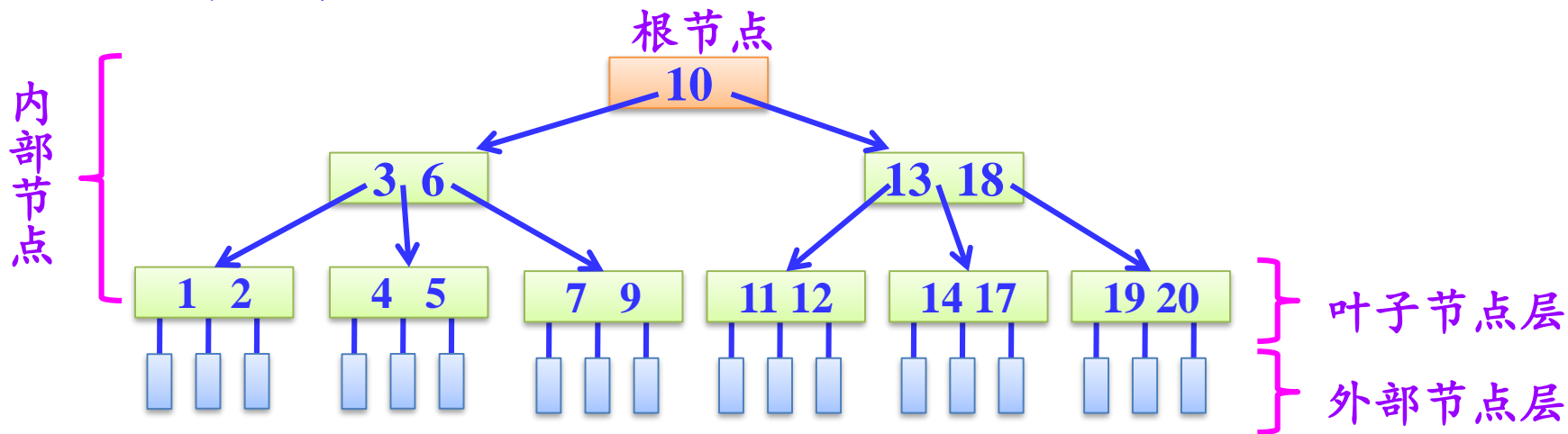


9.3.3 B-树

B-树又称为多路平衡查找树，是一种组织和维护外存文件系统非常有效的数据结构。

1、B-树的定义

一棵3阶B-树：

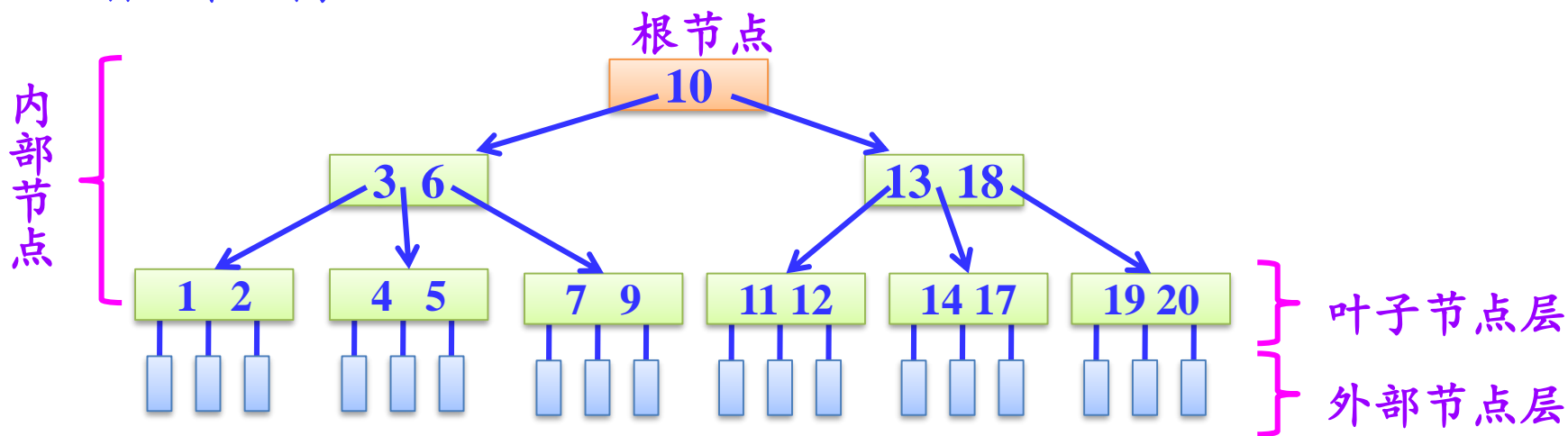


一棵 m 阶B-树或者是一棵空树，或者是满足要求的 m 叉树：

① 树中每个节点至多有 m 个孩子节点（即至多有 $m-1$ 个关键字）

最多关键字个数 $\text{Max} = m-1$

一棵3阶B-树:

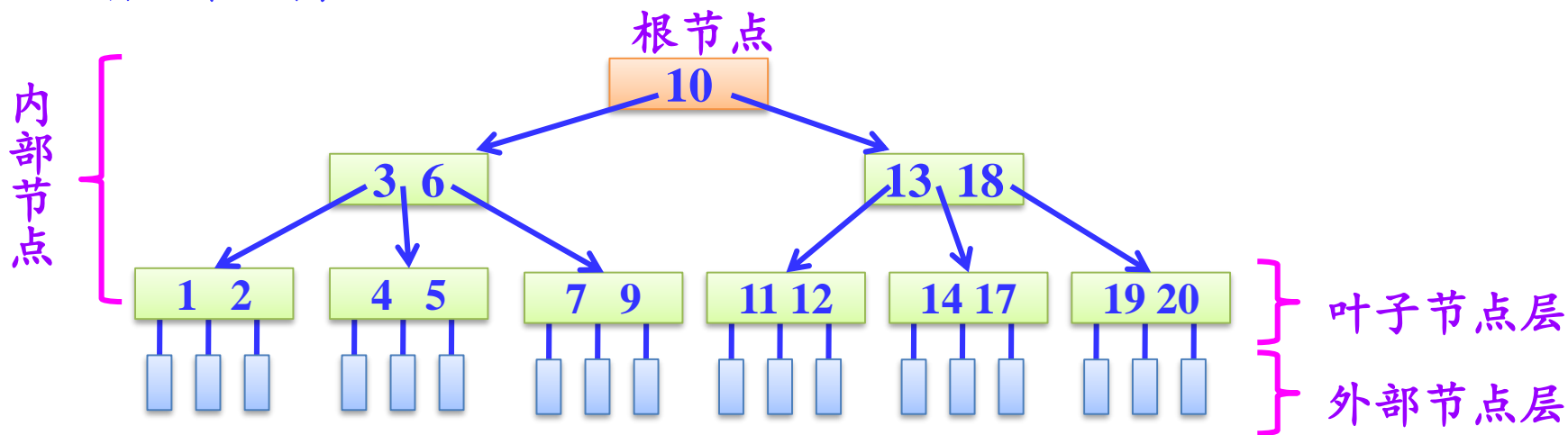


一棵 m 阶B-树或者是一棵空树，或者是满足要求的 m 叉树：

- ② 除根节点外，其他非叶子节点至少有 $\lceil m/2 \rceil$ 个孩子节点（即至少有 $\lceil m/2 \rceil - 1 = \lfloor (m-1)/2 \rfloor$ 个关键字）；

$$\text{最少关键字个数} \text{Min} = \lceil m/2 \rceil - 1$$

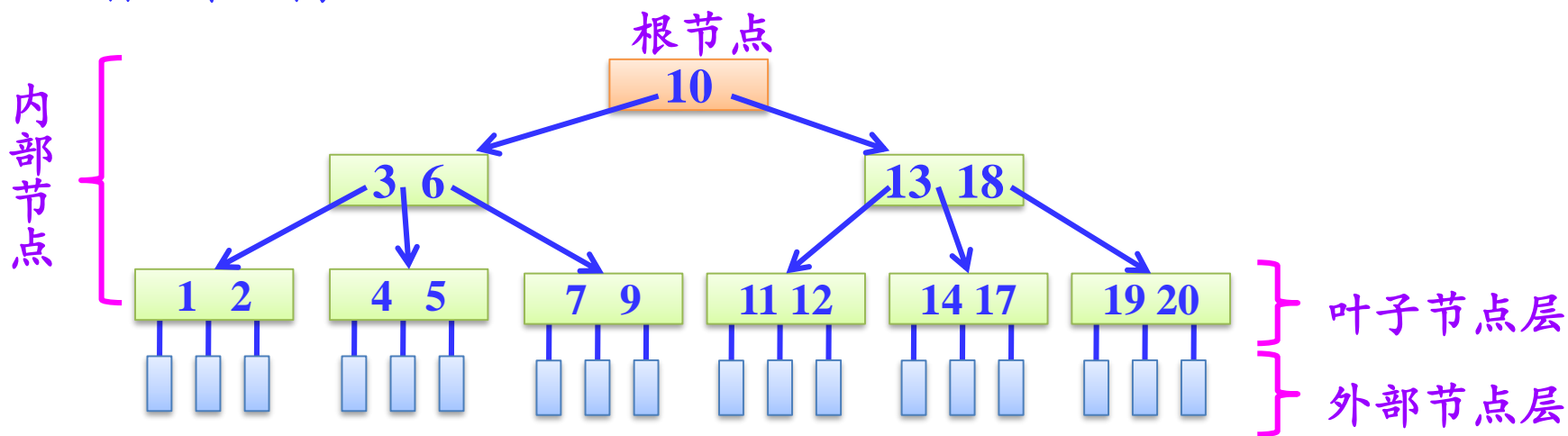
一棵3阶B-树:



一棵 m 阶B-树或者是一棵空树, 或者是满足要求的 m 叉树:

- ③ 若根节点不是叶子节点, 则根节点至少有两个孩子节点;

一棵3阶B-树:

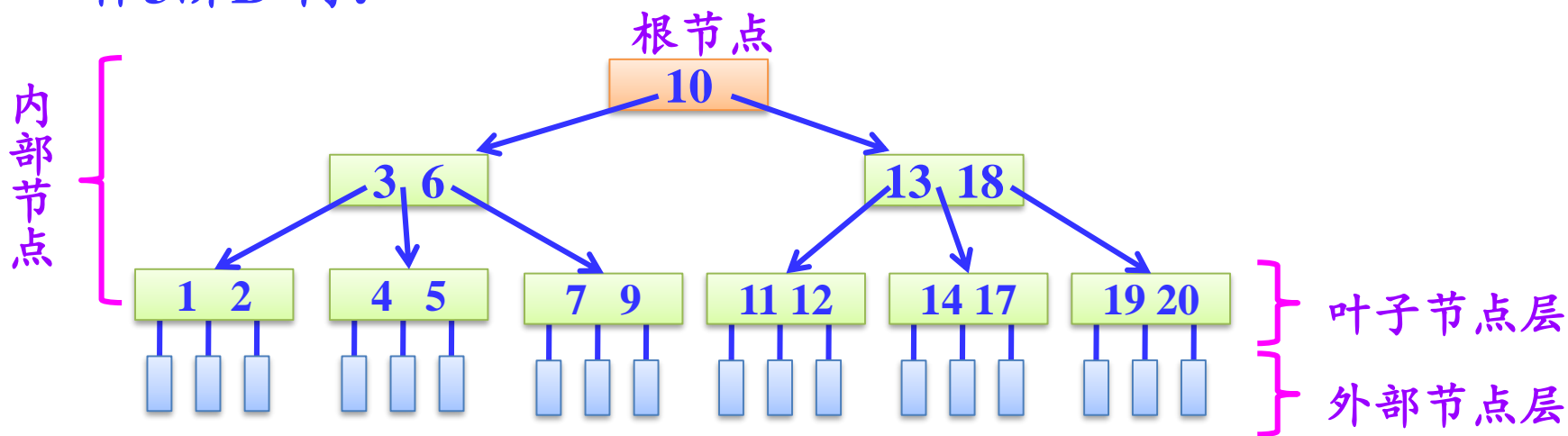


一棵 m 阶B-树或者是一棵空树，或者是满足要求的 m 叉树:

④ 每个节点的结构如下，节点中按关键字大小顺序排列:

n	p_0	k_1	p_1	k_2	p_2	\dots	k_n	p_n
-----	-------	-------	-------	-------	-------	---------	-------	-------

一棵3阶B-树:

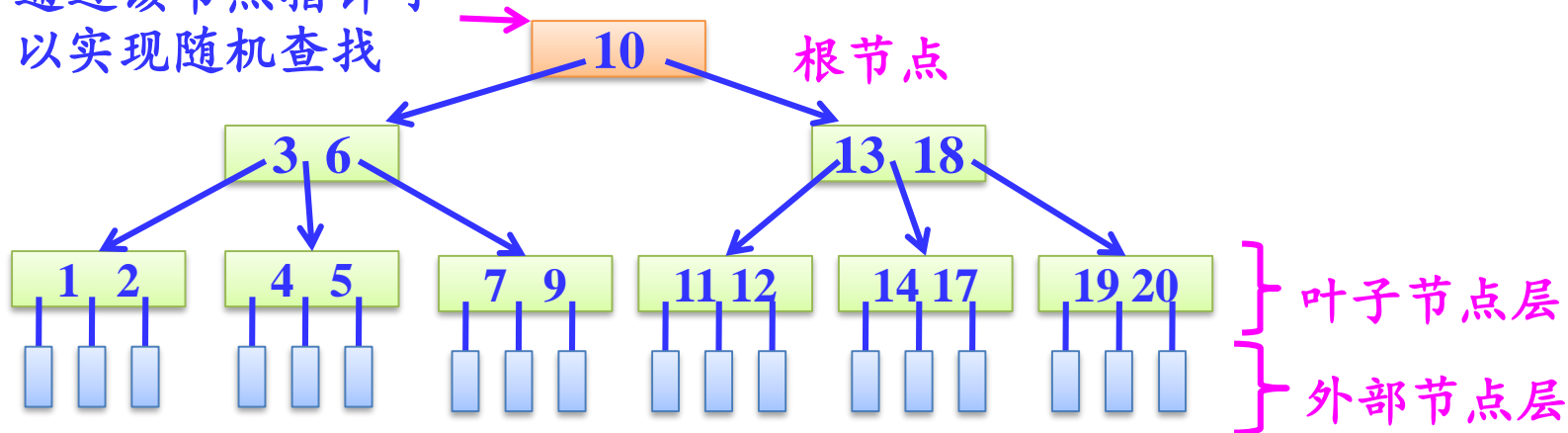


一棵 m 阶B-树或者是一棵空树，或者是满足要求的 m 叉树：

- ⑤ 所有外部节点都在同一层上。B-树是所有节点的平衡因子均等于0的多路查找树。

在计算B-树的高度时，需要计入最底层的外部节点

通过该节点指针可
以实现随机查找



一棵3阶B-树

非根非外部节点的关键字个数：1~2。

非根非外部节点的孩子节点个数：2~3。

说明：外部节点就是失败节点，指向它的指针为空，不含有任何信息，是虚设的。一棵B-树中总有 n 个关键字，则外部节点个数为 $n+1$ 。

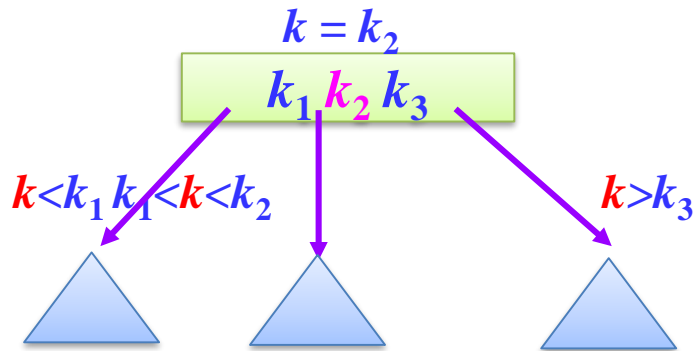
在B-树的存储结构中，节点的类型定义如下：

```
#define MAXM 10                                //定义B-树的最大的阶数
typedef int KeyType;                            //KeyType为关键字类型
typedef struct node
{
    int keynum;                                //节点当前拥有的关键字的个数
    KeyType key[MAXM];                        //[1..keynum]存放关键字
    struct node *parent;                      //双亲节点指针
    struct node *ptr[MAXM];                  //孩子节点指针数组[0..keynum]
} BTreeNode;
```


2、B-树的查找

将 k 与根节点中的 $\text{key}[i]$ 进行比较：

- ① 若 $k = \text{key}[i]$ ，则查找成功；
- ② 若 $k < \text{key}[1]$ ，则沿着指针 $\text{ptr}[0]$ 所指的子树继续查找；
- ③ 若 $\text{key}[i] < k < \text{key}[i+1]$ ，则沿着指针 $\text{ptr}[i]$ 所指的子树继续查找；
- ④ 若 $k > \text{key}[n]$ ，则沿着指针 $\text{ptr}[n]$ 所指的子树继续查找。



说明：当查找到某个叶节点时，若相应的指针为空，落入一个外部节点，表示查找失败。

3、B-树的插入

将关键字 k 插入到B-树的过程分两步完成：

(1) 查找该关键字的插入节点（注意B-树的插入节点一定是叶子节点层的节点）。

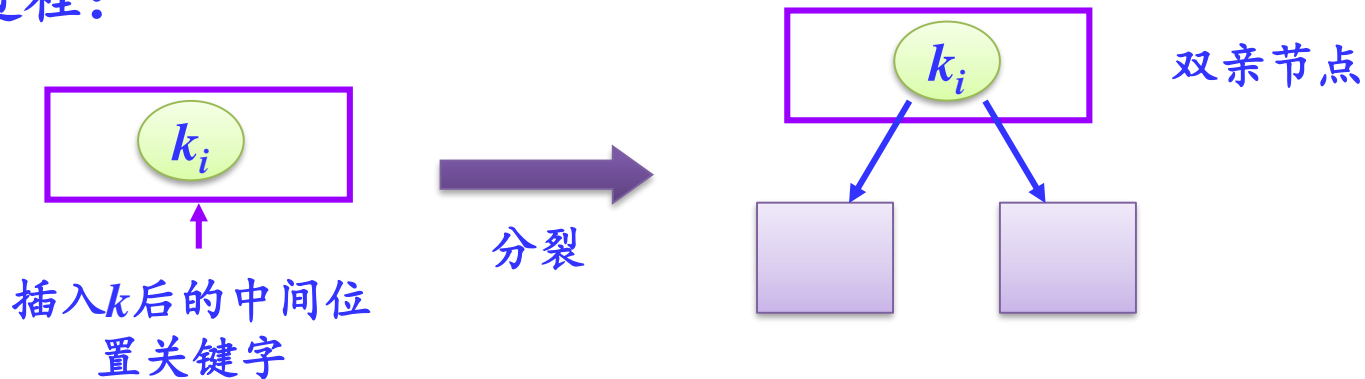
(2) 插入关键字。

在某个叶子节点中插入关键字分两种情况

① 插入节点有空位置，即关键字个数 $n < m-1$ ：直接把关键字 k 有序插入到该节点的合适位置上。

② 插入节点没有空位置，即原关键字个数 $n=m-1 \Rightarrow$ 分裂。

分裂过程：



- 如果没有双亲节点，新建一个双亲节点，树的高度增加一层。
- 如果有双亲节点，将 k_i 插入到双亲节点中。

【例9-4】 关键字序列为：

{1,2,6,7,11,4,8,13,10,5,17,9,16,20,3,12,14,18,19,15}。

创建一棵5阶B-树。

注意： 最多的关键字个数 $\text{Max} = m - 1 = 4$

Max=4

关键字序列

1
↑

2
↑

6
↑

7
↑

1 2 6 7

Max=4

关键字序列

11
↑

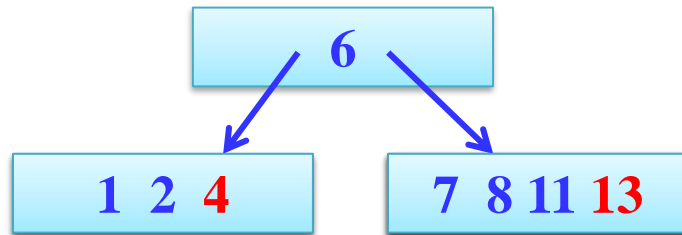
4
↑

8
↑

13
↑

1 2 6 7 11

关键字个数>4



Max=4

关键字序列

10

5

17

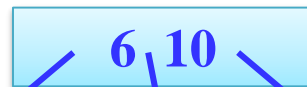
9

16



关键字个数>4

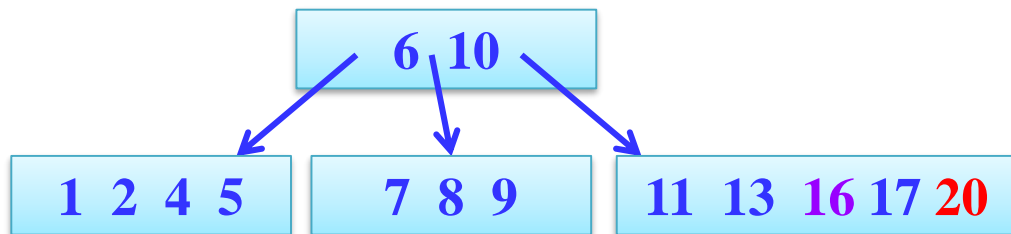
分裂



Max=4

关键字序列

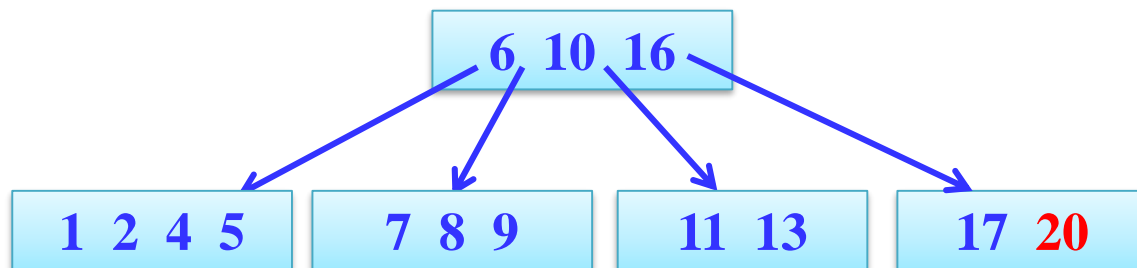
20



关键字个数>4



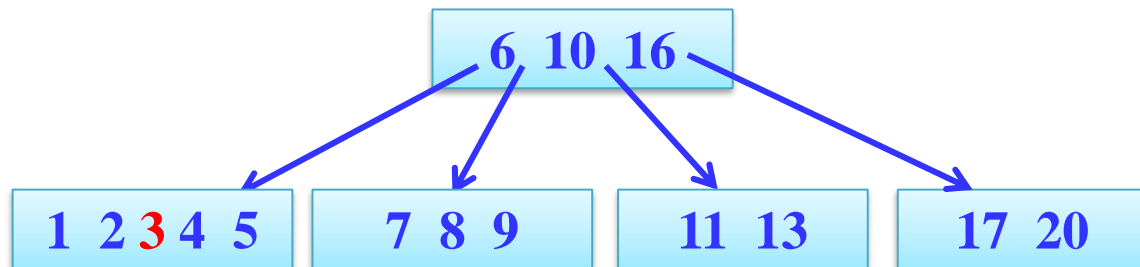
分裂



Max=4

关键字序列

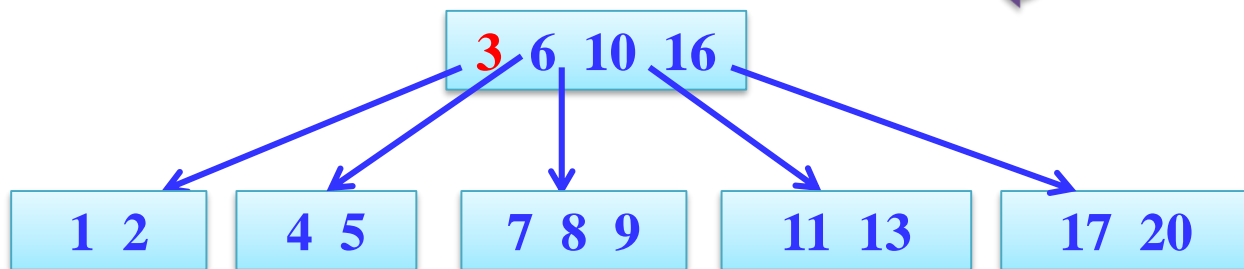
3



关键字个数>4

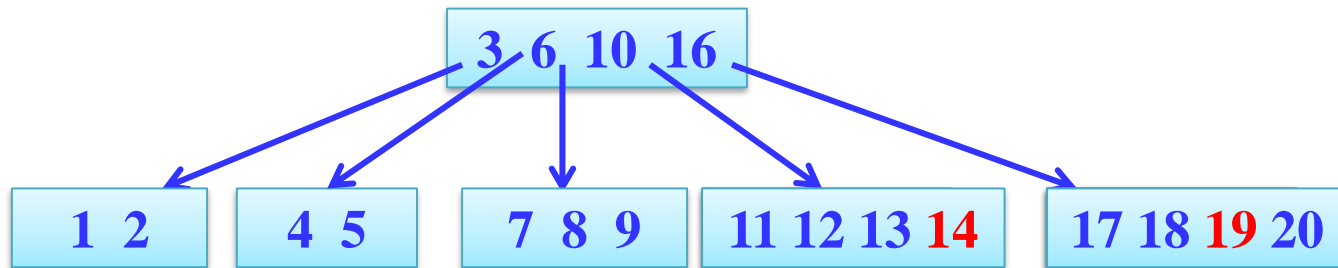


分裂



Max=4

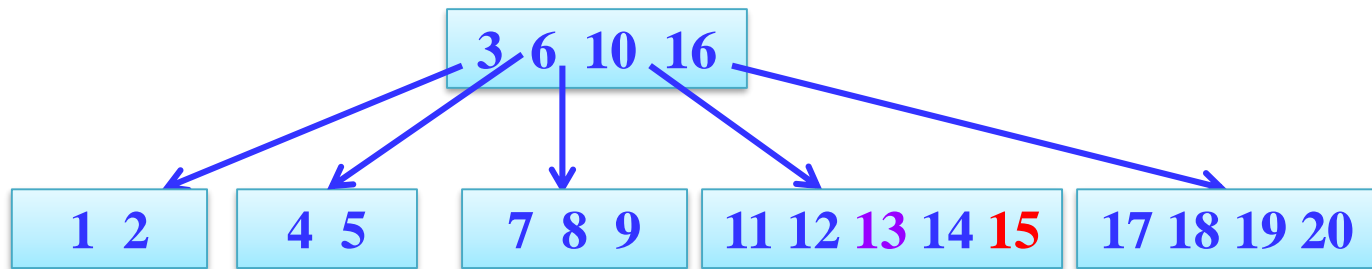
关键字序列 12 14 18 19
 ↑ ↑ ↑ ↑



Max=4

关键字序列

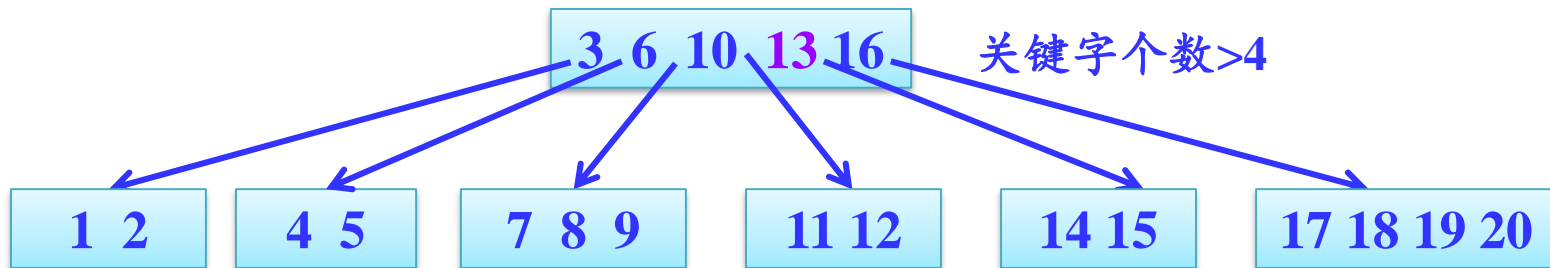
15



关键字个数>4

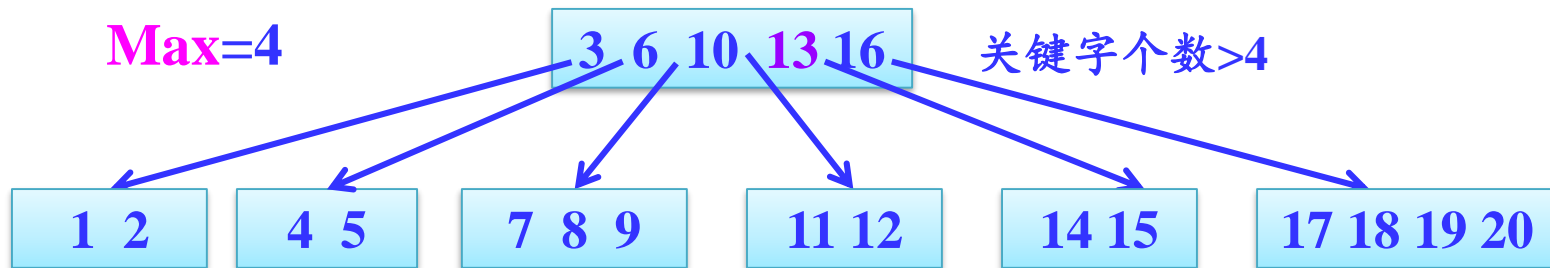


分裂

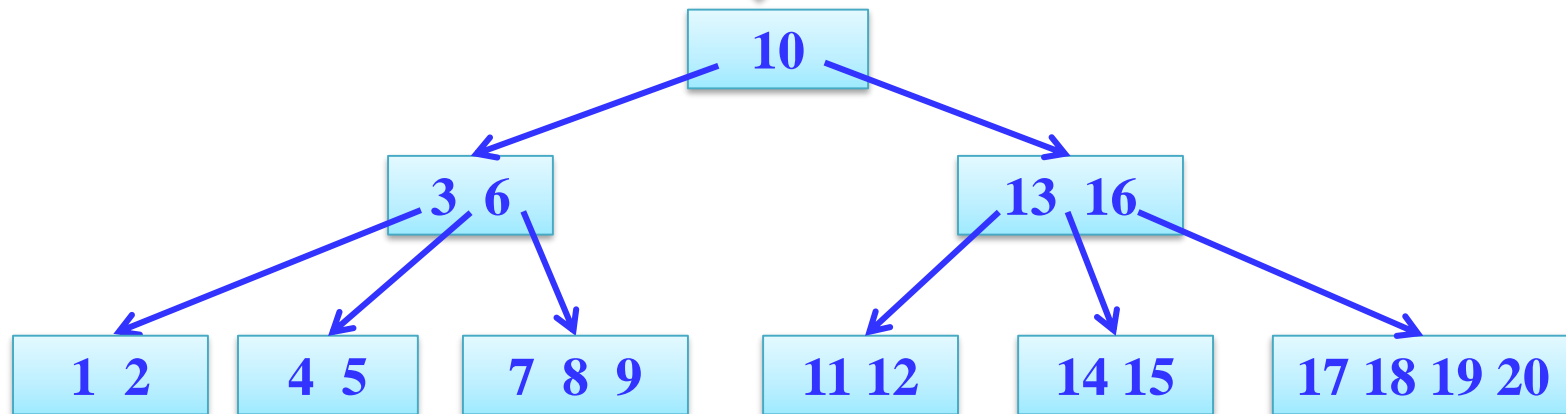


关键字个数>4

Max=4



继续分裂



B-树构建完毕



思考题

- ① 在B-树中每插入一个关键字，都要新建一个节点吗？
- ② 在B-树中插入一个关键字，若引起分裂，树高一定会升高一层吗？

4、B-树的删除

在B-树上删除关键字 k 的过程分两步完成：

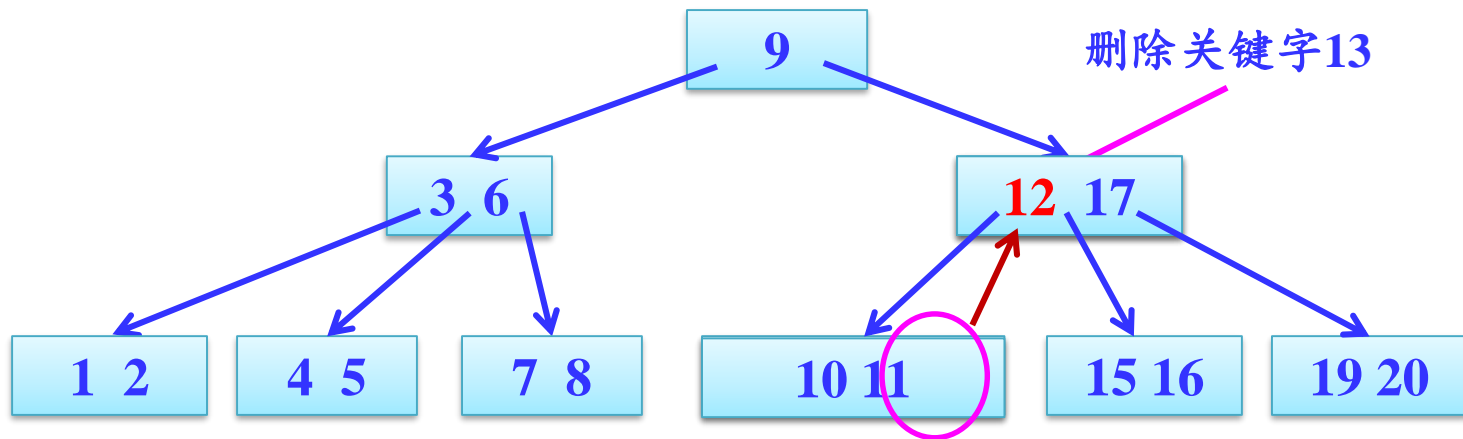
- (1) 查找关键字 k 所在的节点。
- (2) 删除关键字 k 。

删除关键字 k 分两种情况：

- 在叶子节点层上删除关键字 k 。
- 在**非**叶子节点层上删除关键字 k 。

注意：非根、非叶子节点的关键字最少个数 $\text{Min}=\lceil m/2 \rceil - 1$

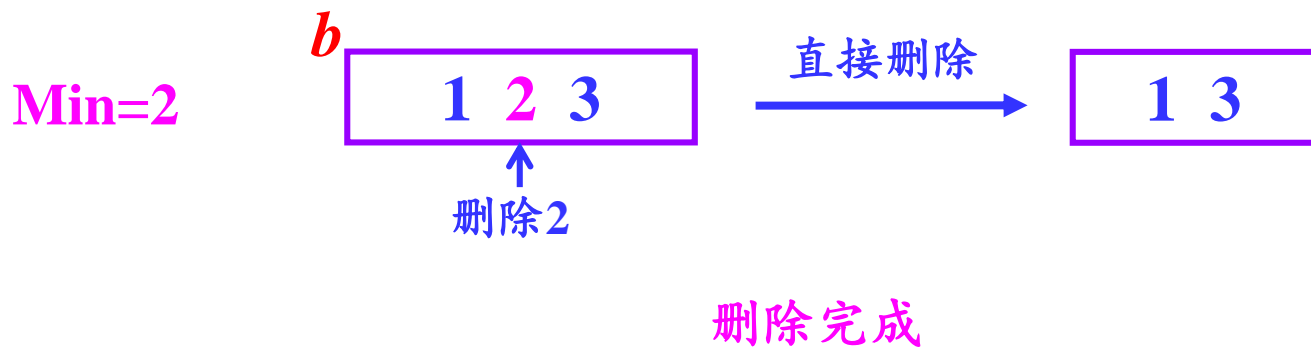
在非叶子节点上删除关键字 $k \Rightarrow$ 在叶子节点上删除关键字 k



删除叶子节点中的12

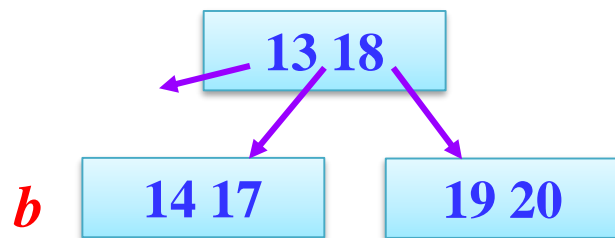
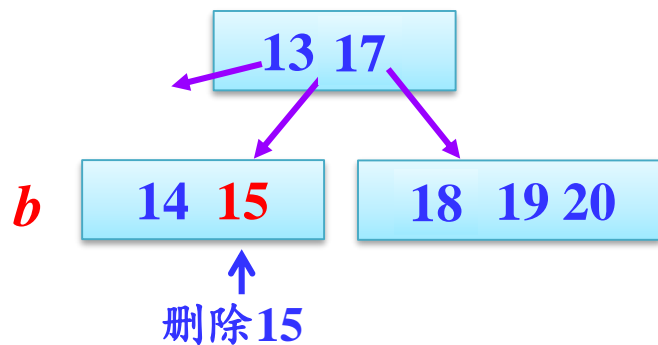
在B-树的叶子节点***b***上删除关键字共有以下3种情况：

① 假如***b***节点的关键字个数大于**Min**，说明删去该关键字后该节点仍满足B-树的定义，则可直接删去该关键字。



② 假如***b***节点的关键字个数等于Min，说明删去关键字后该节点将不满足B-树的定义。若可以从兄弟节点借。

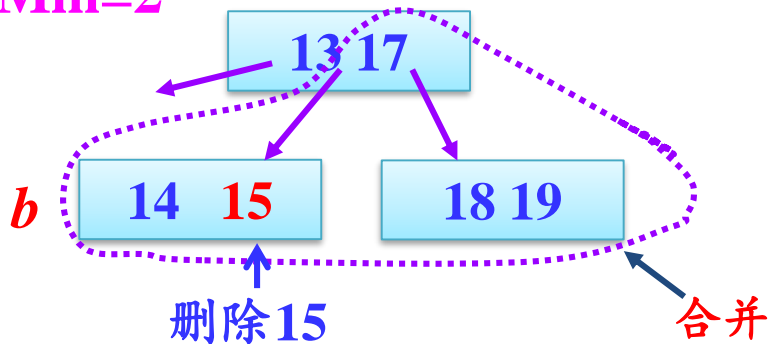
Min=2



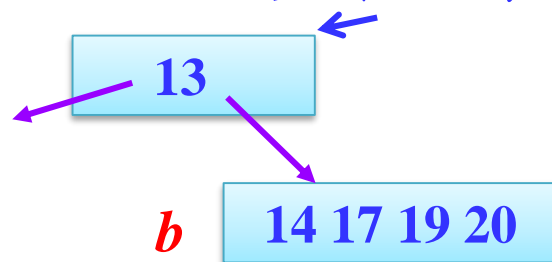
删除完成

③ 假如***b***节点的关键字个数等于Min，说明删去关键字后该节点将不满足B-树的定义。若不能从兄弟节点借。

Min=2



减少1个关键字

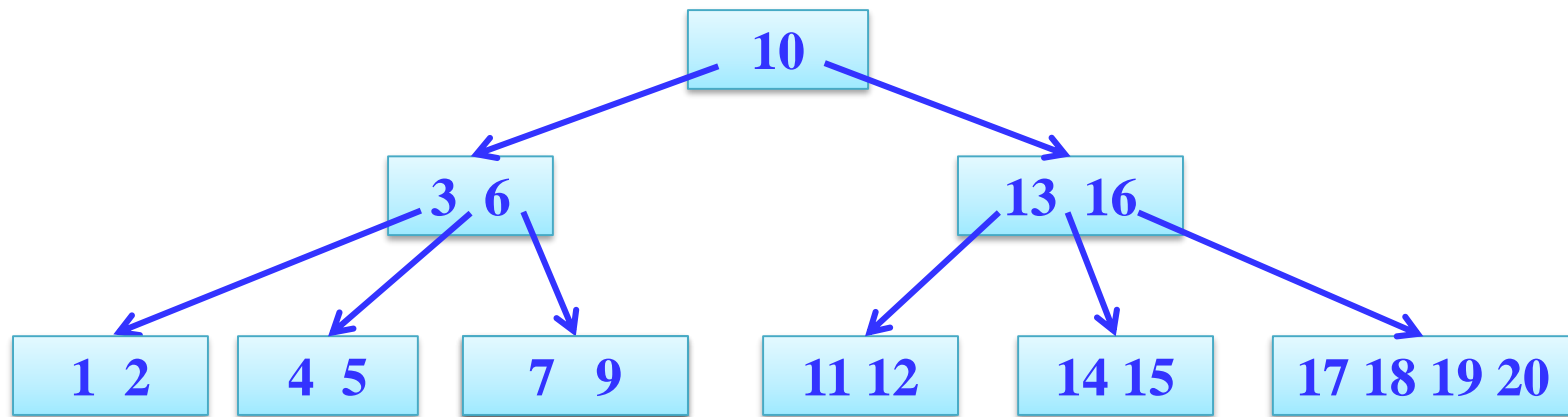


删除完成

【例9-5】 对于前例生成的B-树，给出删除8,16,15,4等4个关键字的过程。

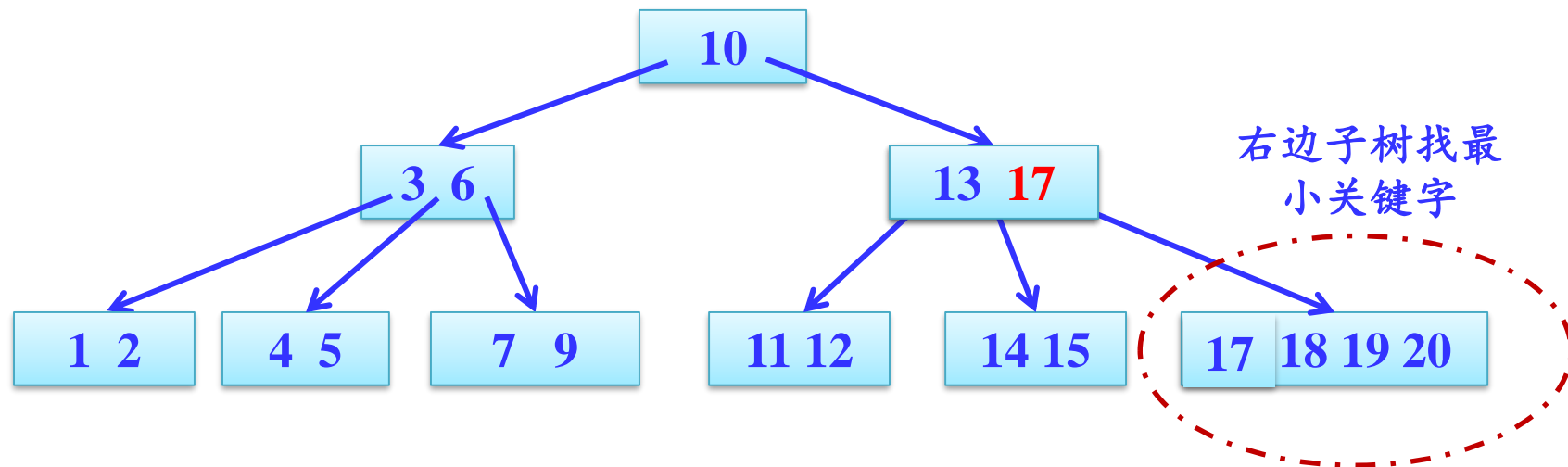
Min=2

删除： 8
↑



Min=2

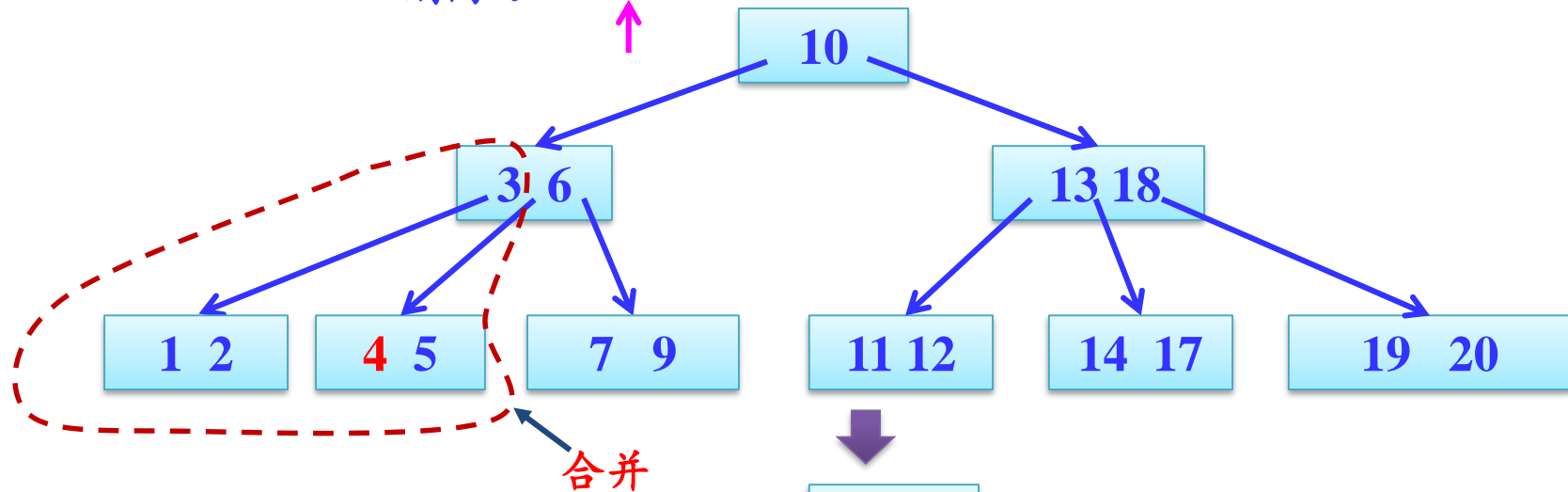
删除：16
↑



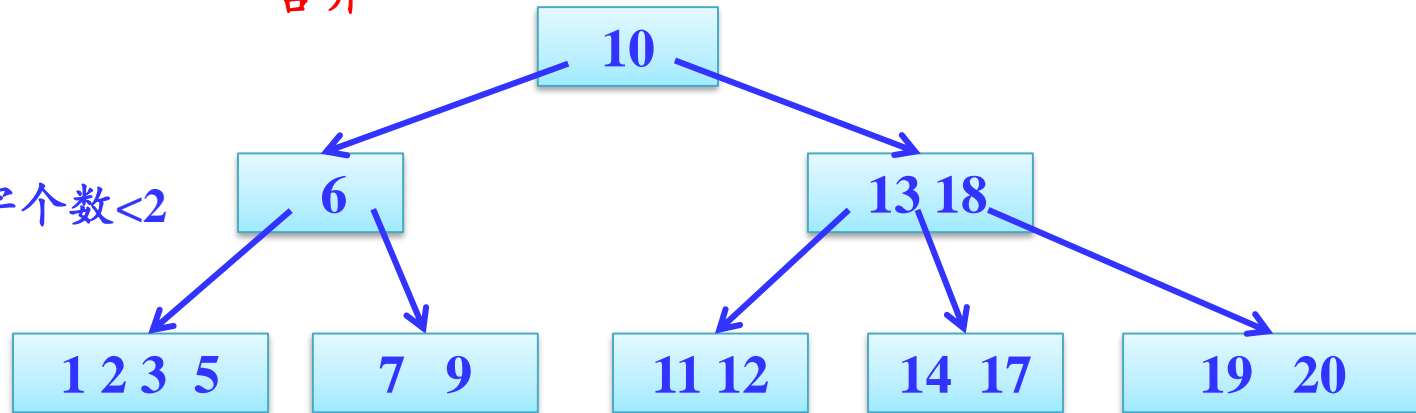
Min=2

删除:

4
↑

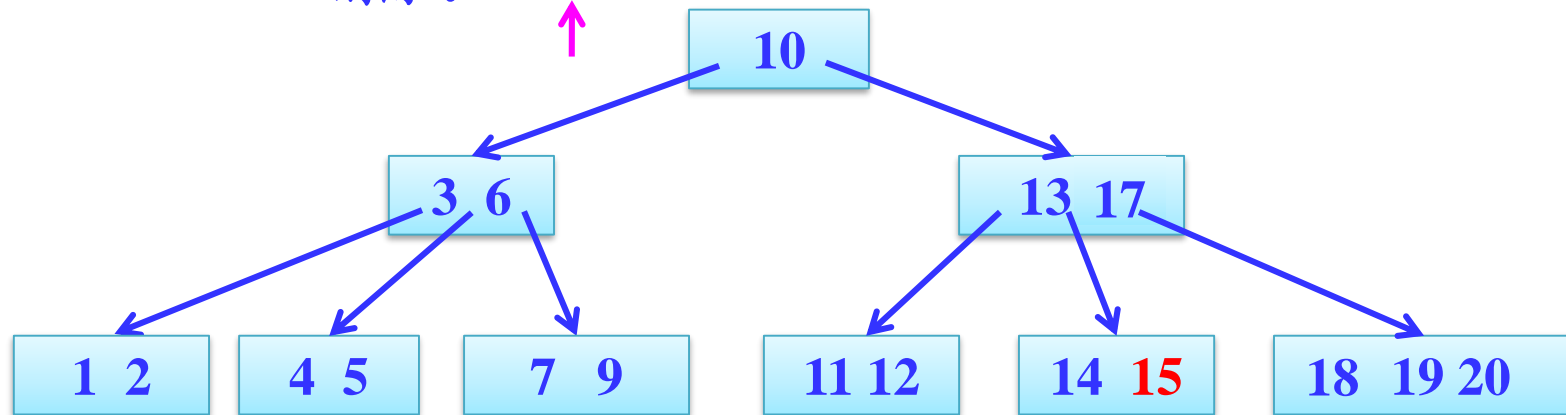


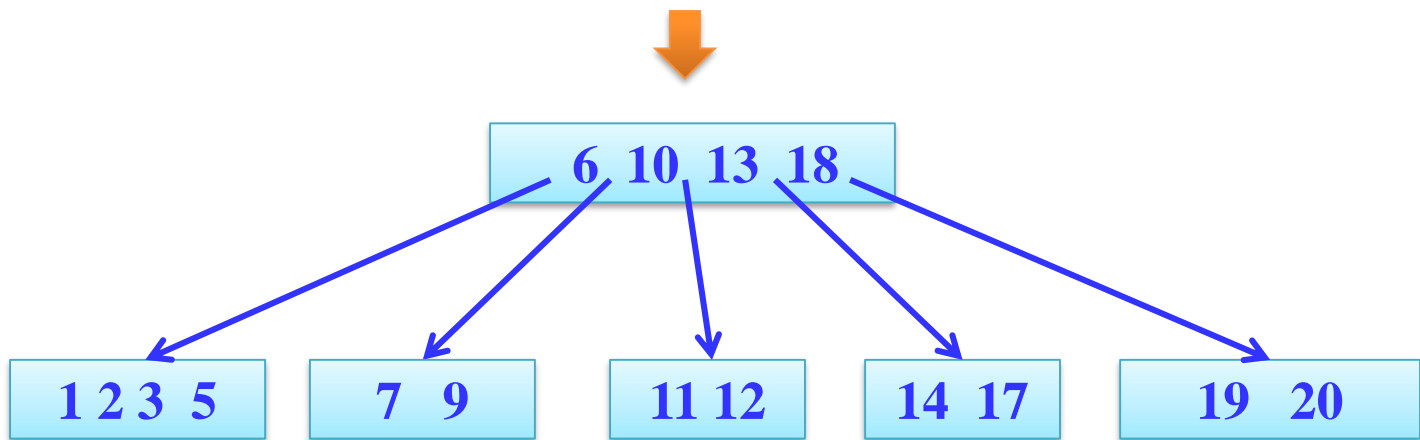
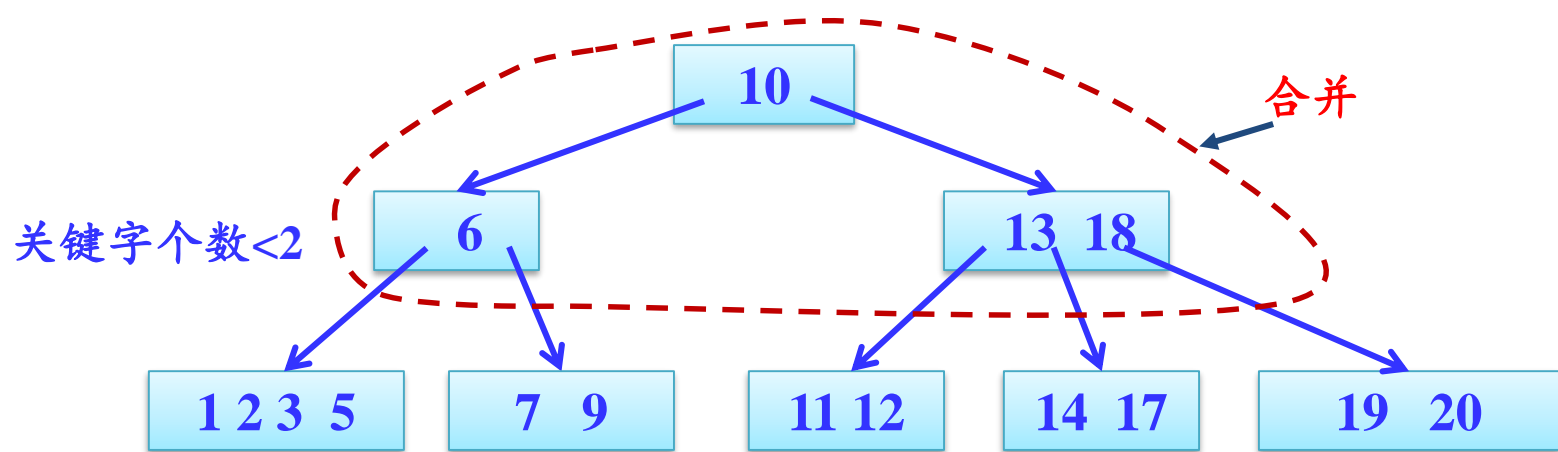
关键字个数<2



Min=2

删除: 15





删除完毕

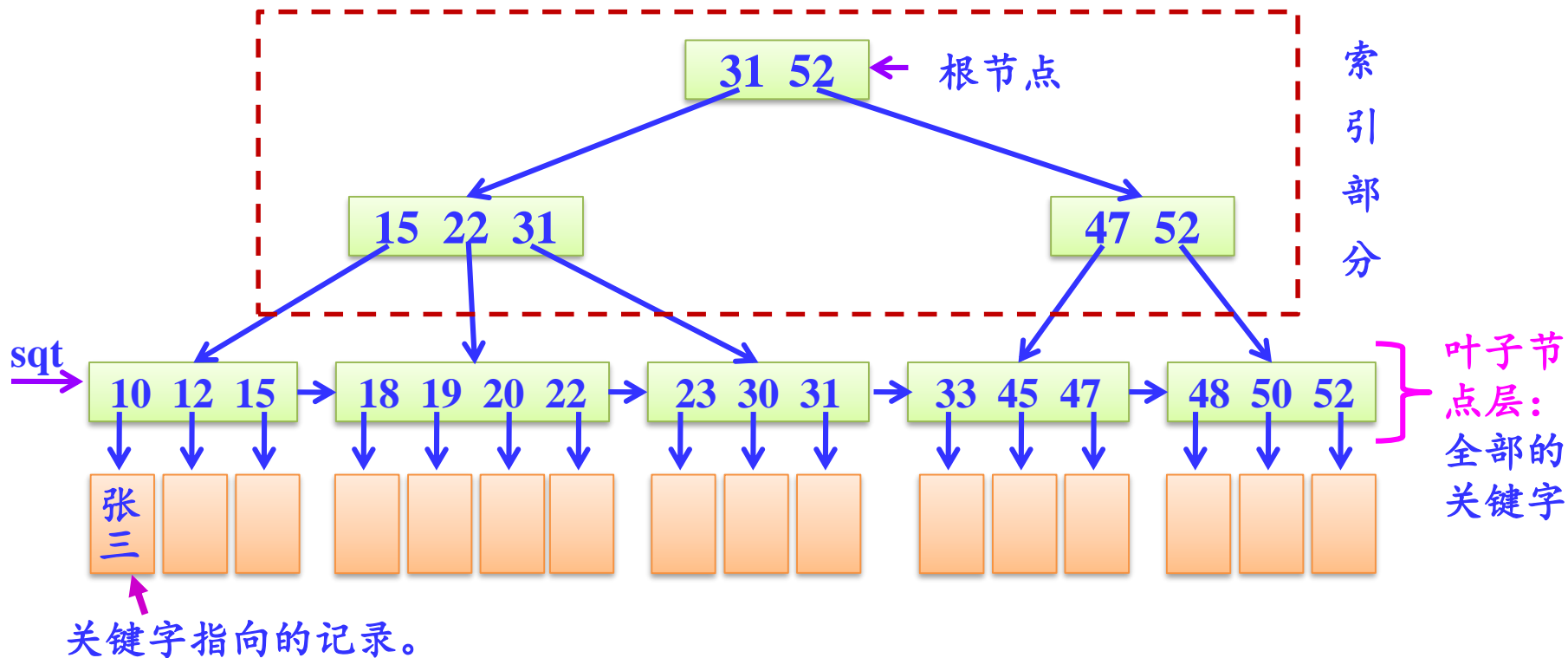


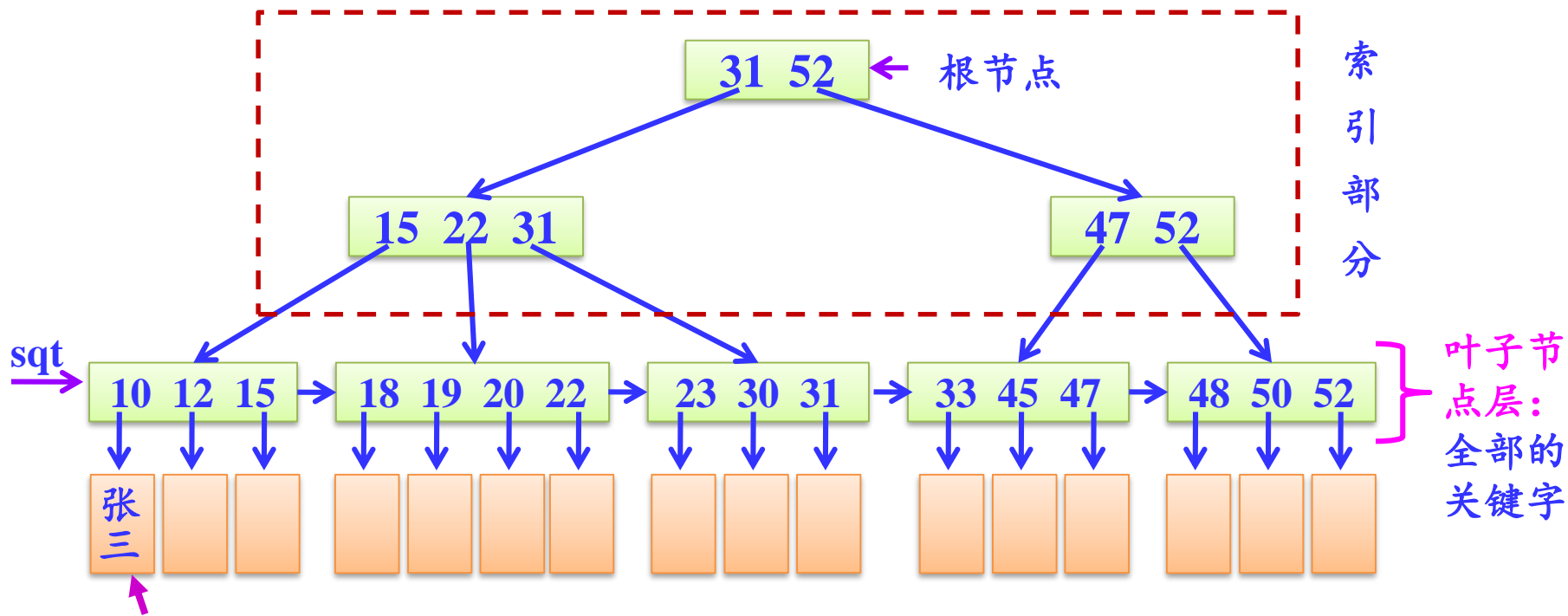
思考题

- ① 在B-树中每删除一个关键字，都要删除一个节点吗？
- ② 在B-树中删除一个关键字，若引起合并，树高一定会降低一层吗？

9.3.4 B+树

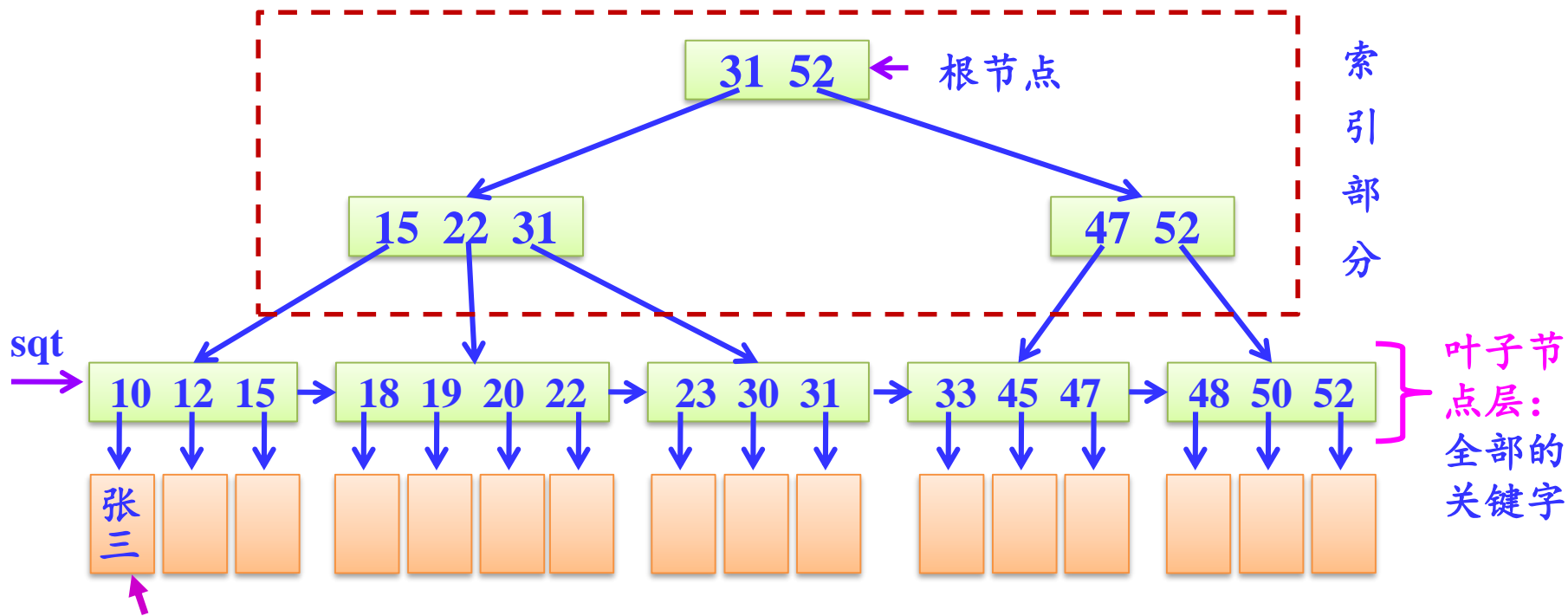
B+树是B-树的一些变形。一棵4阶的B+树示例：





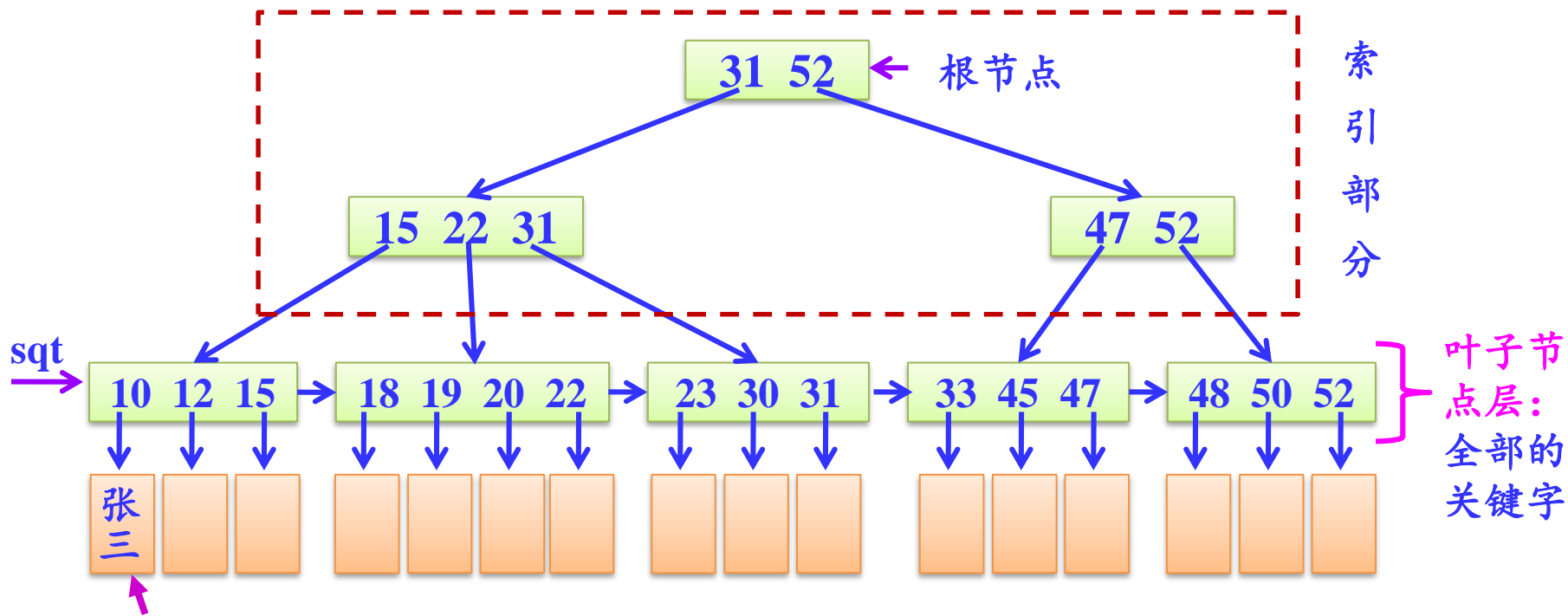
B+树的定义：一棵 m 阶B+树满足下列要求：

- ① 每个分支节点至多有 m 棵子树（这里 $m=4$ ）。



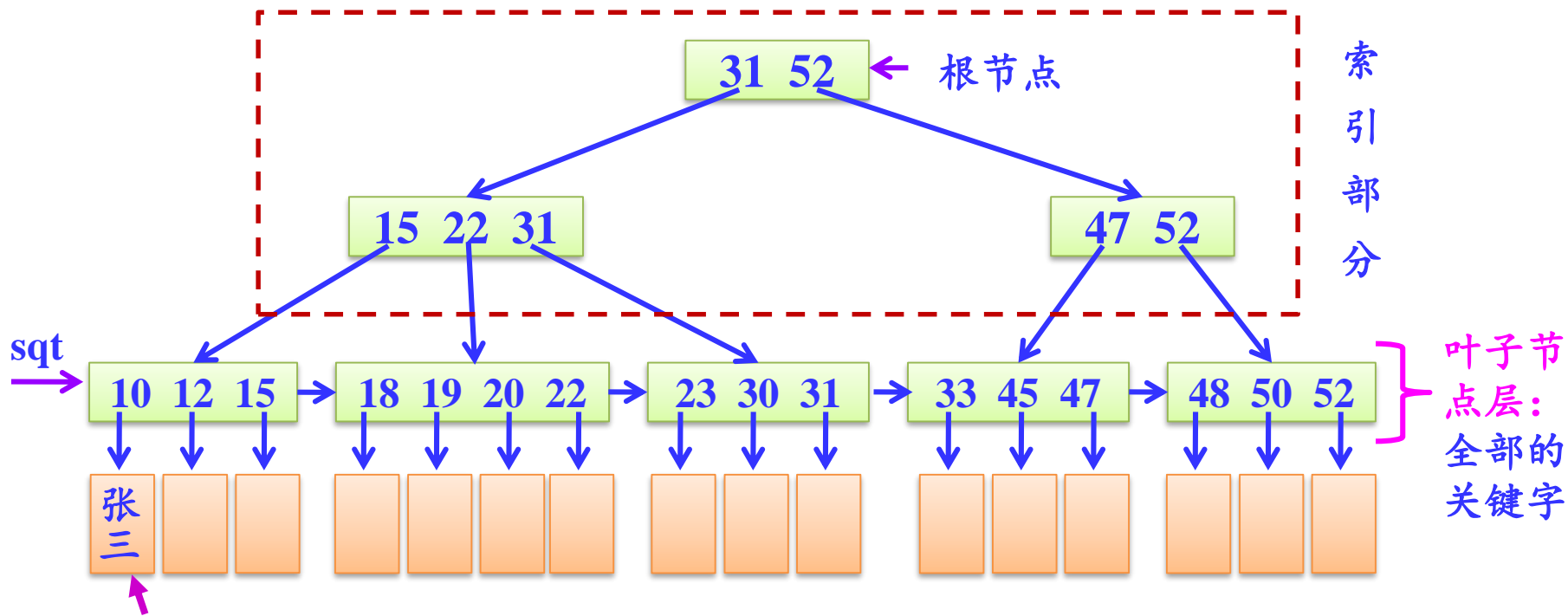
B+树的定义：一棵 m 阶B+树满足下列要求：

- ② 根节点或者没有子树，或者至少有两棵子树。



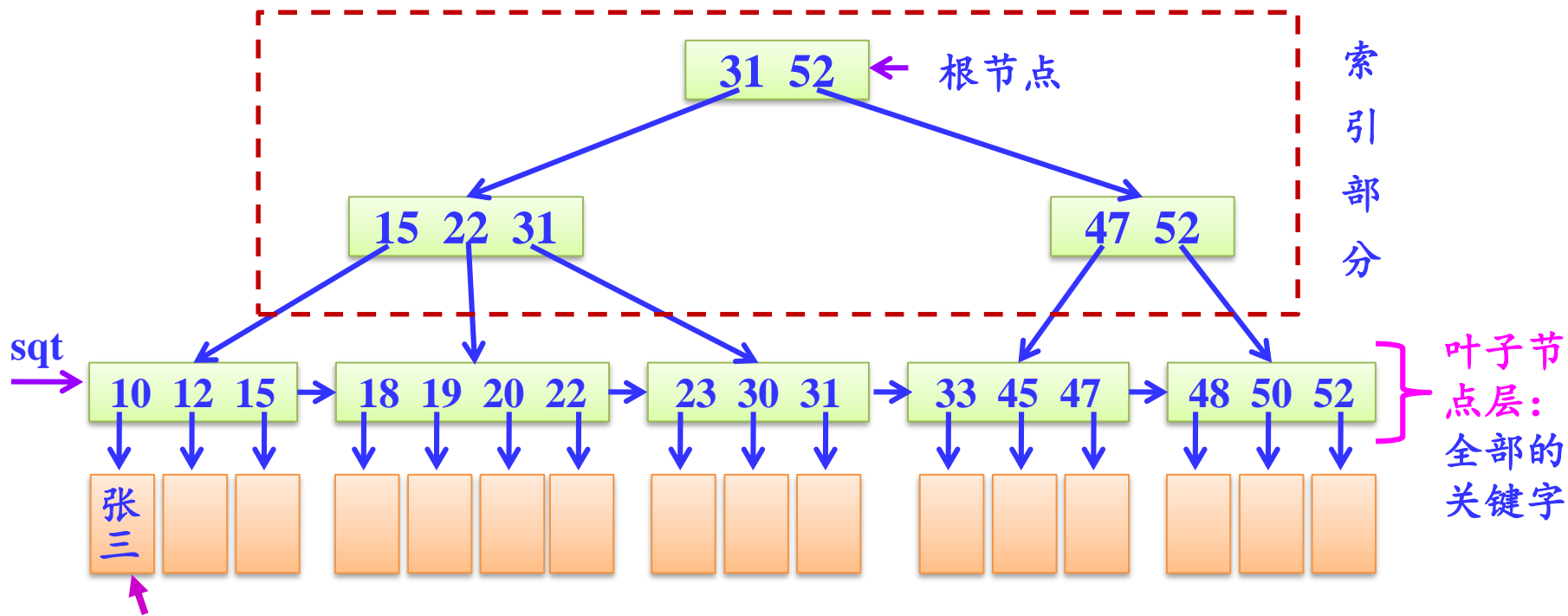
B+树的定义：一棵 m 阶B+树满足下列要求：

- ③ 除根节点外，其他每个分支节点至少有 $\lceil m/2 \rceil$ 棵子树。



B+树的定义：一棵 m 阶B+树满足下列要求：

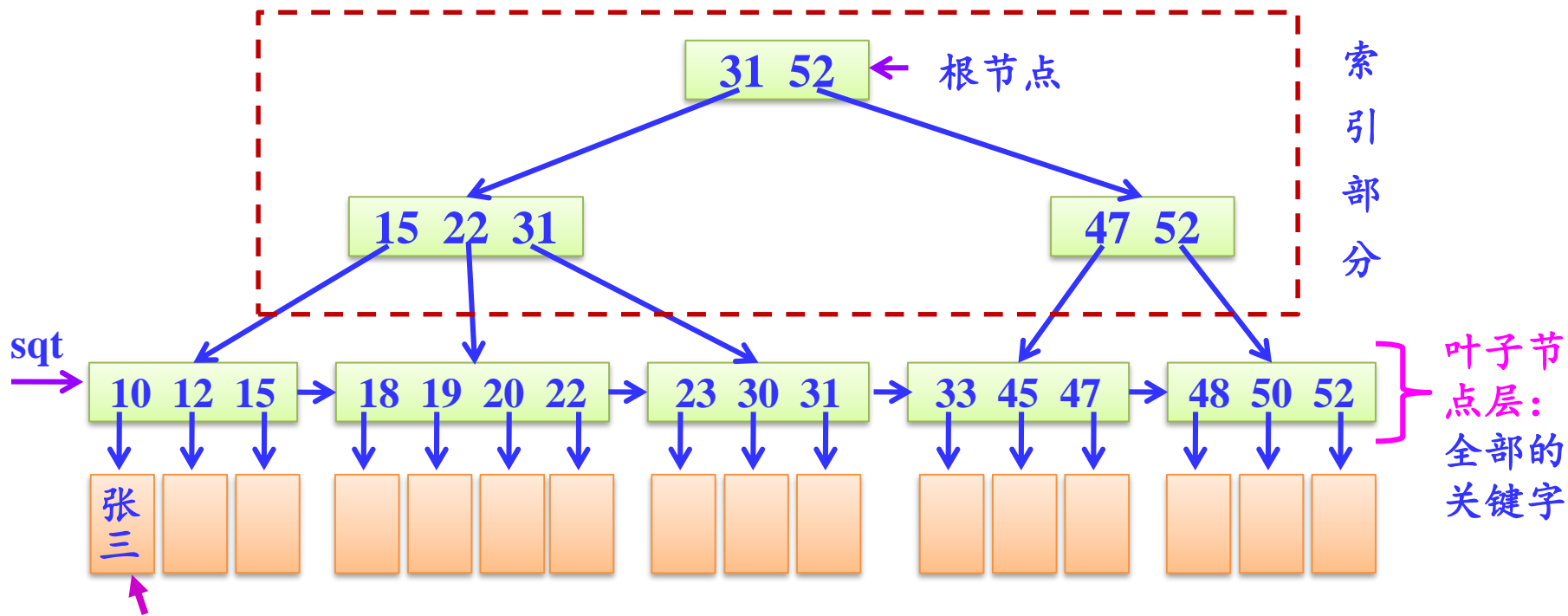
- ④ 有 n 棵子树的节点恰好有 n 个关键字。



关键字指向的记录。

B+树的定义：一棵 m 阶B+树满足下列要求：

- ⑤ 所有叶子节点包含全部关键字及指向相应记录的指针，而且叶子节点按关键字大小顺序链接。并将所有叶子节点链接起来。

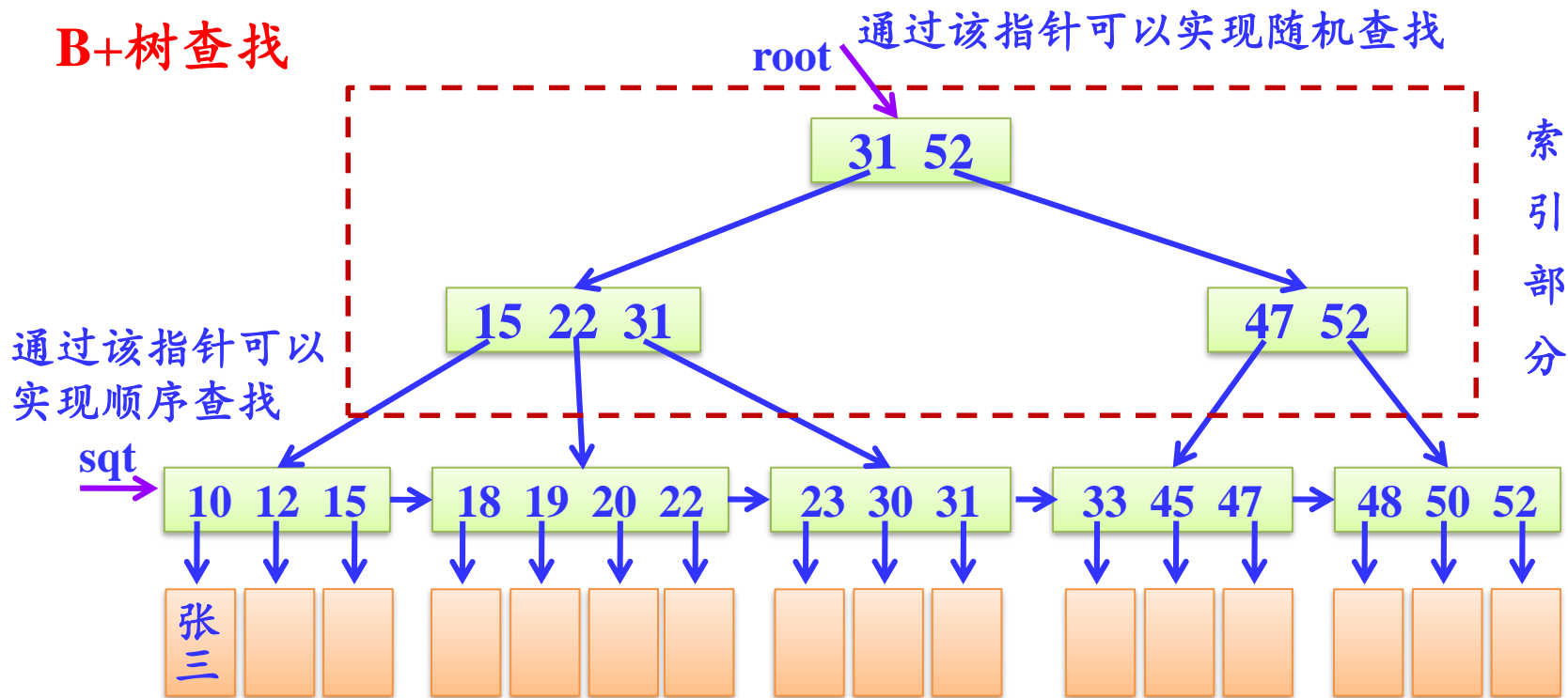


关键字指向的记录。

B+树的定义：一棵 m 阶B+树满足下列要求：

- ⑥ 所有分支节点（可看成是索引的索引）中仅包含它的各个子节点（即下级索引的索引块）中最大关键字及指向子节点的指针。

B+树查找



一棵4阶的B+树

思考题

m 阶的B+树和 m 阶的B-树的主要的差异？

——本讲完——