



本节主题:

最佳归并树

最佳归并树的目标

背景

- 采用置换-选择排序的方法生成的初始归并段长度不等
- 在进行逐趟k路归并时，对归并段的组合不同，会导致归并过程中对外存的读 / 写次数不同。

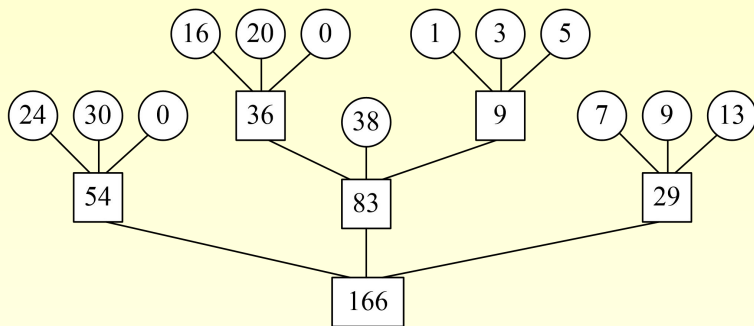
目标

- 为提高归并的时间效率，有必要对各归并段进行合理的搭配组合。
- 按照最佳归并树的设计可以使归并过程中对外存的读 / 写次数最少。



归并树

- 归并树是用于描述归并过程的k次图
- 归并树是只包含度为0和度为k的节点的标准树
 - 每一次k路归并，都需要有k个归并段参加
- 归并树的节点
 - 各叶子节点，代表参加归并的各初始归并段
 - 叶子节点上的权值，为归并段的记录个数
 - 根节点代表最终生成的归并段
 - 各非叶子节点，代表归并出来的新归并段
 - 叶子节点到根节点的路径长度，代表在归并过程中的读记录次数
 - 归并树的带权路径长度WPL，代表归并过程中的总的读记录次数



$$\begin{aligned} WPL &= (24+30+0+38+7+9+13) \times 2 \\ &\quad + (16+20+0+1+3+5) \times 3 \\ &= 377 \end{aligned}$$

归并过程中总的读写次数：
 $2 \times 377 = 754$

归并树性质

- 归并树是正则 k 次数
 - 只有度为0和度为 k 的节点
- 节点数目间的关系
 - 树的总节点数 N
 - 度为0的节点数 m_0
 - 度为 k 的节点数 m_k

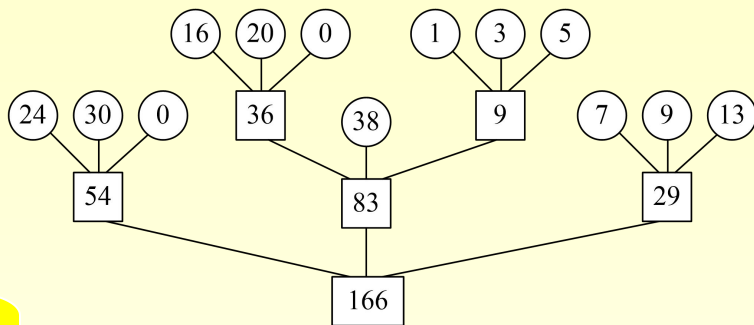
$$N = m_0 + m_k$$

$$N = k \cdot m_k + 1$$

$$\Rightarrow m_0 = (k-1)m_k + 1$$

$$\Rightarrow m_k = \frac{m_0 - 1}{k-1}$$

切，
这也玩湊数！



应用

- 对 n 个初始归并段进行归并
- 若 $t = (n-1) \% (k-1) \neq 0$ ，
增加 $(k-1)-t$ 个“虚”节点，
才能够构造出归并树。

最佳归并树及其构造

目标

- 构造带权路径长度最短的k叉(阶)归并树(最佳归并树),用于指导归并排序过程

策略

- k叉(阶)哈夫曼树

构造步骤:

- 若 $t = (n-1) \% (k-1) \neq 0$, 则需附加 $(k-1)-t$ 个长度为0的虚段, 以使每次归并都可以对应k个段。
- 按照哈夫曼树的构造原则构造最佳归并树, 使权值越小的结点离根结点越远。

例

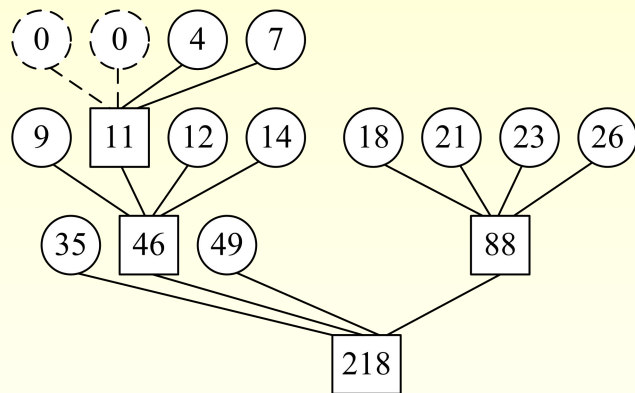
- 设文件经预处理后, 得到长度为11个初始归并段 {47, 9, 39, 18, 4, 12, 23, 7, 21, 16, 26}

- 为4路归并设计一个读写文件次数最少的归并方案

解

- $t = (11-1) \% (4-1) = 2$, 附加2个0长度的归并串

- 构造出如下4路最佳归并树



若每个记录占用一个物理页块, 对外存的读写次数为:
 $2 \times [(4+7) \times 3 + (9+12+14+18+21+23+26) \times 2 + (35+49) \times 1]$
 $= 726$ 次。