

8.7 外部排序（下）

置换-选择排序（生成初始归并段）

实现过程

- 设初始待排文件为FI,初始归并段输出文件为FO,内存工作区为WA,FO和WA的初始状态为空,WA 可容纳 w 个记录
- 1) 从FI输入 w 个记录到工作区 WA
 - 2) 从 WA 中选出其中关键字取最小值的记录,记为MINIMAX记录
 - 3) 将 MINIMAX 记录输出到 FO中去
 - 4) 若FI不空,则从FI输入下一个记录到WA中
 - 5) 从 WA 中所有关键字比MINIMAX 记录的关键字大的记录中选出最小关键字记录,作为新的 MINIMAX记录
 - 6) 重复3) ~ 5) ,直至在WA中选不出新的MINIMAX记录为止,由此得到一个初始归并段,输出一个归并段的结束标志到 FO中去
 - 7) 重复2) ~ 6) ,直至WA为空。由此得到全部初始归并段

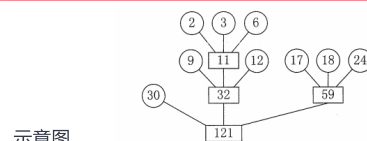
结构概述

- 各叶结点表示一个初始归并段,上面的权值表示该归并段的长度
- 叶结点到根的路径长度表示其参加归并的趟数
- 各非叶结点代表归并成的新归并段
- 根结点表示最终生成的归并段
- 树的带权路径长度WPL为归并过程中的总读记录数

引入哈夫曼树的思想

在归并树中,让记录数少的初始归并段最先归并,记录数多的初始归并段最晚归并,就可以建立总的I/O次数最少的最佳归并树

算法优化

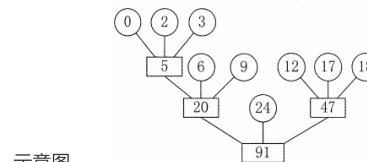


示意图

若初始归并段不足以构成一棵严格 k 叉树时,需添加长度为0的"虚段"

按照哈夫曼树的原则,权为0的叶子应离树根最远

算法修正



示意图

需要修正的条件

- 设度为0的结点有 $n_0 (=n)$ 个,度为 k 的结点有 n_k 个
- 严格 k 叉树有 $n_0 = (k-1)n_k + 1$ 变形可得 $n_k = (n_0 - 1) / (k-1)$
- $(n_0 - 1) \% (k-1) = 0$ 说明正好可以构造 k 叉归并树
- $(n_0 - 1) \% (k-1) = u \neq 0$ 再加上 $k-u-1$ 个空归并段,就可以建立归并树