

# Assignment Four

1951976 李林飞

2021 年 6 月 14 日

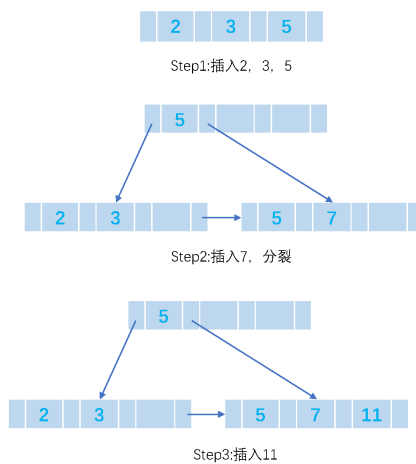
## 11.3

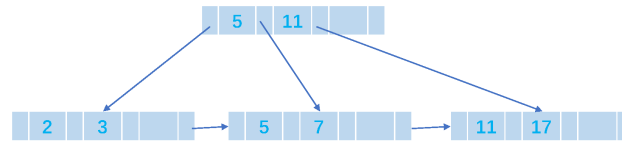
用下面的关键码值集合建立一个  $B^+$  树:

(2, 3, 5, 7, 11, 17, 19, 23, 29, 31)

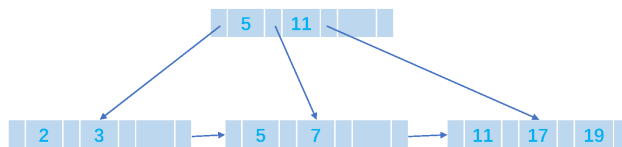
假设树初始为空, 值按上升顺序加入。根据一个结点所能容纳指针数的下列情况分别构造  $B^+$  树:

### a.4

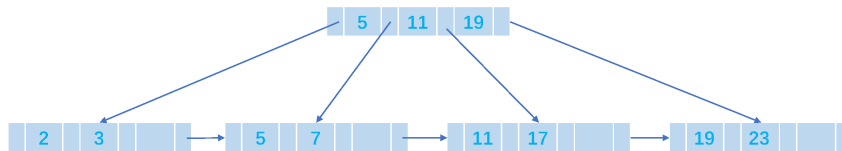




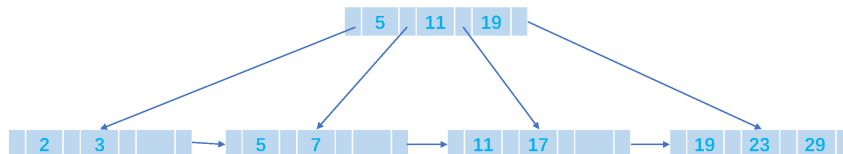
Step4:插入17, 分裂



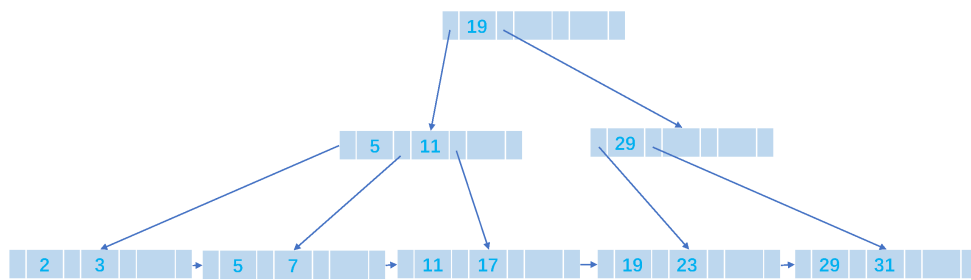
Step5:插入19



Step6:插入23, 分裂

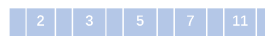


Step7:插入29

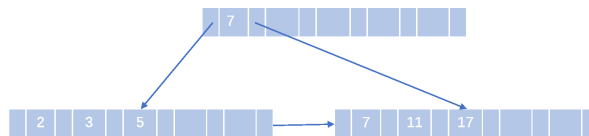


Step8:插入31, 叶结点分裂, 并自底向上递归构建

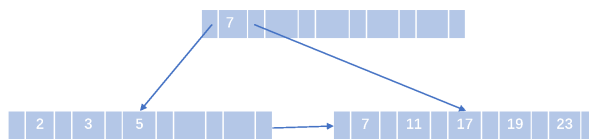
## b.6



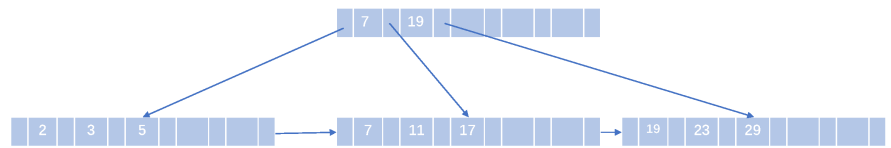
Step1:插入2, 3, 5, 7, 11



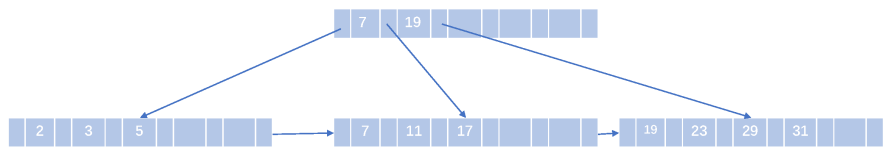
Step2:插入17, 分裂



Step3:插入19, 23



Step4:插入29, 分裂

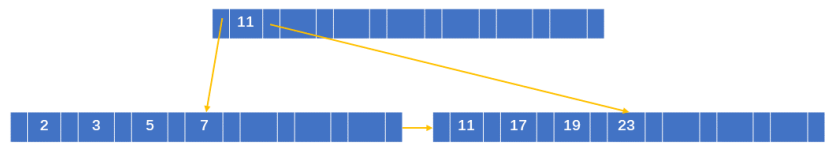


Step5:插入31

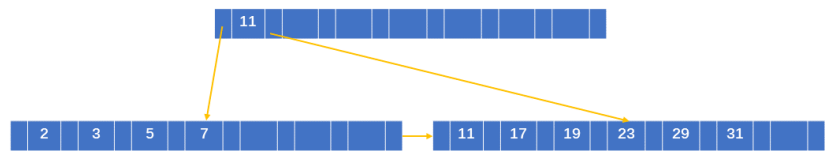
## c.8



Step1:插入2, 3, 5, 7, 11, 17, 19



Step2:插入23, 分裂



Step3:插入29, 31

## 11.12

如果索引项按照已排好的顺序插入， $B^+$  树的每个叶结点的充满情况如何？解释为什么？

**解答：**

题 11.3 的索引项是按照升序排列的，从中可以看出：

当索引项按照升序插入时，新的索引项会被定位到叶结点，当一个叶结点被填满时，再加入一个结点将会导致该结点分裂成两个部分，其中一半保留在当前叶结点，另一半组成新的叶结点。除了最后一个叶结点，其余叶结点均会保持半满的状态，更准确一点是  $\lceil \frac{n}{2} \rceil$ ， $n$  表示一个结点容纳的指针数。

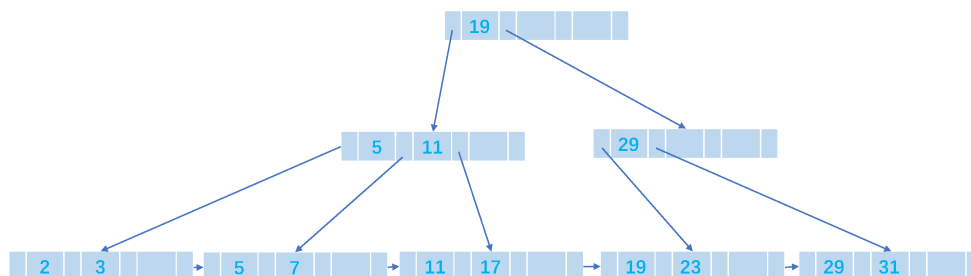
当索引项按照降序插入时，新的结点依然会被定位到叶结点，当叶结点满时也是进行分裂， $\lceil \frac{n}{2} \rceil$  留在原结点，其余组成新的结点。除了最后一个叶结点，其余都是半满状态。

## 11.17

对习题 11.3 中的每一棵  $B^+$  树，给出下列查询中涉及的步骤；

### a. 找出搜索码为 11 的记录

11.3-a

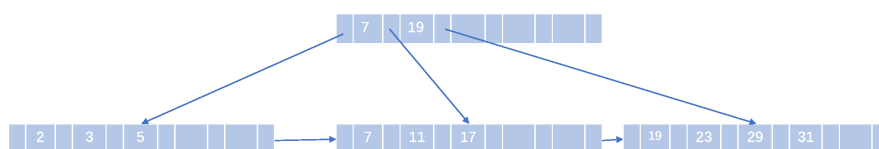


#### 查询过程

- 从根结点开始，找大于等于 11 的最小搜索码，找到 19( $K_1$ )。

- $11 < 19$ , 所以进入搜索码 19 的左指针 ( $P_1$ ) 中查找
- 在搜索码 19 的左孩子结点中找大于等于 11 的最小搜索码, 找到 11( $K_2$ )
- $11 = 11$ , 所以进入 11 的右指针 ( $P_3$ ) 中查找
- 访问到叶结点, 并在该叶结点中找到搜索码为 11( $K_1$ ) 的项, 根据指针 ( $P_1$ ) 指向找到记录并返回

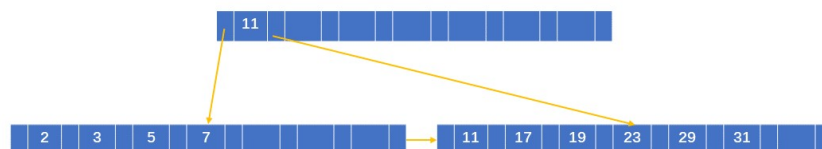
### 11.3-b



#### 查询过程

- 从根结点开始, 查找大于等于 11 的最小搜索码, 找到 19( $K_2$ )
- $11 < 19$ , 所以进入搜索码 19 的左指针 ( $P_2$ ) 中查找
- 该结点即为叶结点, 在叶结点中找到搜索码为 11 的项 ( $K_2$ ), 返回指针 ( $P_2$ ) 指向的记录

### 11.3-c

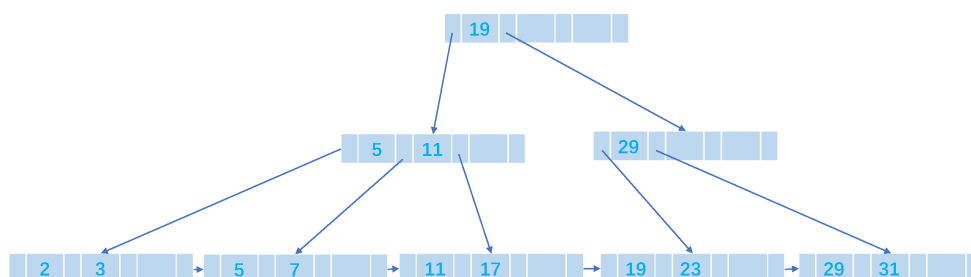


#### 查询过程

- 从根结点开始, 查找大于等于 11 的最小搜索码, 找到 11( $K_1$ )
- $11 = 11$ , 所以进入搜索码 11 的右指针 ( $P_2$ ) 中查找
- 该结点即为叶结点, 在叶结点中找到搜索码为 11 的项 ( $K_1$ ), 返回指针 ( $P_1$ ) 指向的记录

b. 给出搜索码值在 7-17 之间 (包括 7 和 17) 的记录

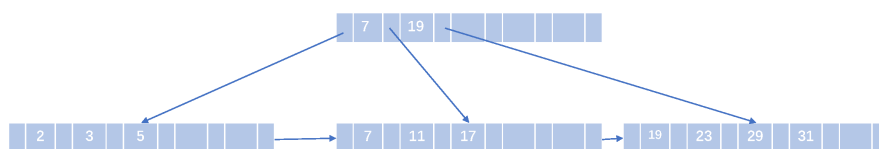
11.3-a



#### 查询过程

- 从根结点开始, 找大于等于 7 的最小搜索码, 找到 19( $K_1$ )
- $7 < 19$ , 从左指针 ( $P_1$ ) 进入下一层, 在结点中找大于等于 7 的最小搜索码, 找到 11( $K_2$ )
- $7 < 11$ , 从 11 的左指针 ( $P_2$ ) 进入下一层, 到达叶结点
- 在叶结点中找到搜索码为 7 的项 ( $K_2$ ), 并根据指针 ( $P_2$ ) 返回记录
- 顺着该叶结点向后查找, 到达最后一个指针 ( $P_4$ ), 根据指针链查找下一叶结点
- 在该叶结点中找到搜索码为 11( $7 \leq 11 \leq 17$ ) 的项 ( $K_1$ ), 根据指针 ( $P_1$ ) 返回记录
- 向后查找到搜索码为 17 的项 ( $K_2$ ), 返回指针 ( $P_2$ ) 指向的记录, 查找结束

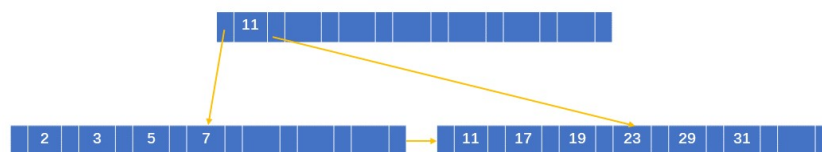
### 11.3-b



#### 查询过程

- 从根结点开始，查找大于等于 7 的最小搜索码，找到  $7(K_1)$
- $7 = 7$ , 从右指针 ( $P_2$ ) 进入下一级，到达叶结点
- 找到搜索码为  $7(K_1)$ 、 $11(K_2)$ 、 $17(K_3)$  的项，根据指针 ( $P_1, P_2, P_3$ ) 找到记录并返回

### 11.3-c



#### 查询过程

- 从根结点开始，查找大于等于 7 的最小搜索码，找到  $11(K_1)$
- $7 < 11$ , 从左指针 ( $P_1$ ) 进入下一级，达到叶结点
- 在该叶结点中找到搜索码为 7 的项 ( $K_4$ ), 由指针 ( $P_4$ ) 找到记录
- 在该叶结点中向后搜索，到达末尾指针 ( $P_8$ ), 由该指针指向查找下一个叶结点
- 查找该叶结点，找到搜索码为  $11(K_1)$ 、 $17(K_2)$  的项，根据指针 ( $P_1, P_2$ ) 找到记录
- 返回记录，查找结束