1. 例题：

**1、海量日志数据，提取出某日访问百度次数最多的那个IP。**

考虑IPV4地址数量是有限的，全球最多2^32个IP地址，先使用hash将IP地址直接存入内存，写入一个大文件。

采用映射的方式，模1000，将大文件映射为1000个小文件，找出每个小文件中出现频率最大的IP（用hash\_map），然后在1000个最大的IP中找到频率最大的IP。

**2、假设目前有一千万个记录（这些查询串的重复度比较高，虽然总数是1千万，但如果除去重复后，不超过3百万个。一个查询串的重复度越高，说明查询它的用户越多，也就是越热门。），请你统计最热门的10个查询串，要求使用的内存不能超过1G。**

先使用HashTable将它们存入内存并自动排序，算法复杂度为O（N）。

HashTable类型为<String , Integer> (<Query名称，出现次数>)。

用小根堆进行堆排序，将每次的Integer与堆顶进行比较，若大，放入堆顶，进行堆排序。

用该堆排序的时间复杂度为：N\*O（logK）

**3、有一个1G大小的一个文件，里面每一行是一个词，词的大小不超过16字节，内存限制大小是1M。返回频数最高的100个词。**

对于每个词，扫描时hash(x)%5000, 这样每个文件大小大约为200KB左右

对于每个文件，找出出现频率最大的100个词，用HashTable存入内存自动排序

把100个词和频率存入文件，再把这5000个文件进行归并。

**4、给定a、b两个文件，各存放50亿个url，每个url各占64字节，内存限制是4G，让你找出a、b文件共同的url？**

一共是50亿乘以64B = 320G，所以肯定不能存在内存里。

遍历a文件和b文件，分别hash(x)%1000，存入1000个小文件中，每个小文件大概300M。

遍历文件b并%1000，同样得到1000个小文件。

这样只用求出1000对小文件中相同的url就行了。因为hash操作是一致的。

**5、在2.5亿个整数中找出不重复的整数，注，内存不足以容纳这2.5亿个整数。**

采用2-Bitmap（每个数分配2bit，00表示不存在，01表示出现一次，10表示多次，11无意义）进行，共需内存内存，还可以接受。然后扫描这2.5亿个整数，查看Bitmap中相对应位，如果是00变01，01变10，10保持不变。所描完事后，查看bitmap，把对应位是01的整数输出即可。

**6、腾讯面试题：给40亿个不重复的unsigned int的整数，没排过序的，然后再给一个数，如何快速判断这个数是否在那40亿个数当中？**

同题5，用Bit数组来做。

**2** 总结：

1. Hashing：

经常用于分治，写入大文件。

1. bit-map：

经常用与查找是否含有某个数，节省（内存）空间

1. 堆：

海量数据排序，堆比较小，可以放入内存中