

# 动态查找

大连理工大学

于 红

# 动态查找

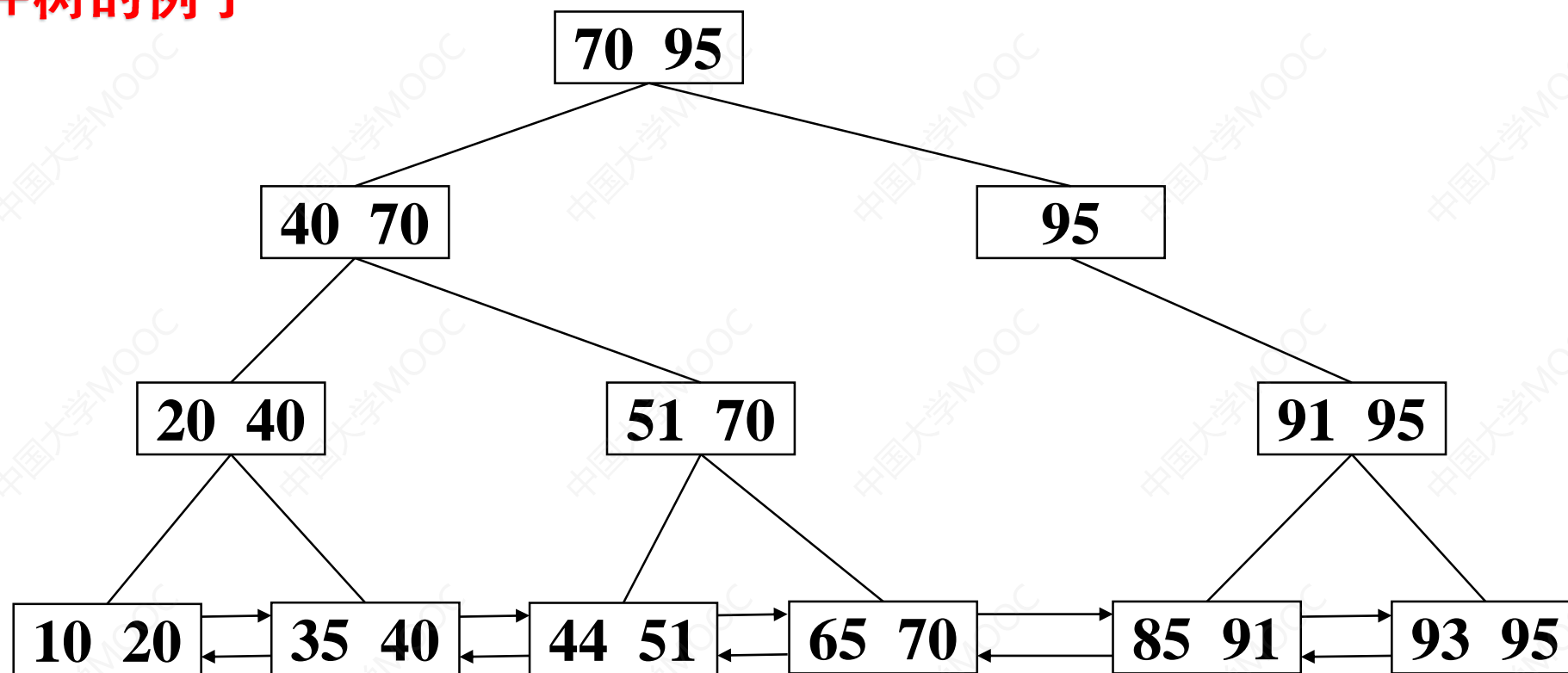
- B树
- B+树

- 是B树的一种变形
- 在叶结点上存储信息的树
  - 所有的关键码均出现在叶结点上
  - 各层结点中的关键码均是下一层相应结点中最大关键码（或最小关键码）的复写

**m阶B+树的结构定义如下：**

- (1)每个结点至多有m个子结点；**
- (2)每个结点(除根外)至少有 $\lceil m/2 \rceil$ 个子结点；**
- (3)根结点至少有两个子结点；**
- (4)有k个子结点的结点必有k个关键码。**

## 2阶B+树的例子



## B+树的查找

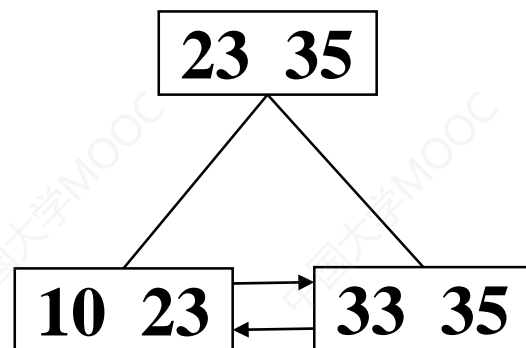
- 查找应该到叶结点层
  - 在上层已找到待查的关键码，并不停止
  - 而是继续沿指针向下一直查到叶结点层的这个关键码
- B+树的叶结点一般链接起来，形成一个双链表
  - 适合顺序检索（范围检索）
  - 实际应用更广

## B+树的插入

- 插入——分裂
  - 过程和B树类似
  - 注意保证上一层结点中有这两个结点的最大关键码（最小关键码）

# B+树

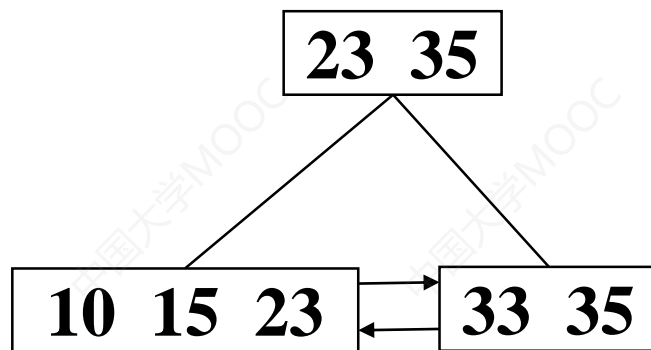
## 3阶B+树插入



插入15

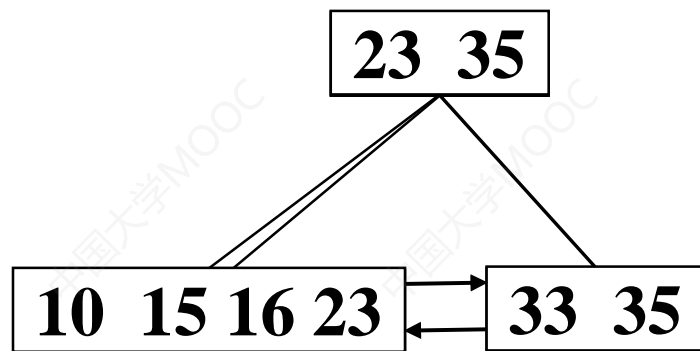


## 3阶B+树插入



插入15后

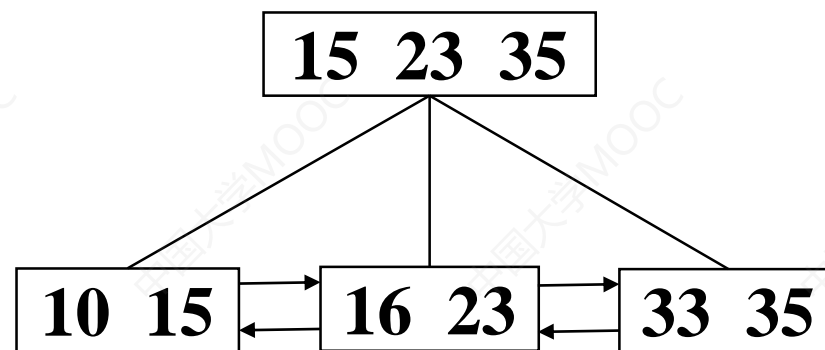
## 3阶B+树插入



插入16

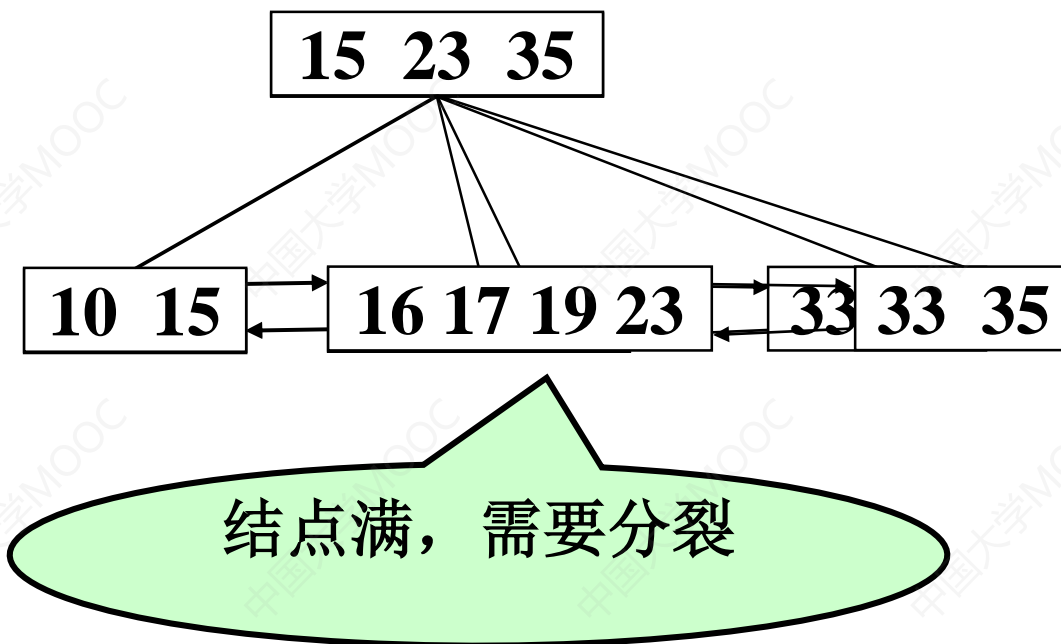
结点满，需要分裂

## 3阶B+树插入

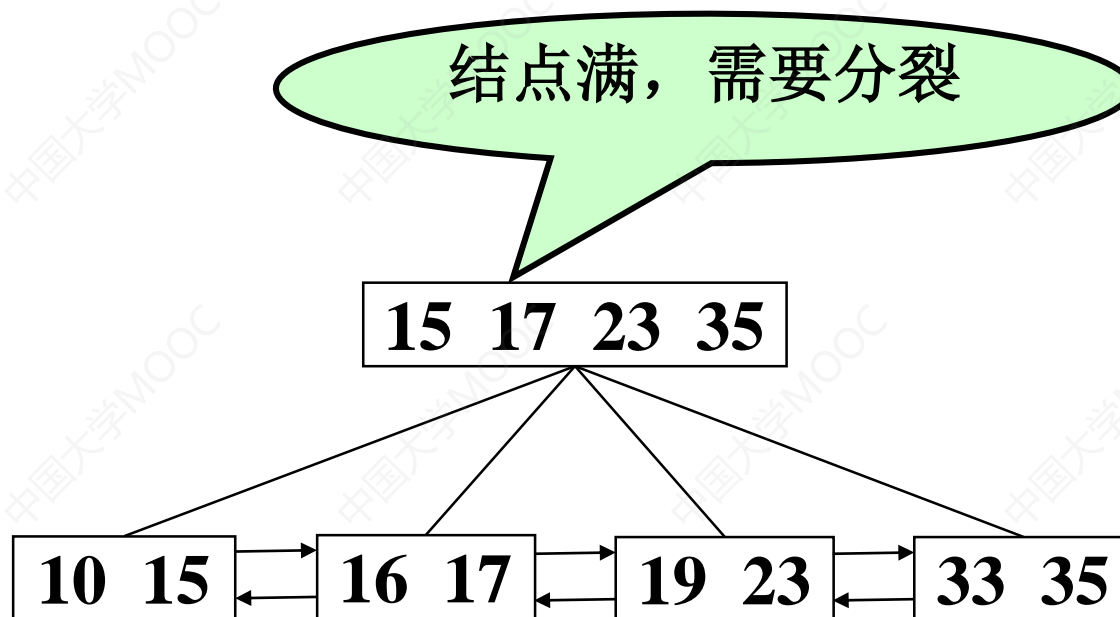


插入17

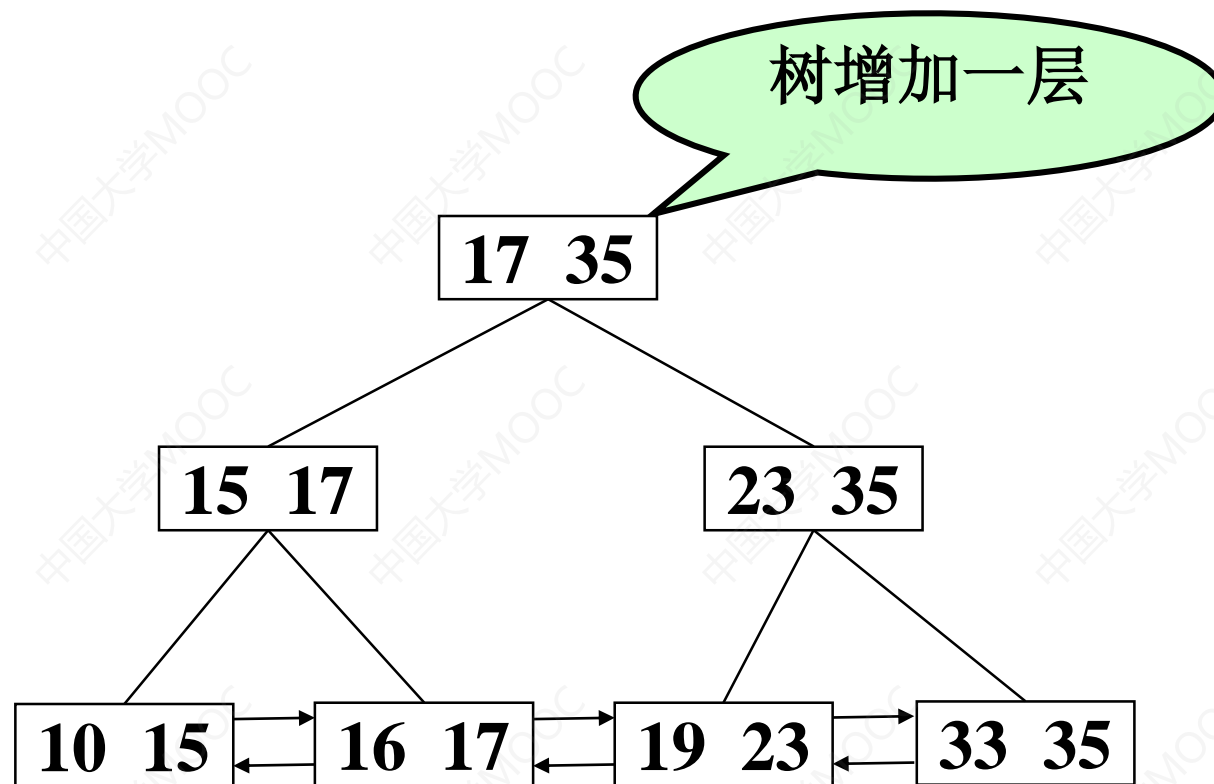
## 3阶B+树插入



## 3阶B+树插入



## 3阶B+树插入

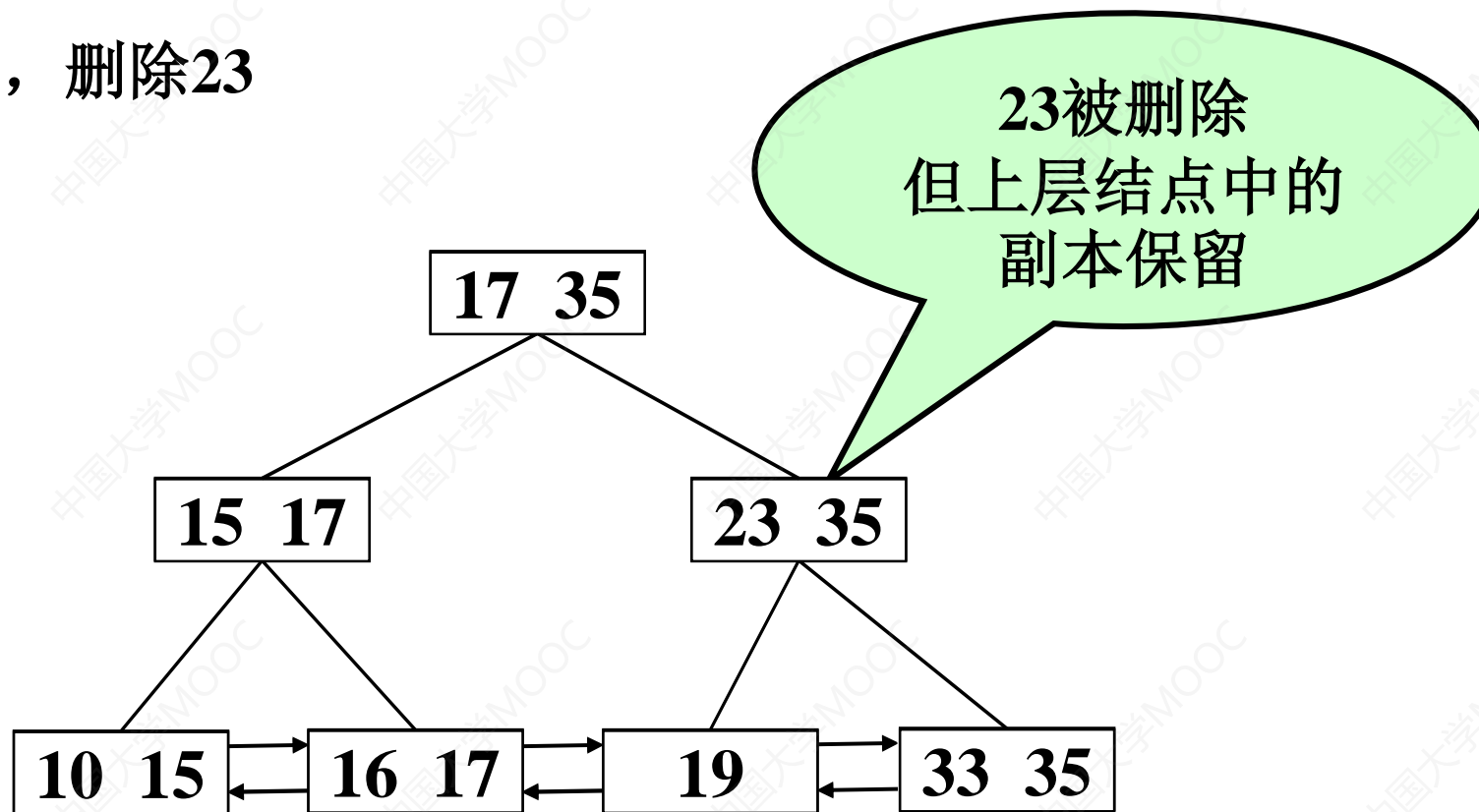


## B+树的删除

- 当关键码不满时，与左右兄弟进行调整、合并的处理和B树类似
- 关键码在叶结点层删除后，其在上层的复本可以保留，做为一个“分界关键码”存在
- 也可以替换为新的最大关键码（或最小关键码）

## B+树的删除

阶 $m=3$ ，删除23





# 动态查找

大连理工大学

于 红