



数据结构与算法（十一）

张铭 主讲

采用教材：张铭，王腾蛟，赵海燕 编写
高等教育出版社，2008.6（“十一五”国家级规划教材）

<http://www.jpk.pku.edu.cn/pkujpk/course/sjjg>



主要内容

- 基本概念
- 11.1 线性索引
- 11.2 静态索引
- 11.3 倒排索引
- 11.4 动态索引
 - 11.4.1 B 树
 - 11.4.2 B 树的性能分析
 - 11.4.3 B⁺ 树
 - 11.4.4 B 树、B⁺ 树索引性能的比较
- 11.5 位索引技术
- 11.6 红黑树



基本概念

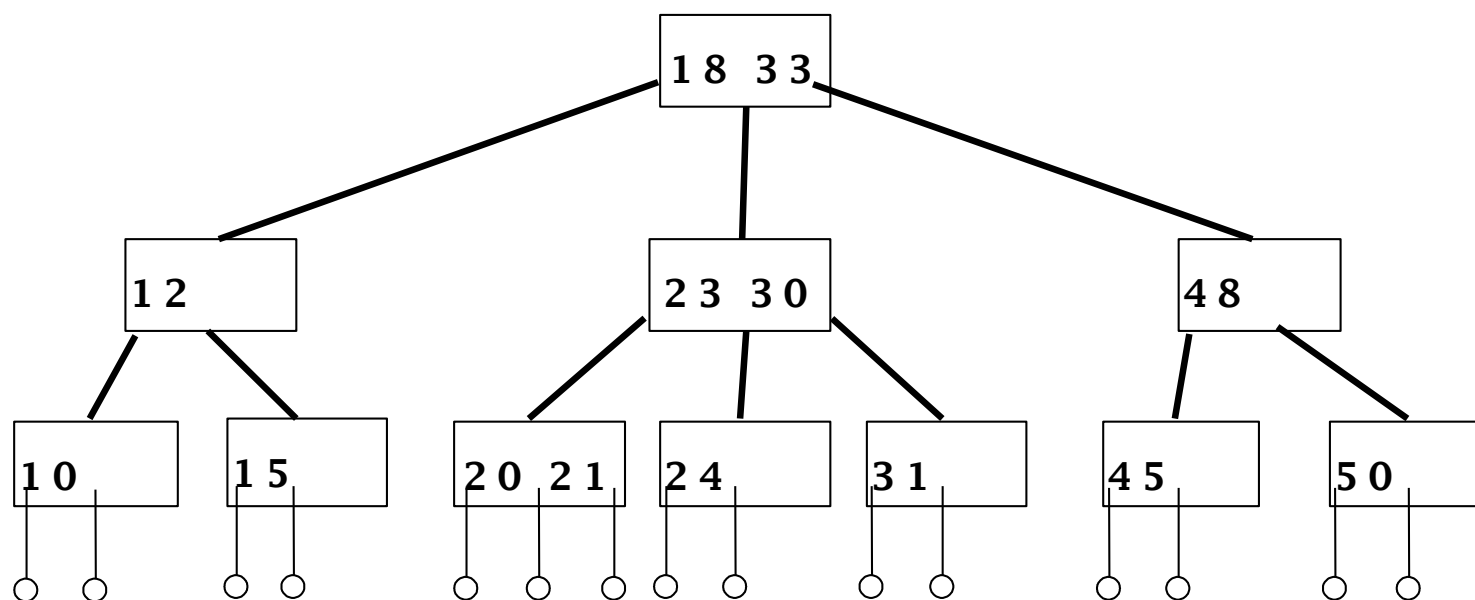
- 动态索引结构
 - 索引结构本身也可能发生改变
 - 在系统运行过程中插入或删除记录时
- 目的
 - 保持较好的性能
 - 例如较高的 检索 效率

11.4 动态索引

11.4.1 B 树

- 一种平衡的多分树 (Balanced Tree)

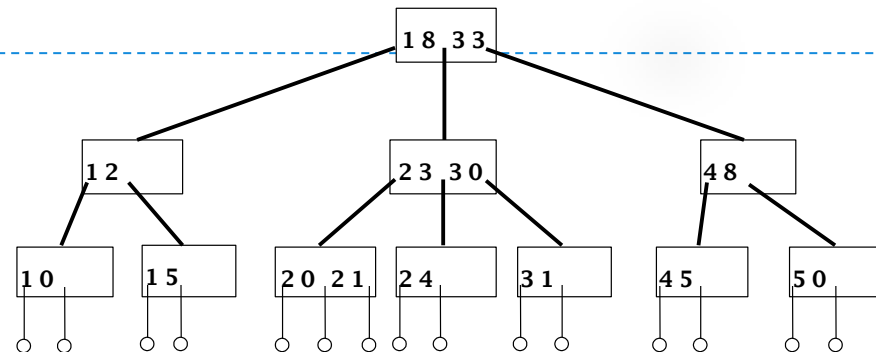
3 阶 B 树 2-3 树





m 阶 B 树的结构定义

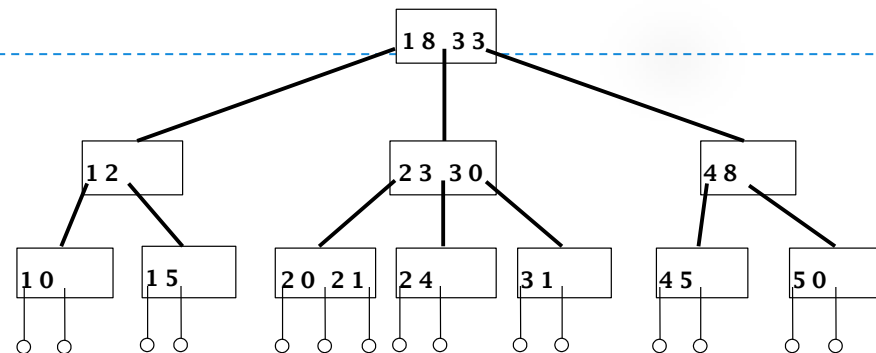
- (1) 每个结点至多有 m 个子结点
- (2) 除根结点和叶结点外，其它每个结点至少有 $\lceil \frac{m}{2} \rceil$ 个子结点
- (3) 根结点至少有两个子结点
 - 唯一例外的是根结点就是叶结点时没有子结点
 - 此时 B 树只包含一个结点
- (4) 所有的叶结点在同一层
- (5) 有 k 个子结点的非根结点恰好包含 $k-1$ 个关键码





B 树的性质

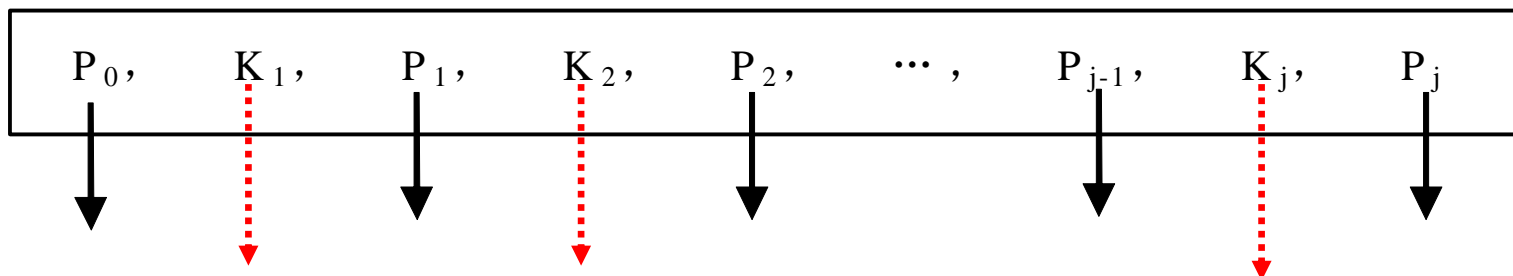
- (1) 树高平衡，所有叶结点都在同一层
- (2) 关键码没有重复，父结点中的关键码是其子结点的分界
- (3) B 树把（值接近）相关记录放在同一个磁盘页中，从而利用了访问局部性原理
- (4) B 树保证树中至少有一定比例的结点是满的
 - 这样能够改进空间的利用率
 - 减少检索和更新操作的磁盘读取数目





B 树的结点结构

B 树的一个包含 j 个关键码, $j+1$ 个指针的结点的一般形式为:

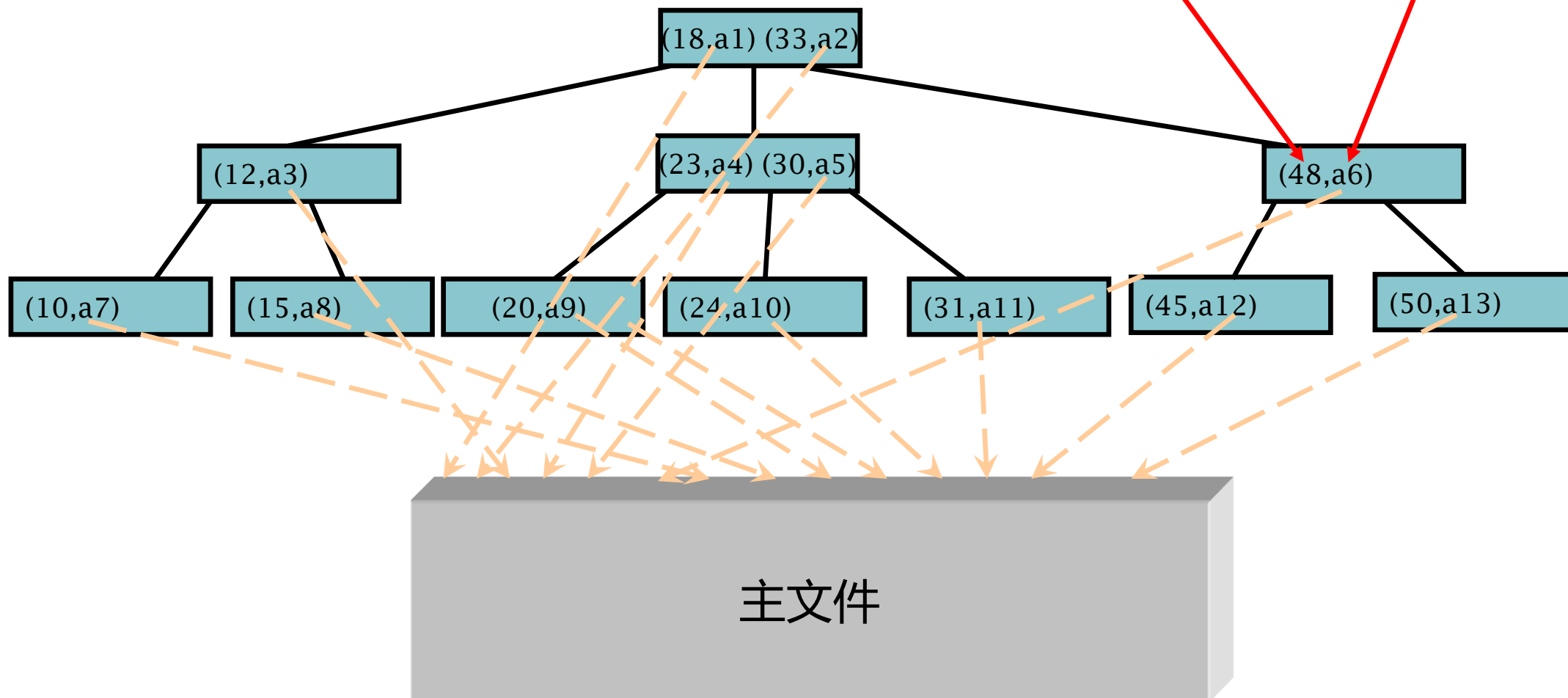


- 其中 K_i 是关键码值, $K_1 < K_2 < \dots < K_j$,
- P_i 是指向包括 K_i 到 K_{i+1} 之间的关键码的子树的指针。
- 还有指针吗?

11.4.1 B 树

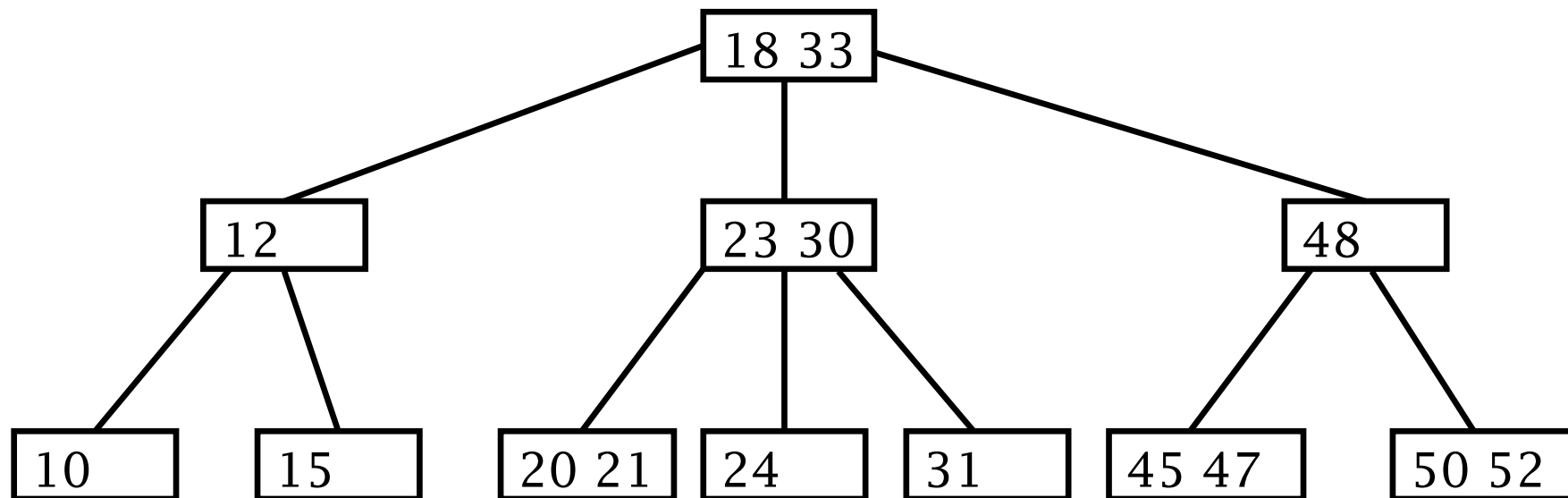
B 树隐含指针

(关键码, 文件页内地址)



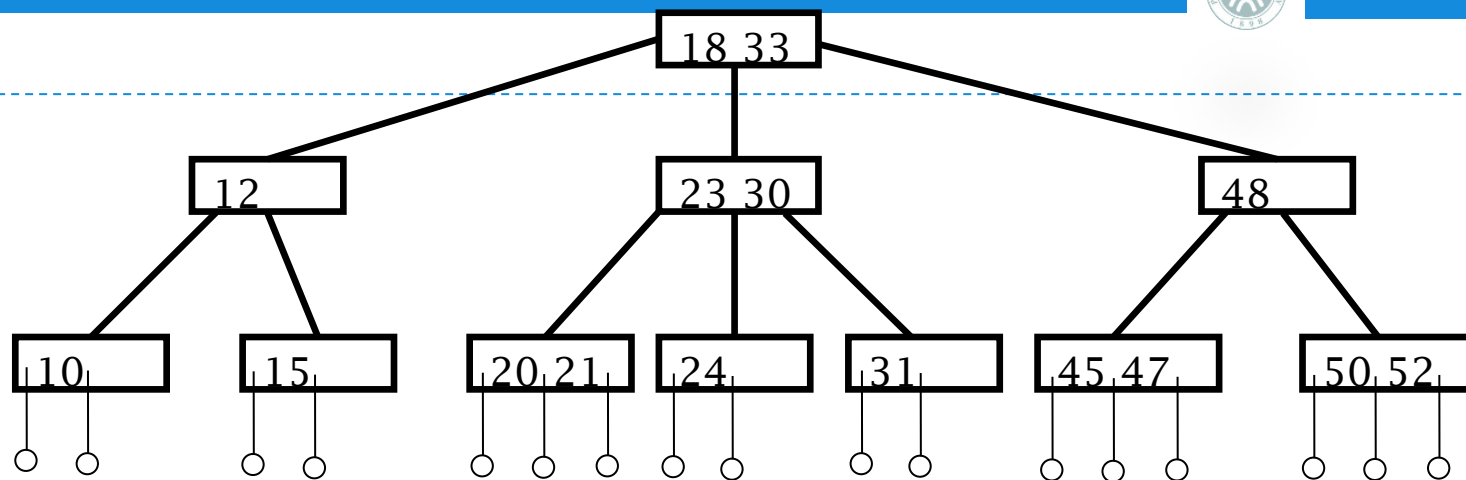


2-3 树 = 3 阶 B 树 (阶应该 ≥ 3)





B 树的查找



· 交替的两步过程

- 1. 把根结点读出来，在根结点所包含的关键码 K_1, \dots, K_j 中查找给定的关键码值
 - 找到则检索成功
- 2. 否则，确定要查的关键码值是在某个 K_i 和 K_{i+1} 之间，于是取 p_i 所指向的结点继续查找
- 如果 p_i **指向外部空结点**，表示检索失败



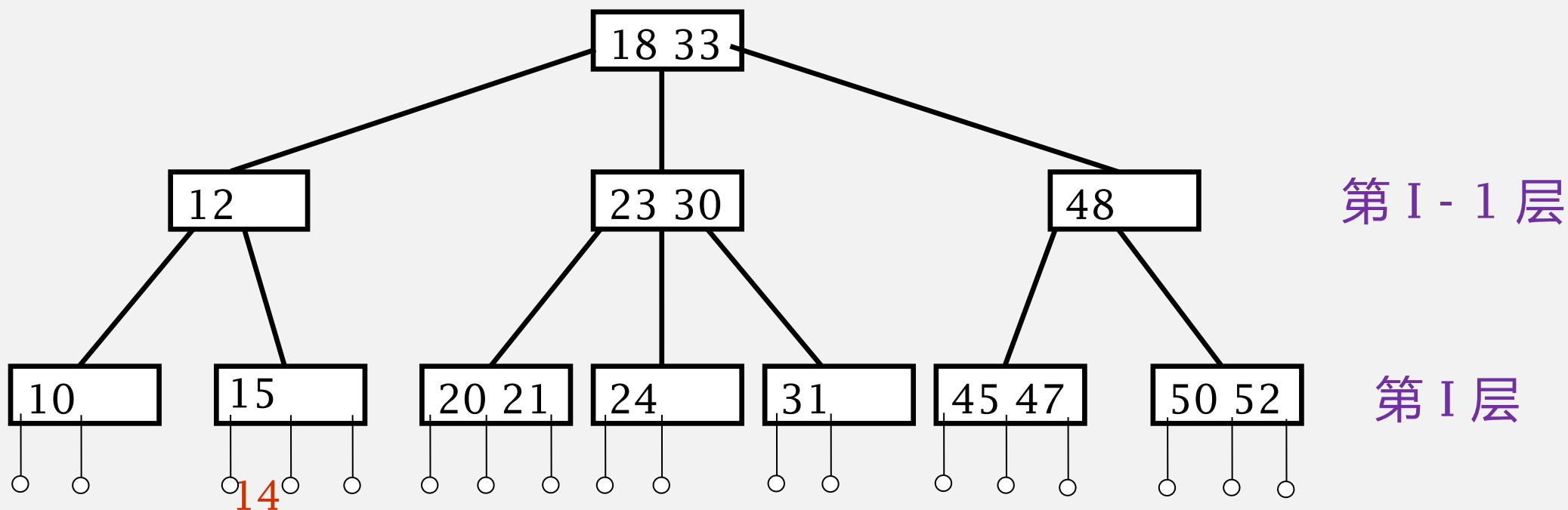
B 树插入

- 注意保持性质，特别是等高和阶的限制
 - 1) 找到最底层，插入
 - 2) 若溢出，则结点分裂，中间关键码连同新指针插入父结点
 - 3) 若父结点也溢出，则继续 **分裂**
 - 分裂过程可能传达到根结点(则树升高一层)

B 树的插入

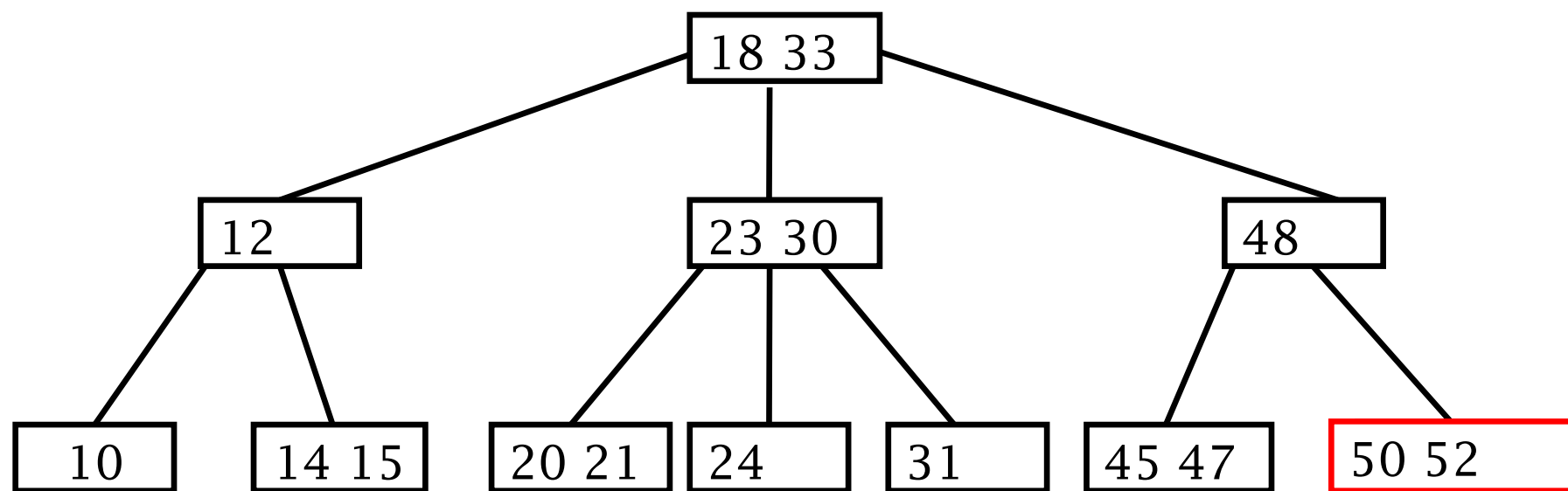
- 外部空结点（即失败检索）处在第 I 层的 B 树，插入的关键码总是在第 I-1 层

3阶B树 插入14





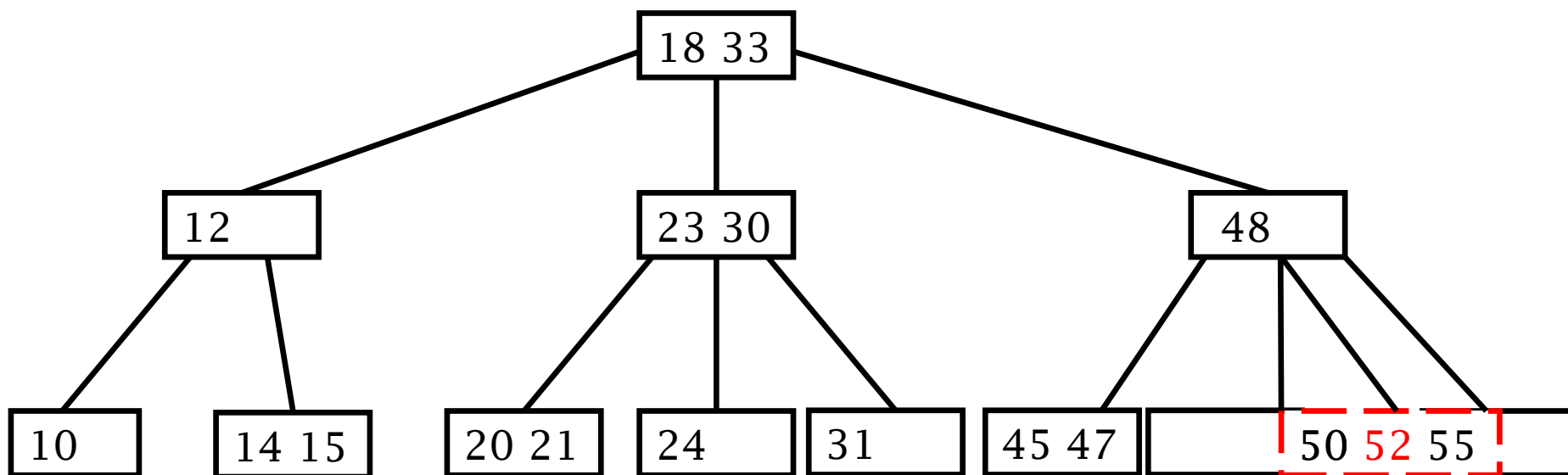
m=3 , 插入 55



55

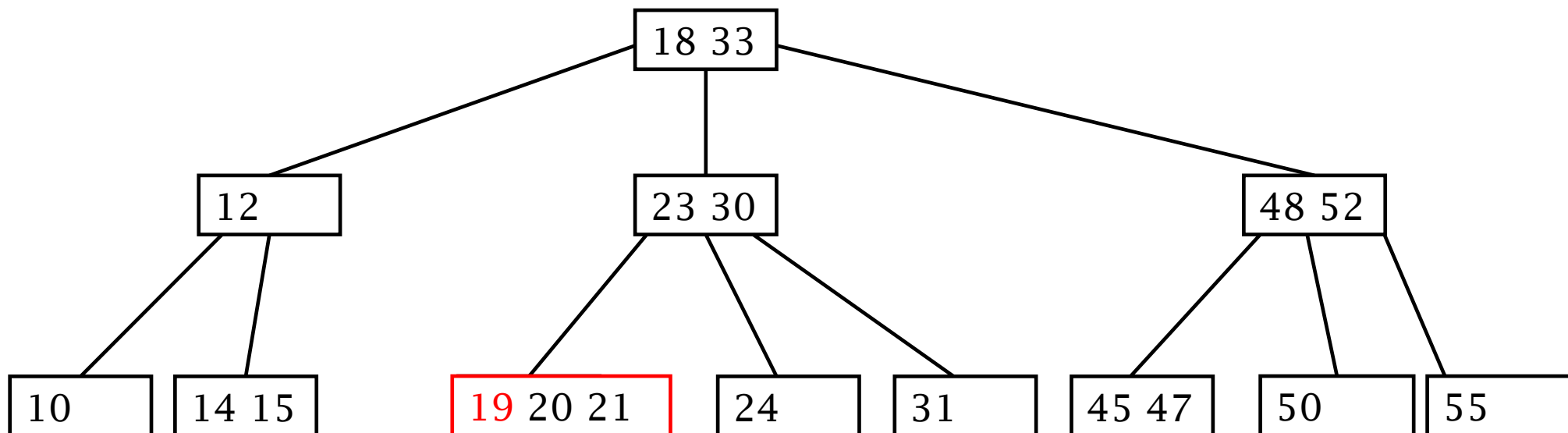


$m=3$ ，叶结点分裂，把 52 提升到父结点





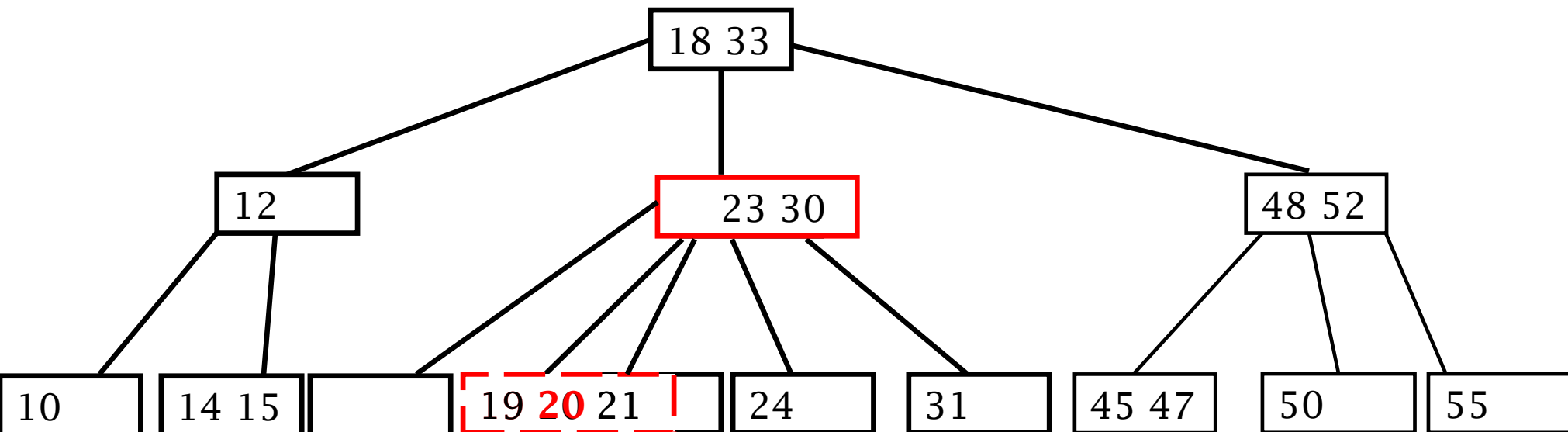
插入 19, 引起 3 阶 B 树根结点分裂



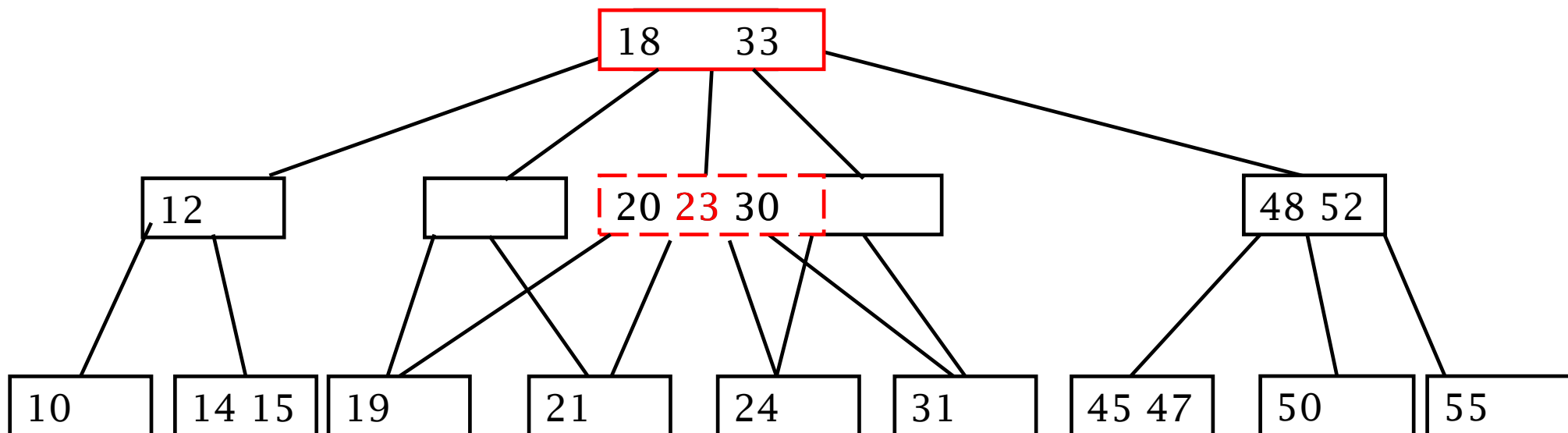
19



m=3, 叶结点分裂

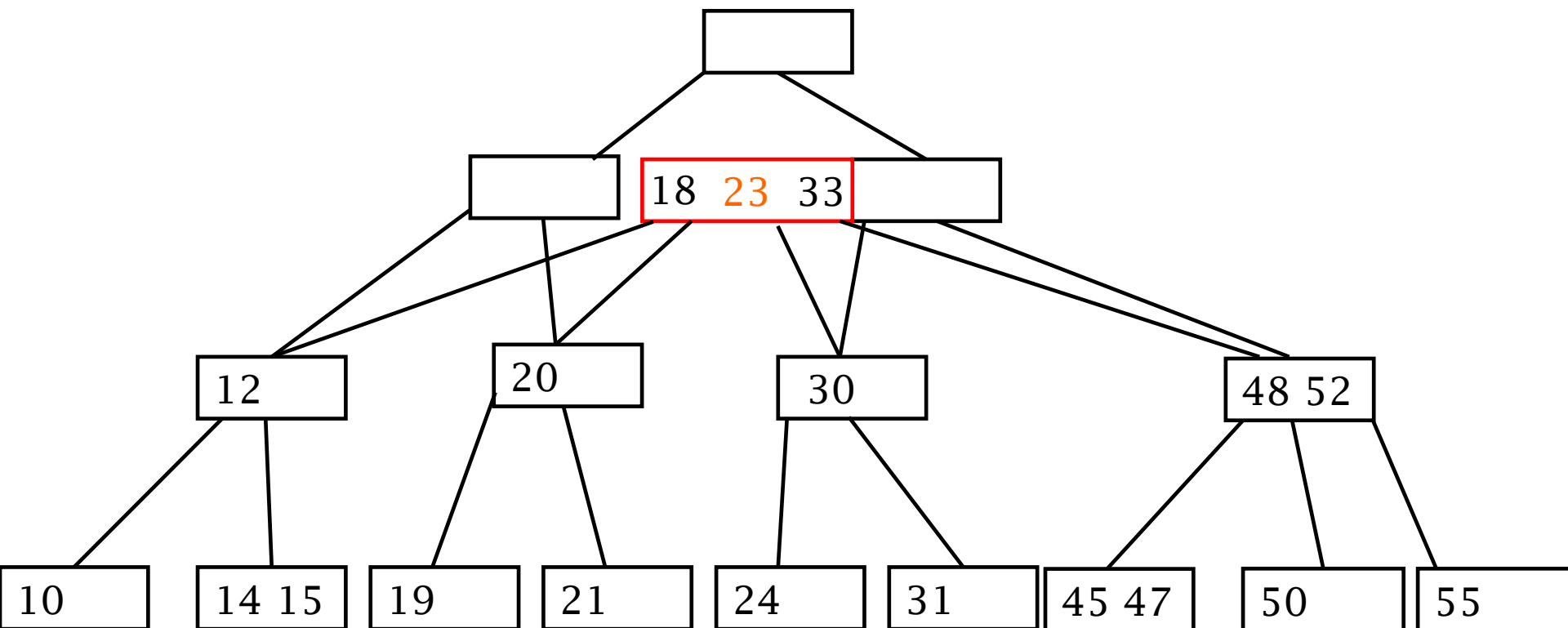


m=3, 第二层结点分裂



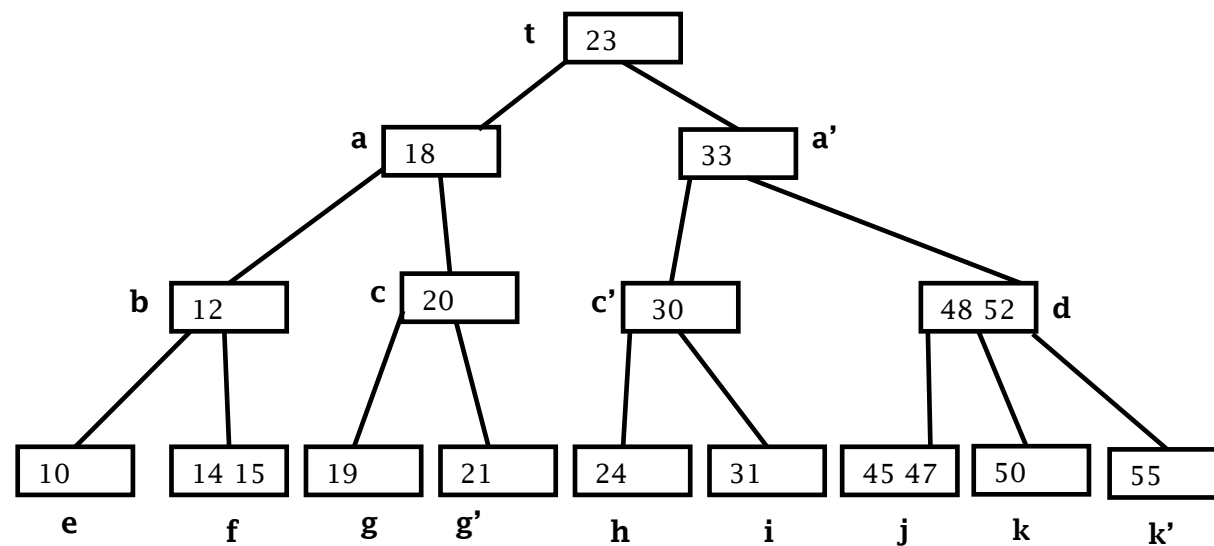
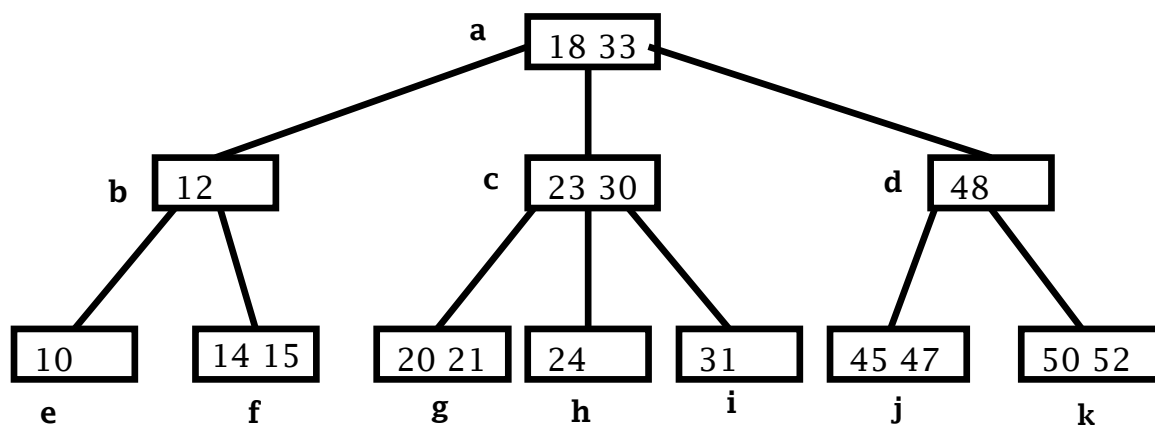


m=3, 根结点分裂



B 树操作的访外次数

- 连续插入14、55、19，假设访问过的都缓存
- 读盘7次 (a,b,f; d,k; c,g)
- 写盘11次 (f; k,k',d; g,g',c,c',a,a',t)



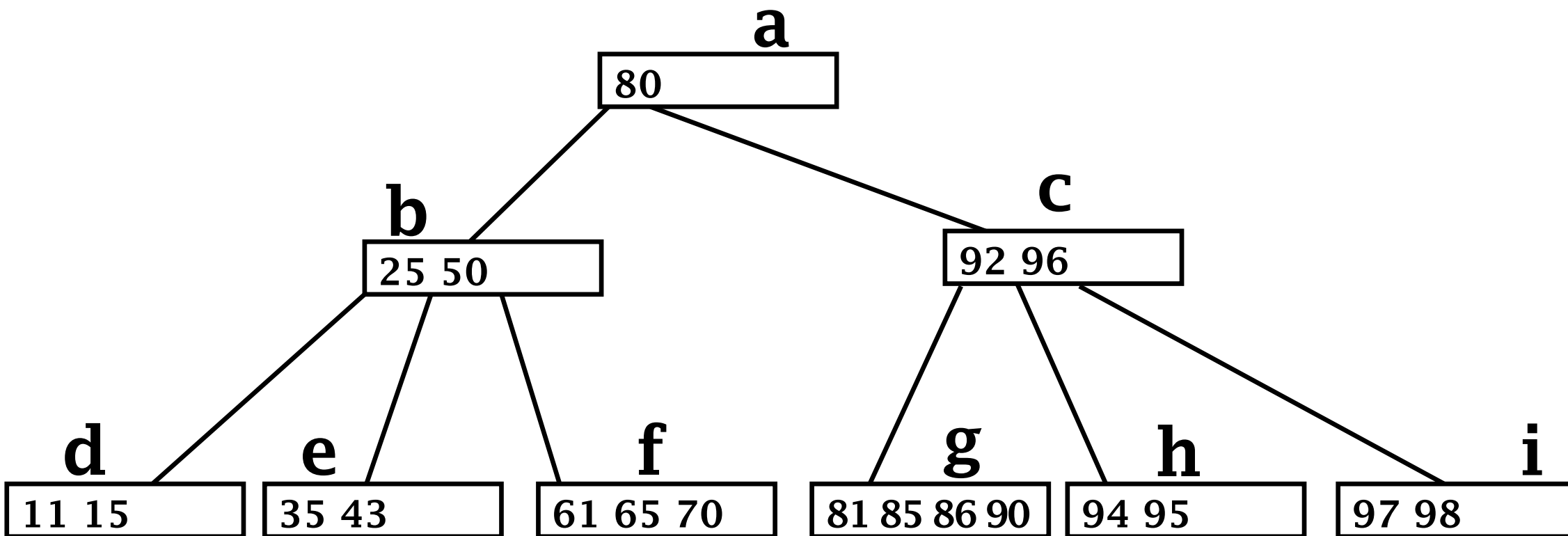


B 树的删除

- 删除的关键码不在叶结点层，跟叶中后继对换
- 删除的关键码在叶结点层
 - 删除后关键码个数**不小于** $\lceil m / 2 \rceil - 1$ ，**直接** 删除
 - 关键码个数**小于** $\lceil m / 2 \rceil - 1$
 - 如果兄弟结点关键码个数不等于 $\lceil m / 2 \rceil - 1$
 - 从兄弟结点移若干个关键码到该结点中来(父结点中的一个关键码要做相应变化)
 - 如果兄弟结点关键码个数等于 $\lceil m / 2 \rceil - 1$
 - 合并



5 阶 B 树删除示例

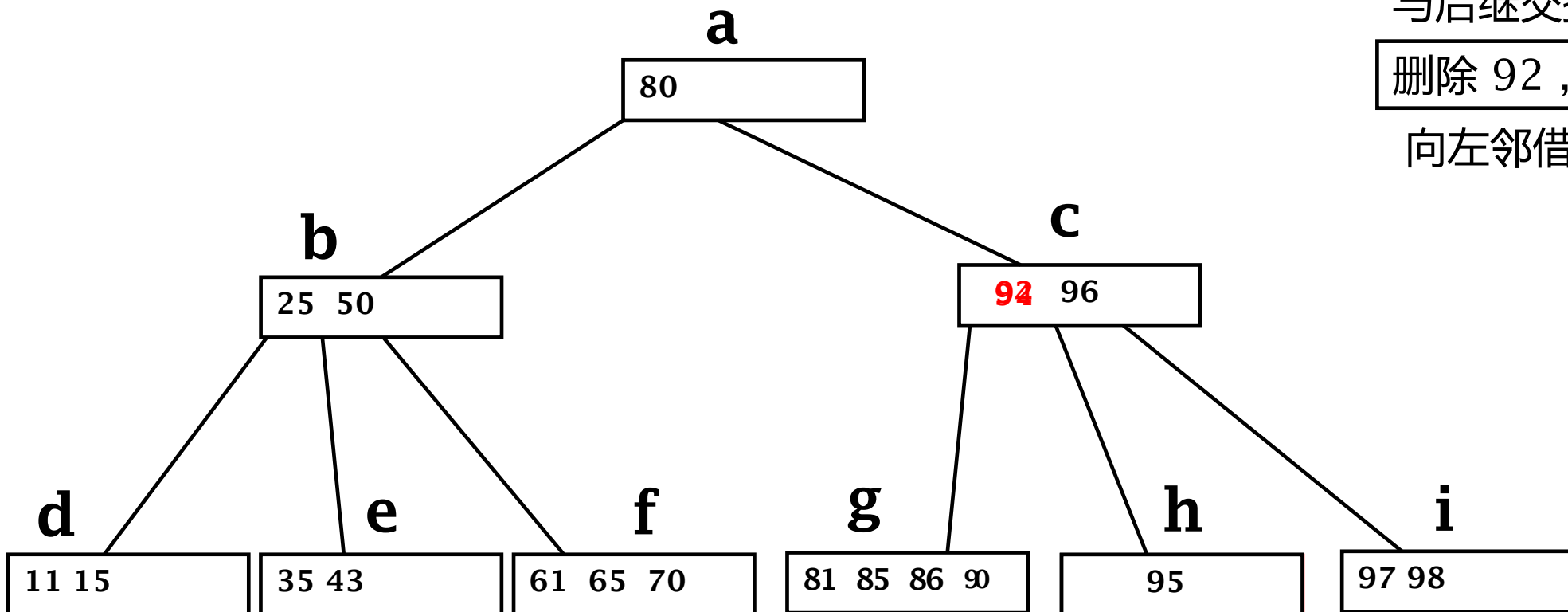


5 阶 B 树删除示例

与后继交换

删除 92，h 溢出

向左邻借关键码

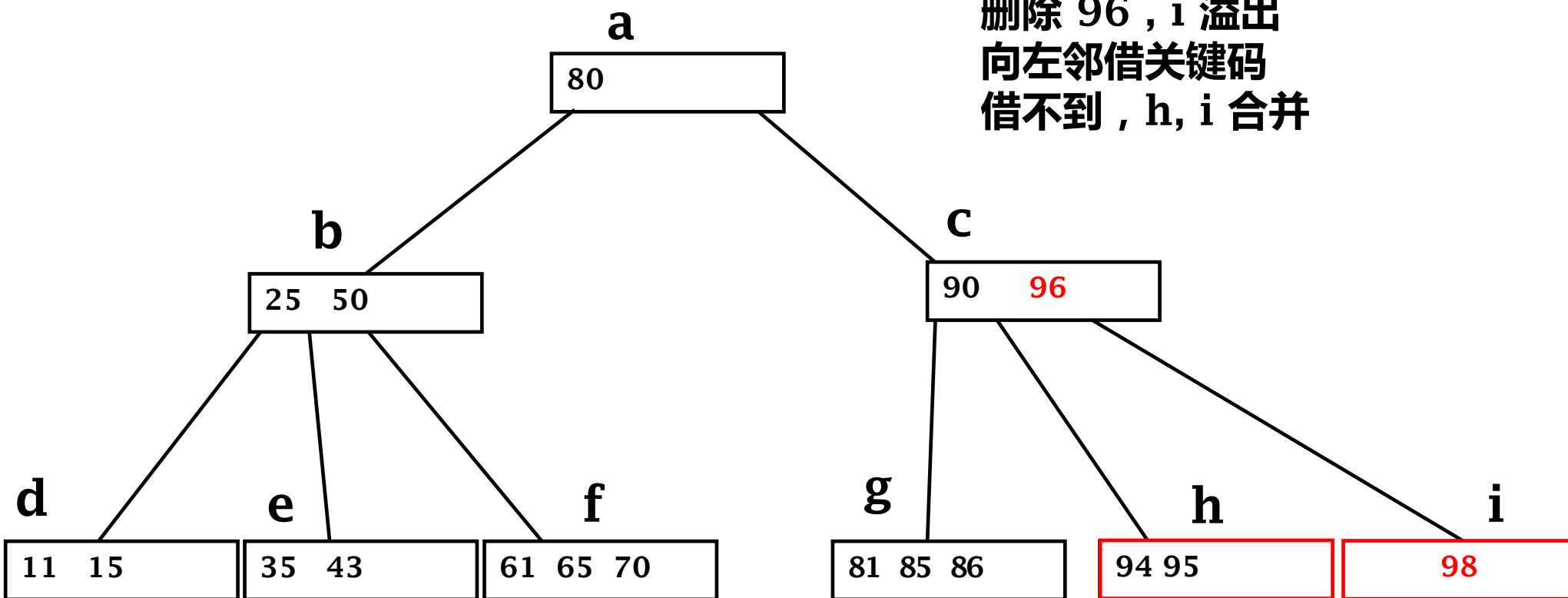


删除92，先与后继交换，删后下溢出，向左邻借关键码



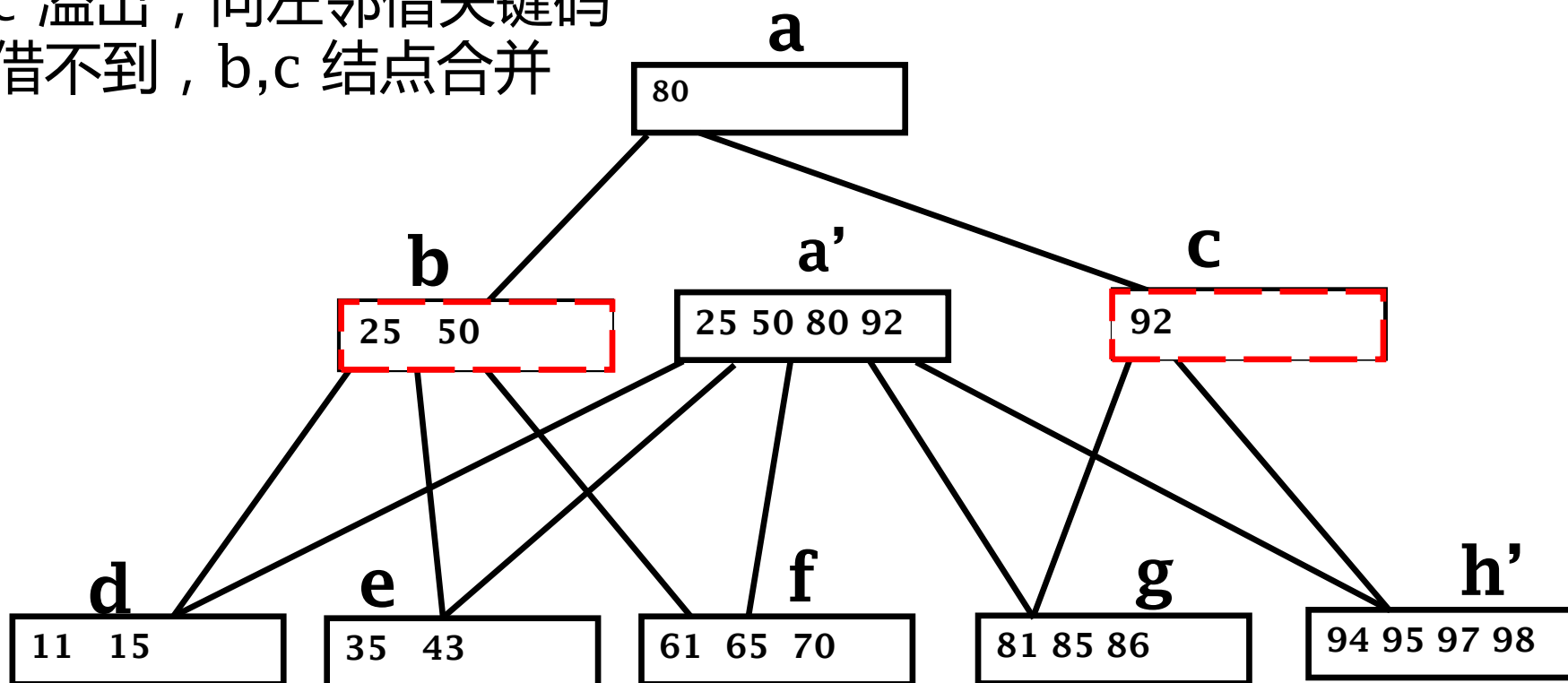
5 阶 B 树删除示例

继续删除 96
与后继交换
删除 96, i 溢出
向左邻借关键码
借不到, h, i 合并



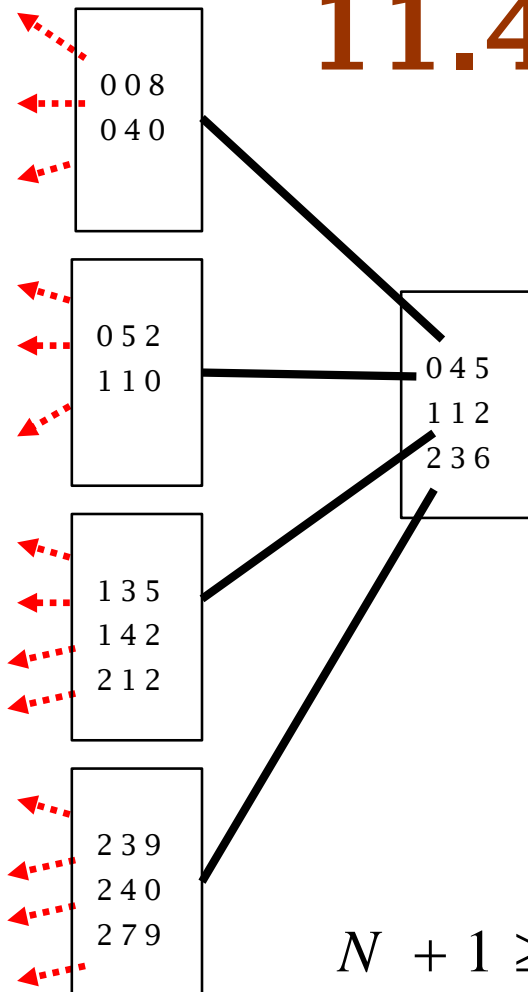
5 阶 B 树删除示例

h, i 合并成为 h'
c 溢出, 向左邻借关键码
借不到, b, c 结点合并



11.4.2 B 树的性能分析

- 包含 N 个关键码的 B 树
 - 有 $N+1$ 个外部空指针
 - 假设外部指针在第 k 层
- 各层的结点数目
 - 第 0 层为根，第一层至少两个结点，
 - 第二层至少 $2 \cdot \left\lceil \frac{m}{2} \right\rceil$ 个结点，
 - 第 k 层至少 $2 \cdot \left\lceil \frac{m}{2} \right\rceil^{k-1}$ 个结点，



$$N + 1 \geq 2 \cdot \left\lceil \frac{m}{2} \right\rceil^{k-1}, \quad k \leq 1 + \log_{\left\lceil \frac{m}{2} \right\rceil} \left(\frac{N + 1}{2} \right)$$



示例

- $N=1,999,998$, $m=199$ 时

$$k \leq 1 + \log_{\lceil m/2 \rceil} \left(\frac{N+1}{2} \right)$$

- $k=4$
- 一次检索最多 4 层



结点分裂次数

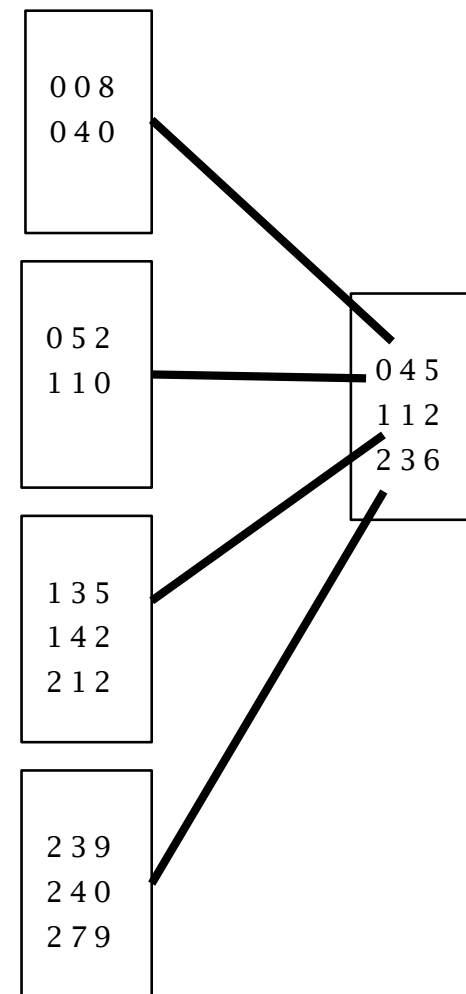
- 设关键码数为 N (空指针 $N+1$) , 内部结点数为 p

即
$$N \geq 1 + (\lceil m/2 \rceil - 1)(p - 1)$$

- 最差情况下每插入一个结点都经过分裂(除第一个), 即 $p-1$ 个结点都是分裂而来的, 则每插入一个关键码平均分裂结点个数为

$$p - 1 \leq \frac{N - 1}{\lceil m/2 \rceil - 1}$$

$$s = \frac{p - 1}{N} \leq \frac{N - 1}{(\lceil m/2 \rceil - 1) \cdot N} \leq \frac{1}{\lceil m/2 \rceil - 1}$$





思考

- 1. 是否存在符合定义的 2 阶 B 树？是否有实用价值？为什么？
- 2. B 树删除时使用先借用再合并的方法，为何在插入的时候不使用先送给兄弟结点再考虑分裂的方法？
- 3. B 树的定义中关于度数的定义为从 $\lceil \frac{m}{2} \rceil$ 到 m 之间，是否可以调整为其他范围？



数据结构与算法

谢谢聆听

国家精品课“数据结构与算法”

<http://www.jpk.pku.edu.cn/pkujpk/course/sjjg/>

张铭，王腾蛟，赵海燕

高等教育出版社，2008. 6。“十一五”国家级规划教材