1、 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，找出 a、b 文件共同的 URL？

（1）1G=1073741824b，1Kb=1024B 可以估计每个文件的大小为 5000000000×64/1073741824=298G,估计为 320G，大于内存限制的 4G，所以不可能将其完全加载到内存中做处理，采用分治法。先遍历文件 a，对每个 URL 取值 hash(URL)%1000，然后根据取得的值，将 URL 分别存储到 1000 个小文件，这样每个小文件的大小约为 300M。再遍历文件 b，采取和 a 相同的方式对文件进行处理，这样处理之后，所有可能相同的文件都会在同一个小文件中，不同的小文件不可能有相同的 URL，然后只需要求 1000 对小文件中相同的 URL 即可。最后，求每对小文件中相同的 URL，可以把其中一个小文件的 URL 存储到 hash set 中，然后遍历另一个小文件的 URL，看是否在刚才的 hash set 中，如果存在，则说明是相同的 URL。

（2）如果允许一定的错误率，可使用布隆过滤器，4G 内存大概可以表示 340 亿 bit。4*1024*1024*1024*8=34359738368，将其中一个文件中的 URL 使用布隆过滤器映射为 340 亿 bit，然后依次读取另一个文件的 URL，检查是否与布隆过滤器，如果是，则 URL 应该是相同的 URL。

（3）hash 后要判断每个文件的大小，如果 hash 分配不均，存在文件较大，应继续 hash 文件，换 hash 算法再次 hash，一直到没有较大的文件为止。如果第一次 hash 有较大文件存在，不能直接用 set。对每个文件用字符串自然顺序排序，具有相同 hash 编号的，可以从头到尾比较一遍，对于不是一次 hash 的，hash 次数少的要和 hash 次数多的每个文件都比较一次，才能确定是相同的 URL。

2、 有 10 个文件，每个文件 1G，每个文件的每一行存放的都是用户的 query，每个文件的 query 都可能重复。按照 query 的频度排序？

（1）顺序读取 10 个文件，按照 hash(query)%10 的结果将 query 写入到另外 10 个文件中，新生成的文件每个的大小大约 1G，假设 hash 函数随机。找一台内存 2G 左右的机器，依次对 10 个文件进行 hashmap(query,query_count)来统计 query 出现的次数，利用快速排序、堆排序或者归并排序按照出现次数进行排序。将排好序的 query 和对应的 query_count 输出到文件中，这样得到了 10 个排好序的文件。对 10 个文件进行归并排序，采用内排序与外排序结合。

（2）一般 query 总量有限，只是重复次数比较多，如果所有的 query 能够一次性加入到内存中，采用前缀树或者 hashmap 直接来统计每个 query 出现的次数，按出现次数进行快速排序、堆排序或者归并排序。

（3）在 hash 之后，分为多个文件，交给多个文件进行处理，采用分布式的架构处理，如 MapReduce，再进行合并。

3、 有一个 1G 大小的文件，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。

（1）顺序读文件中，对于每个词 X，取 hash(X)%5000，然后按照该值存到 5000 个小文件中。每个文件大概 200K，如果已知有的文件大小超过了 1M，则继续往下分，直到分解到小文件的大小都不超过 1M。对于每个小文件，统计每个小文件中出现的词以及频率，可采用前缀树或者 hashmap，并获取出现频率最大的 100 个词，用含有 100 个节点的最小堆，并把 100 个词以及相应的频率存入文件，又得到了 5000 个文件，然后进行归并排序。

4、 海量日志数据，提取出某日访问百度次数最多的 IP。

（1）将这一天访问百度的 IP 取出来，写入到一个大文件中。IP 是 32 位的，最多有 2^32 个 IP。采用映射的方法，模 1000 将整个大文件映射为 1000 个小文件，再找出每个小文件中出现频率最大的 IP，可以采用 hashmap 进行频率的统计，然后找出频率最大的几个，然后在这 1000 个最大的 IP 中，找出频率最大的一个 IP。

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

（1）采用 2-Bitmap 进行 ，每一个数分配 2bit，00 表示不存在，01 表示存在一次，10 表示存在多次，11 没有意义，共需要内存 2^32*2bit=1GB 内存。然后扫描 2.5 亿个数，查看 Bitmap 中对应位，如果 00 变 01,01 变 10,10 保持不变。扫描完之后查看 Bitmap，把对应是 01 的整数输出。

（2）采用类似方法进行小文件的划分，然后在小文件中找出不重复的整数，并且排序。然后再进行归并，去除重复的元素。

6、 海量数据分布在 100 台电脑中，高效统计这批数据的 TOP10

（1）每台电脑上求出 TOP10，采用包含了 10 个元素的堆完成（TOP10 小 用最大堆，TOP10 大用最小堆）。如果求 TOP10 大，首先取 10 个元素调整为最小堆，然后扫描后面的数据，并和堆顶的元素进行对比，如果后面的数据比堆顶的元素大，用该元素替换堆顶，然后再调整为最小堆。最后堆中的元素就是 TOP10 大。求出了每台电脑上的 TOP10 后，把 100 台电脑上的 TOP10 组合起来，共 1000 个数据，再利用类似的方法求出 TOP10。

7、 在海量数据中找出重复次数最多的一个？

（1）先做 hash，然后求模映射为小文件，求出每个小文件中重复次数最多的一个，并且记录重复次数。然后找出上一步求出的数据中重复次数最多的一个。

8、 上千万或者上亿有重复的数据，统计其中出现次数最多的前 N 数据？

（1）在机器内存能够存下时，采用 hashmap 或者搜索二叉树或者红黑树等来进行统计次数。然后取出前 N 出现次数最多分数据。

9、 1000 万字符串，其中有些是重复的，需要把重复的去掉，保留没有重复的字符串，如何设计和实现？

（1）用前缀树或者 hashmap 来实现。

10、 一个文本文件，大约有 10000 行，每行一个词，要求统计出其中最频繁出现的前 10 个词？

（1）用前缀树统计每个词出现的次数，时间复杂度是 O(n*le)(le 表示单词的平均长度)，然后找出出现最频繁的前 10 个词，用堆来实现，时间复杂度是 O(n*lg10)，总的时间复杂度是两个中较大的。

11、 一个文本文件，找出前 10 个经常出现的词，文件比较长，上亿行，无法一次读入内存，求最优解。

（1）根据 hash 进行求模，将文件分解为多个小文件，对于单个文件利用上题的方法求出每个文件汇总 10 个最长出现的词，然后进行归并处理，找出最终的 10 个最常出现的词。

12、 100W 个数，找出最大的 100 个数 。

（1）采用局部淘汰法，选取前 100 个元素，并且排序，记为序列 L。然后依次扫描剩余的元素 X，与排好序的 100 个元素中最下的元素比，如果比这个元素大，则把小的元素删除，并且利用插入排序的思想将新的数插入到序列中。依次循环，直到扫描了所有的元素。复杂度为 O(100W*100)。

（2）采用快速排序的思想，每次分割之后只考虑比轴大的部分。直到比轴大的部分多于 100 个数，采用传统排序算法排序，取前 100 个。复杂度为 O(100W*100)。

（3）用一个含有 100 个数的最小堆完成，复杂度为 O(100W*100)。

13、 寻找热门查询。

搜索引擎会通过日志文件把用户每次检索使用的所有检索串都记录下来，每个查询串的长度为 1-255 字节，假设目前有 1000 万个记录，这些查询串的重复读比较高，虽然总数是 1000 万，但是如果去除重复和，不超过 300 万个。一个查询串的重复度越高，说明查询的用户越多，也越热门。统计最热门的十个查询串，要求使用的内存不能超过 1G。

采用 trie 数，关键字存该查询串出现的次数，没有出现为 0。最后用 10 个元素的最小堆来对出现频率进行排序。

14、 一共 N 台机器，每个机器上有 N 个数。每个机器最多存 O(N)个数并对它们操作，如何找到 N^2 个数的中位数？

（1）先大体估计数的范围，比如假设数时 32 位无符号整数(2^32 个)。把 0 到 2^32-1 的整数划分为 N 个范围段，每个段包含(2^32)/N 个整数。然后扫描每个机器上的 N 个数，把属于第一个区段的数放在第一个机器上，属于第二个区段的数放在第二个机器上，属于第 N 个区段的数放到第 N 个机器上。每个机器上存储的数应该是 O(N)的。然后依次统计每个机器上数的个数，依次累加，直到找到第 K 个机器，在

该机器上累加的数大于或者等于(N^2)/2，而在第 K-1 个机器上的累加数小于(N^2)/2，并且把这个数记为 x，则要找的机器在第 K 个机器中，排在第(N^2)/2-x 位。然后对第 K 个机器上的数排序，找出第(N^2)/2-x 个数，即要求的数。复杂度为 O(N^2)。

（2）先对每台机器上的数进行排序，排好序后，采用归并排序的思想，将这 N 个机器上的数归并得到最终的排序，然后找第(N^2)/2 个。复杂度 O(N^2*lgN^2)。

15、 最大间隙问题。给定 N 个实数，求这 N 个实数在实轴上相邻 2 个数之间的最大差值，求线性的时间算法。

（1）先对 N 个数进行排序，然后一遍扫描确定最大间隙。但不能满足线性。因此找到 N 个数据中最大和最小的数据。用 N-2 个带你等分区间[min , max]，分为 n-1 个前闭后开的区间，将这些区间看成是桶。且桶 i 的上界和桶 i+1 的下界相同，桶的边界则构成了一个等差数列，将 min 放入第一个桶，将 max 放入第 n-1 个桶。将 N 个数放入 n-1 个桶，将每个元素分配到某个桶，并且求出桶的最大和最小数据。除了最大最小数据之外的 N-2 个数据放在了 n-1 个桶中。抽屉原理可知最少一个桶是空的，每个桶的大小相同，最大间隙不会在同一个桶中，一定在桶的上界和某个桶的下界，且两个桶之间的桶一定是空桶，最大间隙在桶 i 的上界和桶 j 的下界之间产生，j>=i+1.一趟扫描完成。

16、 将多个集合合并成没有交集的集合。

给定一个字符串的集合，要求将其中交集不为空的集合合并，合并完成的集合之间没有交集。

采用并查集。首先所有的字符串都在单独的并查集中。然后依次扫描每个集合，顺序合并将两个相邻元素合并。复杂度 O(NlgN)。

17、 最大子序列和最大子矩阵问题。

数组的最大子序列问题，给定一个有负数和正数的数组，找出一个连续子序列，使得和最大。

这个问题可以使用动态规划解决，b[i]表示以第 i 个元素 a[i]结尾的最大子序列 b[i+1]=b[i]>0?b[i]+a[i+1]:a[i+1]。