8.7 外部排序（上）

- **外部排序的基本概念**
  - 对大文件进行排序,因为文件中的记录很多、信息量庞大,无法将整个文件复制进内存中进行排序
  - 需要将待排序的记录存储在外存上,排序时再把数据一部分一部分地调入内存进行排序,在排序过程中需要多次进行内存和外存之间的交换
- **外部排序的方法**
  - 基本概念
    - 文件通常是按块存储在磁盘上的,操作系统也是按块对磁盘上的信息进行读写的
    - 外部排序过程中的时间代价主要考虑访问磁盘的次数,即 I/O次数
  - 外部排序通常采用归并排序法　　算法实现的两个阶段
    - 据内存缓冲区大小,将外存上的文件分成若干长度为f的子文件,依次读入内存并利用内部排序方法对它们进行排序, 并将排序后得到的有序子文件重新写回外存（归并段或顺串）
    - 对这些归并段进行逐趟归并,使归并段（有序子文件）逐渐由小到大,直至得到整个有序文件为止
  - 耗费时间　　外部排序的总时间= 内部排序所需的时间 + 外存信息读写的时间 + 内部归并所需的时间
  - 归并排序优化
    - 增大归并路数k
    - 减少初始归并段个数r

    都能减少归并趟数s,进而减少读写磁盘的次数,达到提高外部排序速度的目的
- **多路平衡归并与败者树**
  - 引入败者树的背景　　为了使内部归并不受k（归并路数）的增大的影响
  - 基本思想
    - 败者树是树形选择排序的一种变体, 可视为一棵完全二叉树
    - k个叶结点分别存放k个归并段在归并过程中当前参加比较的记录,内部结点用来记忆左右子树中的"失败者",而让胜者往上继续进行比较,一直到根结点
    - 若比较两个数,大的为失败者、小的为胜利者,则根结点指向的数为最小数
  - 性能分析
    - k 路归并的败者树深度　　$\lceil \log_2 k \rceil$
    - 总的比较次数　　$S(n-1)\lceil \log_2 k \rceil = \lceil \log_k r \rceil (n-1)\lceil \log_2 k \rceil = (n-1)\lceil \log_2 r \rceil$
  - 注意
    - 归并路数k并不是越大越好。归并路数k增大时,相应地需要增加输入缓冲区的个数
    - 当k值过大时,虽然归并趟数会减少,但读写外存的次数仍会增加
  - 优化
    - 增加归并路数 k，进行多路平衡归并
      - 代价1：需要增加相应的输入缓冲区
      - 代价2：每次从k个归并段中选一个最小元素需要（k-1）次关键字对比
    - 减少初始归并段数量r