解决问题：1.局部有序整体无序的小文件到大文件的排序

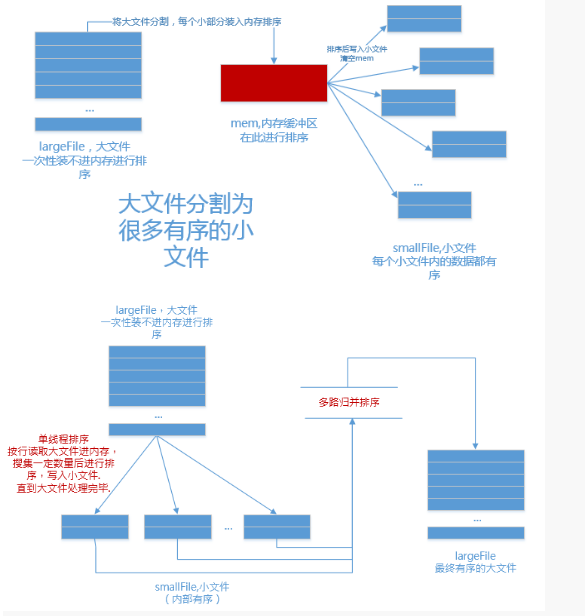
2.整体有序，局部无序的排序

外部排序：

内部排序就是把数据全部倒入内存中，然后利用（快排，插入排序之类的算法）等排序算法进行排序。那么当数据很大，一下子不能全部装进内存，怎么办，这时候就可以用外部排序。

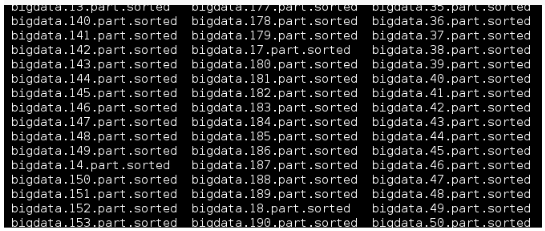
外部排序

内存极少的情况下，利用分治策略，利用外存保存中间结果，再用多路归并来排序;



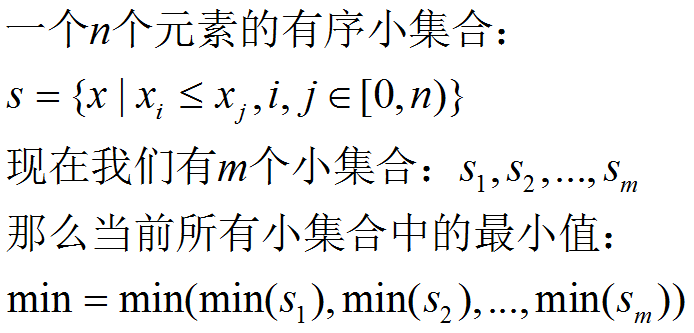
### 1.分

内存中维护一个极小的核心缓冲区memBuffer，将大文件bigdata按行读入，搜集到memBuffer满或者大文件读完时，对memBuffer中的数据调用内排进行排序，排序后将有序结果写入磁盘文件bigdata.xxx.part.sorted.  
循环利用memBuffer直到大文件处理完毕，得到n个有序的磁盘文件：



### 2.合

现在有了n个有序的小文件，怎么合并成1个有序的大文件？  
把所有小文件读入内存，然后内排？  
no!

利用如下原理进行归并排序：  
  
我们举个简单的例子：

我们举个简单的例子：

文件1：3,6,9  
文件2：2,4,8  
文件3：1,5,7

第一回合：  
文件1的最小值：3 , 排在文件1的第1行  
文件2的最小值：2，排在文件2的第1行  
文件3的最小值：1，排在文件3的第1行  
那么，这3个文件中的最小值是：min(1,2,3) = 1  
也就是说，最终大文件的当前最小值，是文件1、2、3的当前最小值的最小值，绕么？  
上面拿出了最小值1，写入大文件.

第二回合：  
文件1的最小值：3 , 排在文件1的第1行  
文件2的最小值：2，排在文件2的第1行  
文件3的最小值：5，排在文件3的第2行  
那么，这3个文件中的最小值是：min(5,2,3) = 2  
将2写入大文件.

也就是说，最小值属于哪个文件，那么就从哪个文件当中取下一行数据.（因为小文件内部有序，下一行数据代表了它当前的最小值）

****Pro2：20路已经有序+20路合并求Top500****

有 20 个数组，每个数组有 500 个元素，并且是有序排列好的，现在在这 20\*500个数中找出排名前 500 的数。

答：从 20 个数组中各取一个数，并记录每个数的来源数组， 建立一个含 20 个元素的大根堆 。此时堆顶就是最大的数，取出堆顶元素，并 ****从堆顶元素的来源数组中取下一个数加入堆**** ，再取最大值，一直这样 进行 500 次即可 。

****Pro3:** **最小堆** **K** **路合并****

请给出一个时间为O(nlgk)、用来 将 k 个已排序链表合并为一个排序链表 的算法，此处n为所有输入链表中元素的总数。

算法思想:

1. 从k个链表中取出每个链表的第一个元素， 组成一个大小为 k 的数组 arr ，然后将数组 arr 转换为最小堆，那么 arr[0] 就为最小元素了；

2. 取出 arr[0] ，将其放到新的链表中，然后将 arr[0] 元素在原链表中的下一个元素补到 arr[0] 处，即 arr[0].next ，如果 arr[0].next为空， 即它所在的链表的元素已经取完了，那么将堆的最后一个元素补到 arr[0] 处， 堆的大小自动减 1 ， 循环即可。

****Pro4: 整体有序局部无序问题****

一个有100亿个元素的整型数组，它的 元素是有序的 ，现在把它分成若干段，每段不超过20个元素，**每段的元素个数不等**，现在在 **每段内将这些元素的顺序打乱** ，然后重新 将这100亿个元素的数组排序，请问时间复杂度最小的算法是什么？并给出时间复杂度。

分析：如果每段长度相等，则可以考虑采用上面的K路归并，但此处长度不相等，需另行考虑其它方法。

解：（直接插入排序）

观察这个结果，假设第1到第5n个数已经有序为sort(5n)，那么我们要将5n+1到5n+5这5个数据添加到已排序的数组中，只需要进行插入排序，将这5个数添加进即可。由于分段的长度不超过5，所以第5n+1个数在插入的时候，最多只需要搜索到第5n-4个数就可以了，比较个数不会超过5次。又因为5n+1到5n+5是已经排好序的，所以，后面的数比较次数也不会超过5次（最多比较到前一个插入的位置）。因此， 每加入 5 个数到已排序数组中，时间复杂度是 O （ 5\*5 ）

假设长度为N，每段长不超过K。则每段插入的时间复杂度即为 O （ K\*K ）。

而对于以段为单位插入的操作，需要进行N/K次，所以，总的时间复杂度是 O(K\*K)\*O(N/K)=O(NK)