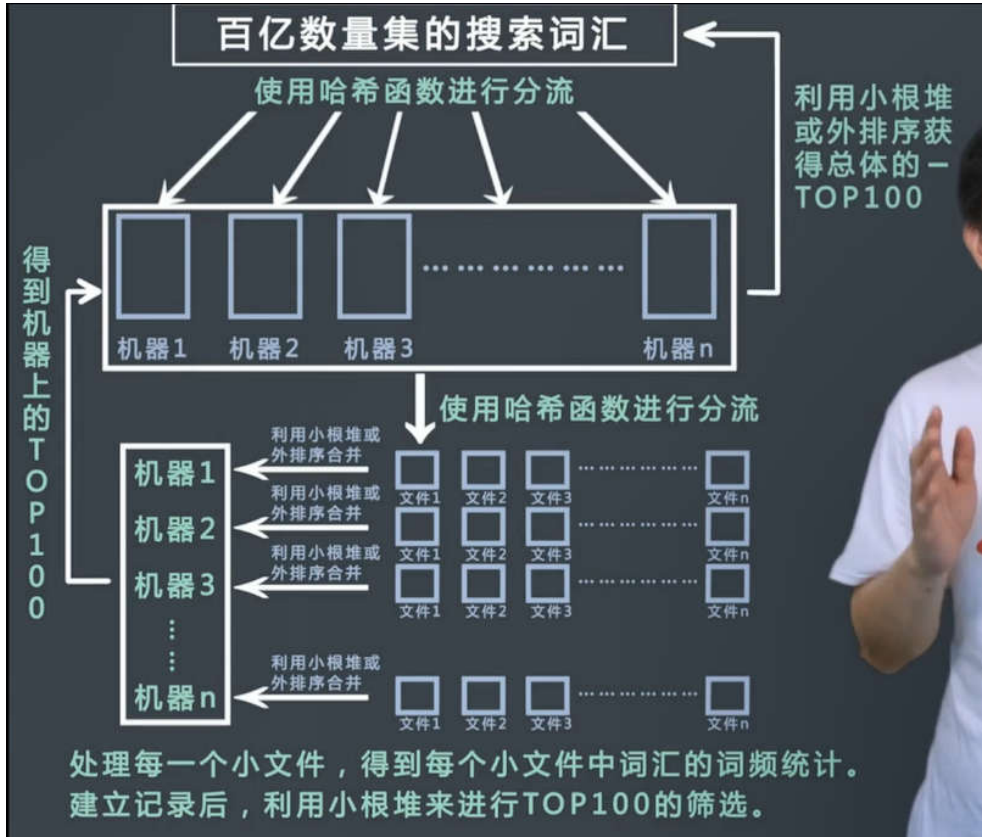

bitmap:
0 1 2……$2^{32}-1$
每个位置为1个bit，只能表示0或1两种状态。
大约占用500m空间。

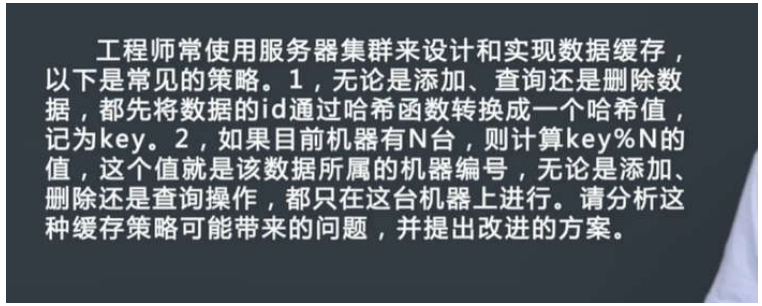

0~2^32-1范围分成64个区间
区间1
区间2
区间3
区间a
区间64
单个区间应该装下2^32/64个数
总共的范围为42亿，但数一共为40亿，
所以必然会有区间计数不足2^32/64。

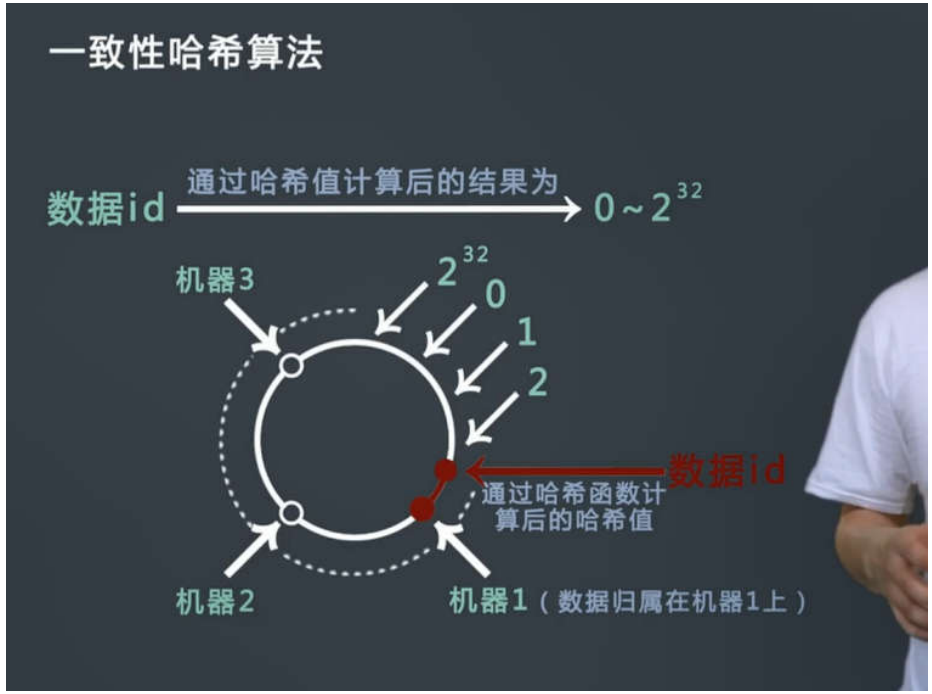

区间a
再遍历一次40个数，此时只关注区间a上的数，
并用bitmap统计区间a上的数的出现情况。
占用差不多8m空间。

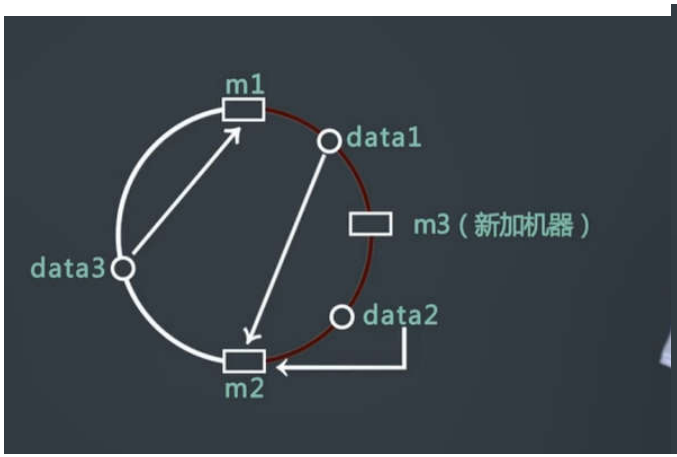

总结：
1、根据内存限制决定区间大小，根据区间大小，得到有多少个变量，来记录每个区间的数出现的次数。
2、统计区间上的数的出现次数，找到不足的区间。
3、利用bitmap对不满的区间，进行这个区间上的数的词频统计。

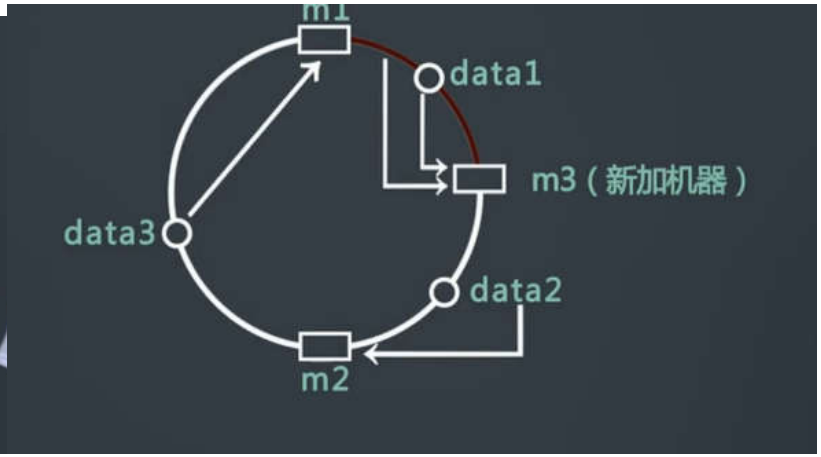
某搜索公司一天的用户搜索词汇是海量的，假设有百亿的数据量，请设计一种求出每天最热100词的可行办法。




工程师常使用服务器集群来设计和实现数据缓存，以下是常见的策略。1，无论是添加、查询还是删除数据，都先将数据的id通过哈希函数转换成一个哈希值，记为key。2，如果目前机器有N台，则计算key%N的值，这个值就是该数据所属的机器编号，无论是添加、删除还是查询操作，都只在这台机器上进行。请分析这种缓存策略可能带来的问题，并提出改进的方案。


一致性哈希算法
数据id
通过哈希值计算后的结果为
0~2^32
机器3
2^32
0
1
2
数据id
通过哈希函数计算后的哈希值
机器2
机器1（数据归属在机器1上）


m1
data1
m3（新加机器）
data3
data2
m2


m1
data1
m3（新加机器）
data3
data2
m2