第九章 外排序

1. FIFO(First in First out)方法和 LRU(Least recently used)方法是用来作为缓冲区替换算法的两种选择。FIFO 每次将最先进入缓冲区的结果替换出去；LRU 每次将距离上次使用时间最长的结果替换出去。假定我们有一个大小为 3 的缓冲区，和一个如下输入的输入序列：
2 3 1 4 0 3 2 4 1 2 4 3
(1) 请分别写出 FIFO 方法和 LRU 方法的运行结果（缓冲区的变化），并比较这两个算法，给出 FIFO 方法和 LRU 方法的主要区别。
(2) 想要减少访问内存的 miss rate，一个直观的想法是增大缓冲区的大小。请问这样做对于 LRU 和 FIFO 方法是否都是有效的？如果不是请举出反例。
(3) 描述一种你所知道的其他替换算法，或者你所设计的一种替换算法。并说明这个算法可能的优势和劣势。

答：
(1) FIFO：2、23、231、314、140、403、032、324、241、241（命中）、241（命 中）、413 结果：4 1 3
LRU：2、23、231、314、140、403、032、324、241、241（命中）、241（命 中）、243 结果：2 4 3
FIFO 反映了时间这个要素，LRU 反映了频率这个要素。（合理即可）
(2) 如果增大缓冲区的程度足够大，以至于能够包含整个工作集，则充足的增大缓冲区对两种方法都是有效的。如果增大缓冲区的程度不够大，则不一定总是有效，比如当输入序列为 3 2 1 0 3 2 4 3 2 1 0 4，使用大小为 3 的缓冲区会有 9 次 miss；使用大小为 4 的缓冲区会有 10 次 miss。
(3) 这个是自由发挥的题目。

2. 置换选择排序的核心思想是利用堆对数据进行处理。每输出一个值，就从缓冲区中读入下一个数。如果堆的大小是 M，一个顺串的最小长度就是 M 个记录，至少原来在堆中的那些记录将成为顺串的一部分。最好的情况下，例如输入已经被排序，有可能一次就把整个文件生成为一个顺串。
(1) 现在给出一组输入关键字(17, 2, 20, 40, 10, 19, 8, 13, 11, 25, 21, 7)，假设堆的大小是 5 且起始为空，请写出得到的初始顺串和最后堆的状态。排序时较小的元素在前。
(2) 置换选择排序一定要用堆来实现吗？请任意给出一个不同的实现，用上面的输入和设定比较一下两者的差异。

答：
(1) 顺串是(2, 10, 17, 19, 20, 25, 40)。堆的状态是
7
21 11
13 8
(2) 不一定。堆只是一种非常直观的实现，而且空间利用率高。
可以用优先队列实现，不过相比堆需要 2 倍的空间。也可以有其它自由的实现。

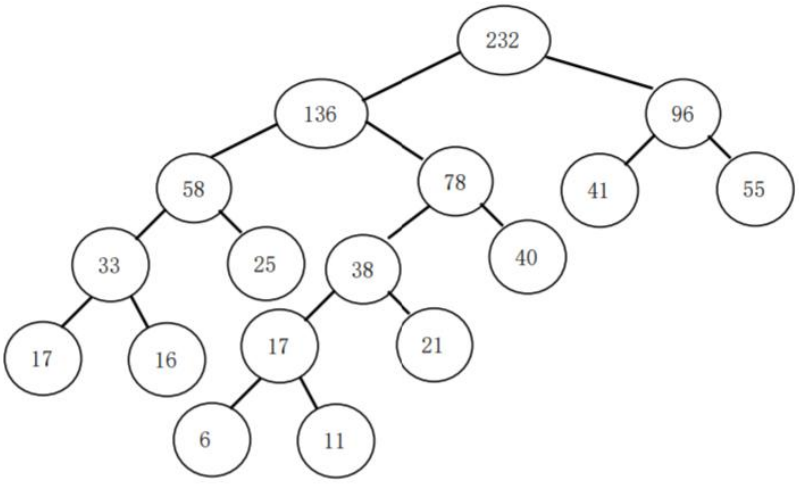
3. 现在有 9 个长度不同的的顺串，其长度分别为 17, 40, 16, 55, 25, 11, 6, 21, 41。请用二路归并的方法对其进行归并。

(1) 构造最佳归并树。

(2) 根据最佳归并树计算每一趟和总的读记录次数。

(3) 在多路归并的时候，显然归并的路数 K 越大速度越快，那么是什么限制了 K 的大小？

答：(1)



(2) 每一趟的次数相加为 17+33+38+58+78+96+136+232=688。总次数为 688。

(3) 若不使用胜者树或败者树等数据结构，则随着 k 的增大，每次都要进行 k 次比较才能找到最小值，很慢。若使用胜者树或败者树等数据结构，需要与 k 相关的额外的空间进行存储，则内存空间限制了 k 的大小。

4. 胜者树和败者树都是完全二叉树，是树形选择排序的一种变型。每个叶子结点相当于一个选手，每个中间结点相当于一场比赛，每一层相当于一轮比赛。不同的是，胜者树的中间结点记录的是胜者的标号；而败者树的中间结点记录的败者的标号。

胜者树与败者树可以在 log(n)的时间内找到最值。任何一个叶子结点的值改变后，利用中间结点的信息，还是能够快速地找到最值。在 k 路归并排序中经常用到。

举例说明为什么败者树的访问内存次数要比胜者树少，并分析是什么原因造成的。

答：例子略。

在重构的时候，败者树每一层和父亲节点比较，并更新父亲节点；胜者树每一层和兄弟节点比较，再更新父亲节点。两者比较次数是相同的，但胜者树的读写操作一般更多一些。假设更新的路径上有 m 个内部节点，胜者树需要读取路径上所有兄弟节点（m 次读），写入路径上所有父亲节点（m 次写），或者读取路径上所有父亲节点，只写入有更新的节点（m 次读+小于 m 次写），所以最优的读写次数为 2m。而败者树需要读取路径上所有父亲节点（m 次读），只写入有更新的节点（小于 m 次写），所以读写次数小于 2m。

5. 假设一个记录长为 32 个字节，一个块长 1024 个字节（每个块有 32 个记录），工作内存是 1MB（还有用于 I/O 缓冲区、程序变量等的其他存储空间）。使用置换选择和多路归并，其中归并算法只允许扫描两遍。预计能得到的文件最长为多少？并解释这个结果是怎样计算出来的。

答：每个记录长 32 字节，因此 1MB 内存可以容纳的记录数为：1MB/32 字节=32K 个记录。根据扫雪机原理，平均得到的顺串长度为 2 x 32K=64K 个记录，顺串的大小为 2MB。每个块大小为 1024 个字节，因此 1MB 工作内存可以同时处理 1024 块，多路归并的最大数目为 1024 路归并，因此一遍扫描可以得到最长 2MB x 1024=2048MB=2GB 的顺串，两遍扫描可以得到最长 2GB x 1024=2TB 的顺串。