1、已知一棵B+树，如图1所示

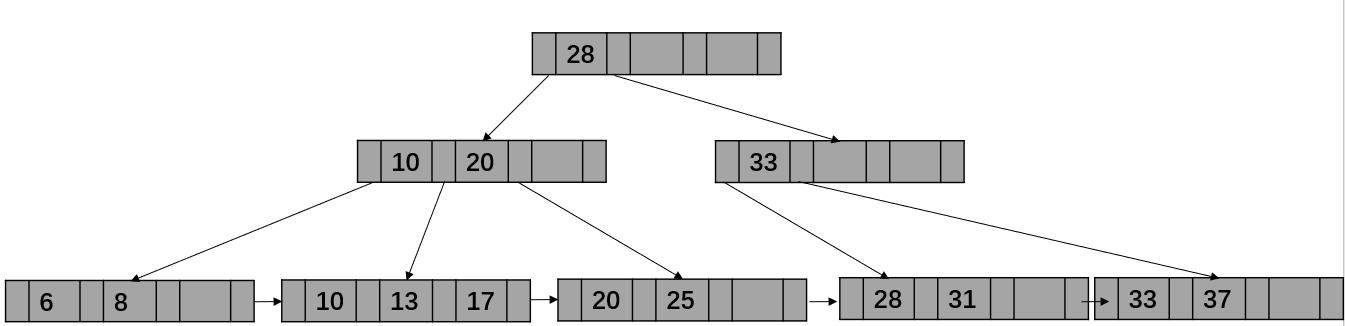


图1

（1）请画出在图1中插入19后所得的新的B+树

1. 请画出在图1中删除31后所得的新的B+树。

2、已知一棵B+树，如图2所示

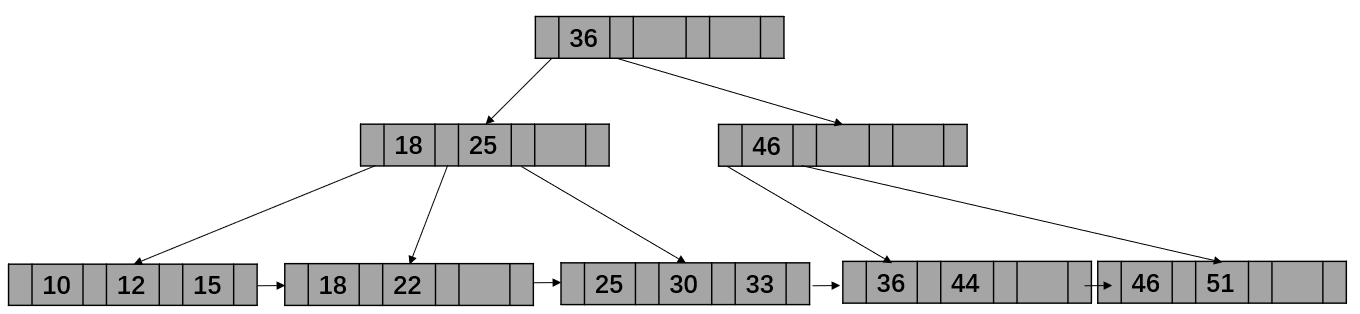


图2

（1）请画出在图2中插入8后所得的新的B+树

1. 请画出在（1）的结果中删除46后所得的新的B+树

3、利用可扩展hash方法对以下记录进行hash存储：

3, 5, 7, 12, 16

设hash函数h(x)= x mod 8，其中散列函数h(k)是一个b(足够大)位二进制序列，序列的前d位用作索引，来区分每个元素属于哪个桶。

现要求每个桶至多包含2个元素，以上元素按从左往右的顺序依次添加。开始时只使用序列的前1位作索引（即d=1），当桶满时进行分裂，d相应增大。请画出添加完以上所有元素后，最终的索引结构。

4、可扩展hash表存储整型元素。设hash函数为h(x) = x mod 16，h(x)的结果是⼀个4位二进制序列，其前d位用于确定元素x属于哪个桶。现将下列元素依次添加到hash表中：5, 19, 24, 26, 35, 48, 53, 61。

现要求每个桶至多包含2个元素，以上元素按从左往右的顺序依次添加。开始时只使用序列的前1位作索引（即d=1），当桶满时进行分裂，d相应增大。请画出添加完以上所有元素后，最终的索引结构。

5、利用可扩展hash方法对以下记录进行hash存储：

3, 16, 20, 21, 30, 44, 47，61

设hash函数h(x)= x mod 16，其中散列函数h(k)是一个b(足够大)位二进制序列，序列的前d位用作索引，来区分每个元素属于哪个桶。

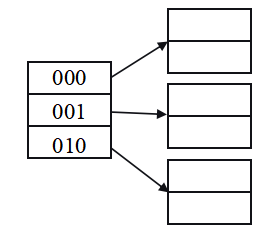
现要求每个桶至多包含2个元素，以上元素按从左往右的顺序依次添加。开始时只使用序列的前1位作索引（即d=1），当桶满时进行分裂，d相应增大。请画出添加完以上所有元素后，最终的索引结构。

6、利用线性hash方法对以下记录进行hash存储：

5, 19, 24, 26, 35, 48, 53, 61

设hash函数h(x)= x mod 16，请画出请画出添加以上所有元素后，最终的索引结构以及关键步骤（进行桶的线性增长时）的索引结构。

注：线性hash表中最多容纳个记录，；

初始哈希桶结构为右图。

7、利用线性hash方法对以下记录进行hash存储：

3, 16, 20, 21, 30, 44, 47，61

设hash函数h(x)= x mod 16，请画出请画出添加以上所有元素后，最终的索引结构以及关键步骤（进行桶的线性增长时）的索引结构。

注：线性hash表中最多容纳个记录，；

初始哈希桶结构为右图。