

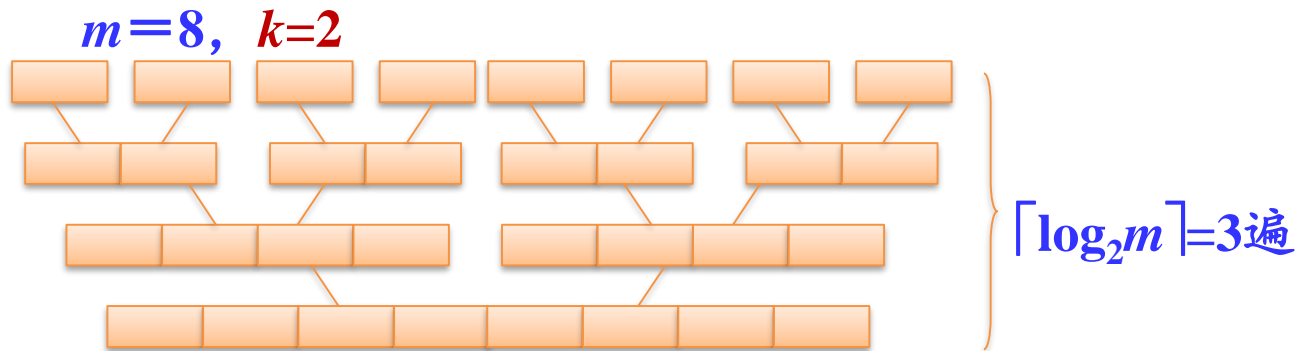
11.2.2 多路平衡归并

1、 k 路平衡归并概述

① 什么是 k 路平衡归并

2路平衡归并：每一趟从 m 个归并段得到 $\lceil m/2 \rceil$ 个归并段。

例如：



一般地，2路平衡归并的前提：初始归并段的记录个数都相同。

⇒ 可以推广到 k 路平衡归并

② 影响 k 路平衡归并的因素

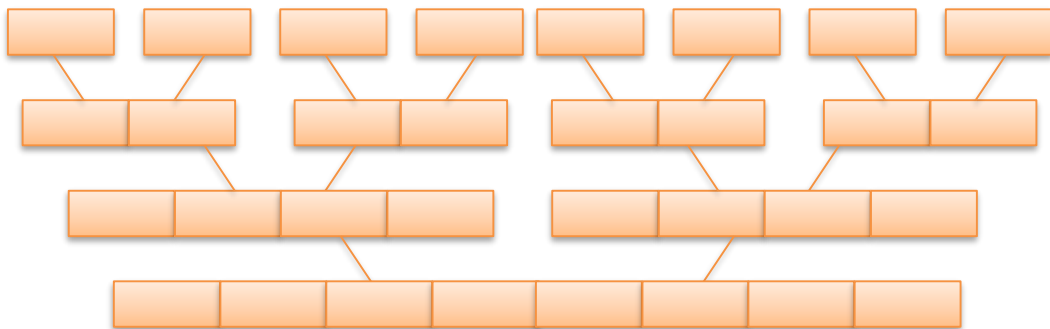
影响 k 路平衡归并的效率 的因素：

- 归并时需要读写磁盘的次数
- 归并时需要关键字比较的次数。

③ k 路平衡归并时读写磁盘次数的计算

例如

$m=8$, 假设每个归并段4个记录: $k=2$



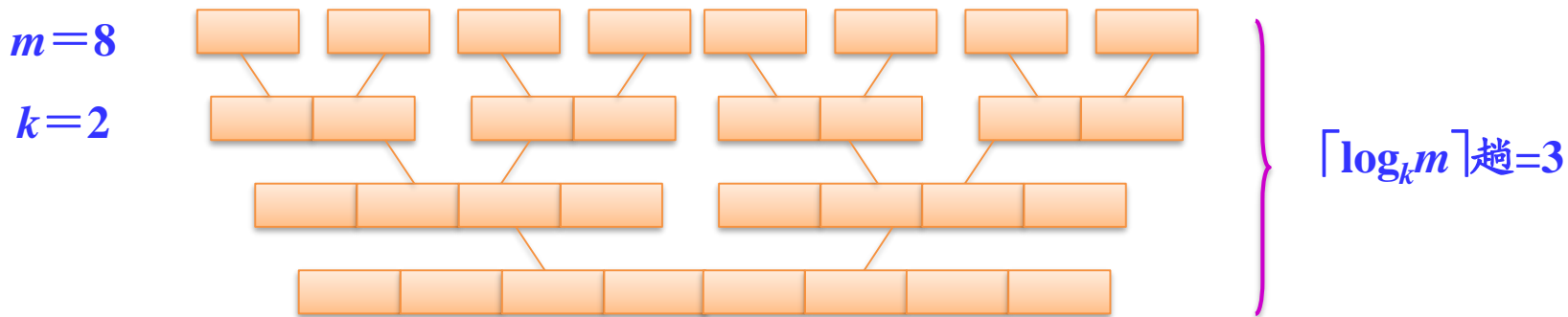
读记录次数 = $WPL = 8 \times 4 \times 3 = 96$

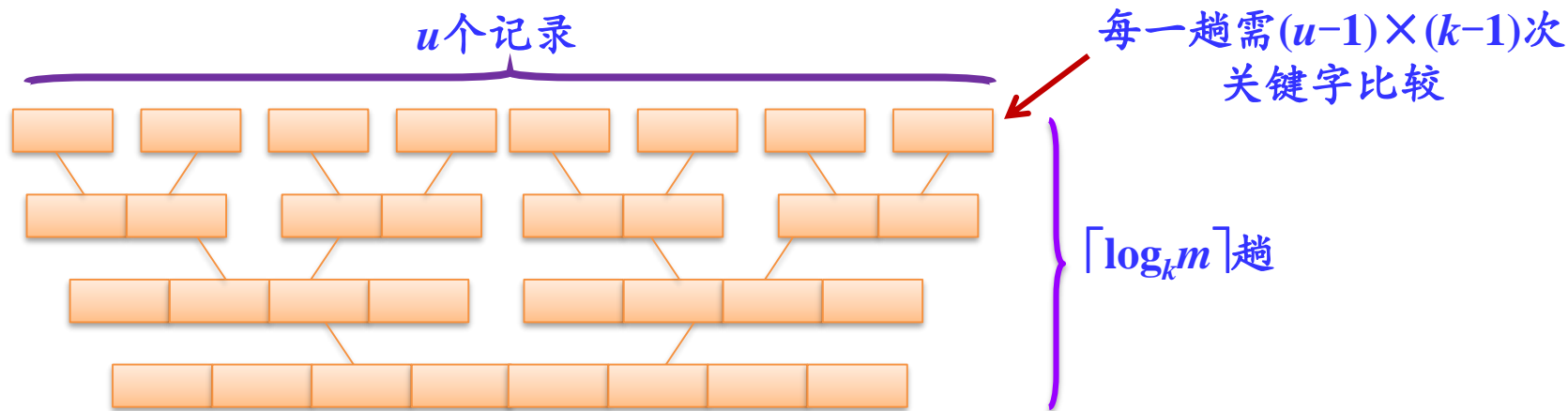
(如果每个记录占用一个物理块, 读写磁盘次数 = $96 \times 2 = 192$ 次)

采用 k 路平衡归并时, 通常 k 越大, 读写磁盘次数会减少。

④ k 路平衡归并时关键字比较次数的计算

采用 k 路平衡归并时，则相应的归并树有 $\lceil \log_k m \rceil + 1$ 层，要对数据进行 $\lceil \log_k m \rceil$ 趟扫描。





总共需要的关键字比较次数为：

$$\begin{aligned}
 & \lceil \log_k m \rceil \times (u-1) \times (k-1) \\
 &= \lceil \log_2 m \rceil / \lceil \log_2 k \rceil \times (u-1) \times (k-1) \\
 &= \lceil \log_2 m \rceil \times (u-1) \times (k-1) / \lceil \log_2 k \rceil
 \end{aligned}$$

总共需要的关键字比较次数：

$(k-1) / \lceil \log_2 k \rceil$ 在 k 增大时会增大

$$\lceil \log_2 m \rceil \times (u-1) \times (k-1) / \lceil \log_2 k \rceil$$

在初始归并段个数 m 与记录个数 u 确定时是常量



结论：增大归并路数 k ，读写磁盘次数减少，而关键字比较次数会增大。若 k 增大到一定的程度，就会抵消掉由于减少读写磁盘次数而赢得的时间。

2、利用败者树实现 k 路平衡归并过程

败者树用于在 k 个记录中选取最小关键字的记录。败者树类似于堆排序中的**堆**。

利用败者树实现 k 路平衡归并的过程是：

- 先建立败者树。
- 然后对 k 个输入有序段进行 k 路平衡归并。

【例11-2】 设有5个初始归并段，它们中各记录的关键字分别是：

$F_0: \{17, 21, \infty\}$

$F_1: \{5, 44, \infty\}$

$F_2: \{10, 12, \infty\}$

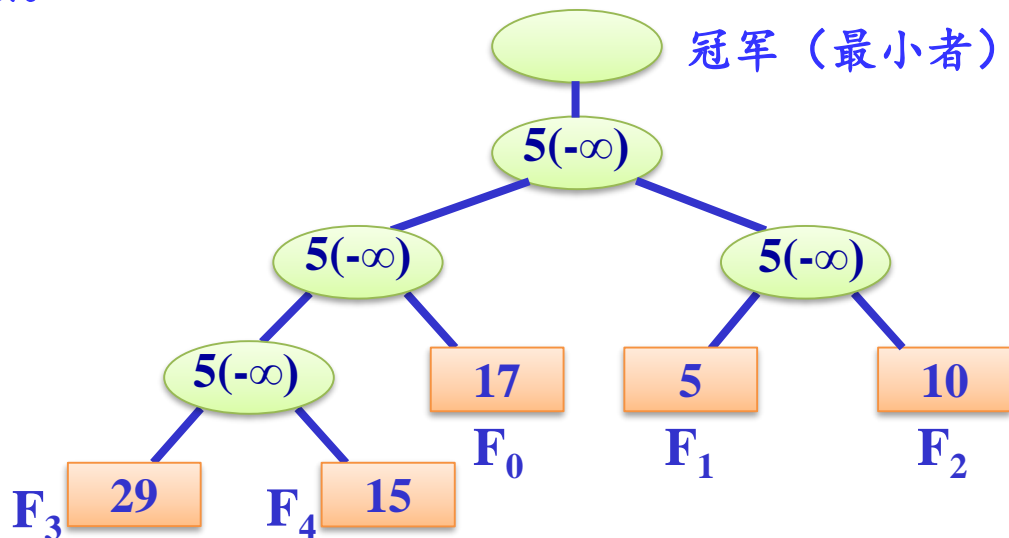
$F_3: \{29, 32, \infty\}$

$F_4: \{15, 56, \infty\}$

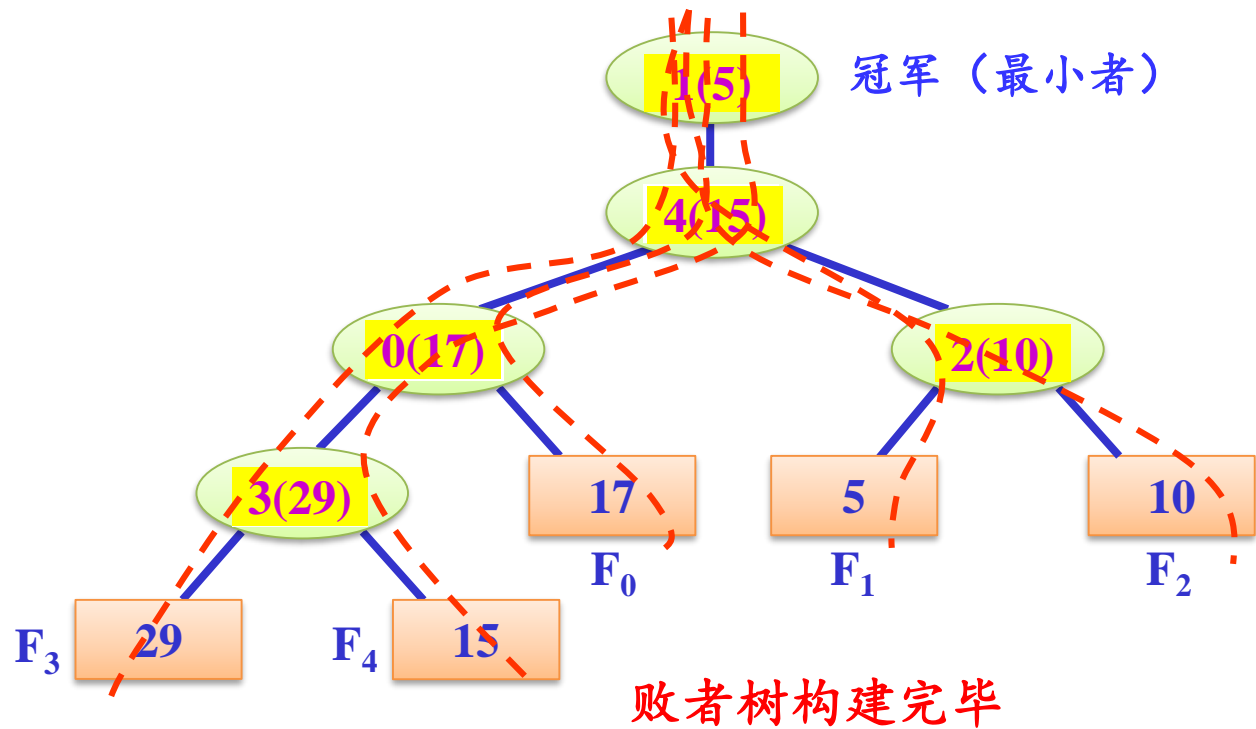
其中， ∞ 是段结束标志。说明利用败者树进行 $k=5$ 路平衡归并排序的过程。

① 构建败者树

- $k=5$: 创建含有 k 个叶子节点的完全二叉树，总共 $2k-1=9$ 个节点，另外添加一个冠军节点。

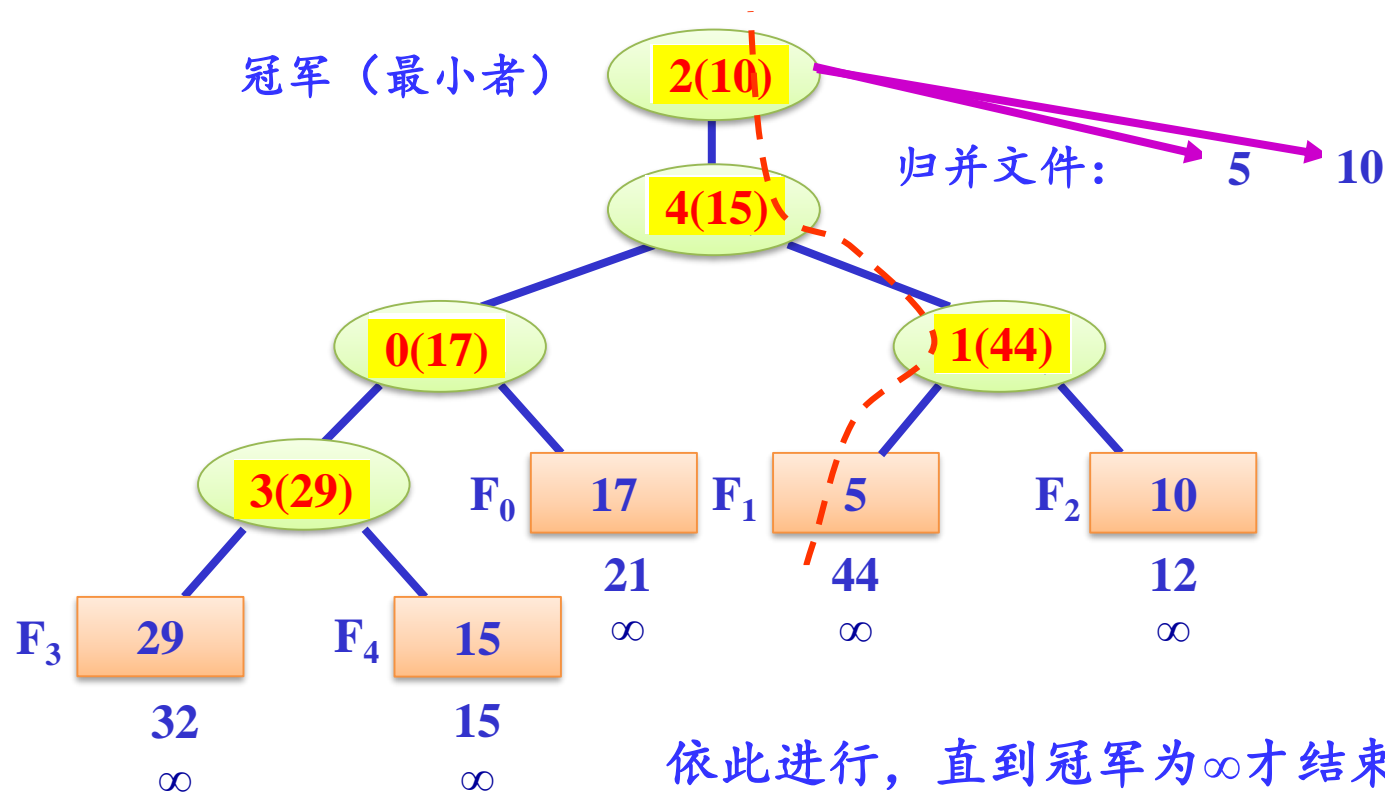


- 每个叶子节点对应一个归并段，段号为0~4。
- 初始时每个分支节点（含冠军节点）取值“ $5(-\infty)$ ”，5表示段号（此时为虚拟段号）， $-\infty$ 表示最小关键字。例如，某节点取值为“ $4(15)$ ”，表示节点值来自4号段的关键字15对应的记录。



调整产生冠军 (最小者) 的过程：从 $F_4 \Rightarrow F_0$ 操作：将当前节点的关键字与父节点比较，将大的 (败者) 放在父节点中，小者 (胜者) 继续进行，直到根节点。最后将胜者放在冠军节点中。

② 用败者树进行归并



利用败者树实现 k 路平衡归并时，总共需要的关键字比较次数为：

$$\begin{aligned} & \lceil \log_k m \rceil \times (u-1) \times \lceil \log_2 k \rceil \\ &= \lceil \log_2 m \rceil \times (u-1) \times \lceil \log_2 k \rceil \lceil \log_2 k \rceil \\ &= \lceil \log_2 m \rceil \times (u-1) \end{aligned}$$

结论：关键字比较次数与 k 无关 \Rightarrow 总的内部归并时间不会随 k 的增大而增大。



利用败者树实现 k 路平衡归并

只要内存空间允许，尽可能增大归并路数 k 。

思考题

- ① 败者树的作用是什么？
- ② 败者树类似于堆，两者有什么不同？

采用败者树，置换-选择排序中关键字比较次数分析

共有 n 个记录，内存工作区WA的容量为 w ：

- 若在 w 个记录中选取最小关键字的采用败者树方法，每次需要 $\log_2 w$ 次比较。
- 总的时间复杂度为 $O(n \log_2 w)$ 。

——本讲完——