

作为例子, 设对 H_3 执行一次 `deleteMin`, 它在图 6-46 中表示。最小的根是 12, 因此我们得到图 6-47 和图 6-48 中的两个优先队列 H' 和 H'' 。合并 H' 和 H'' 得到的二项队列是最后的答案, 如图 6-49 所示。

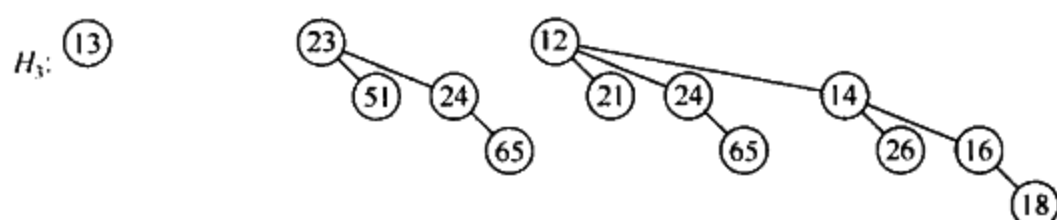


图 6-46 二项队列 H_3



图 6-47 二项队列 H' , 包含除 B_3 外 H_3 中所有的二项树

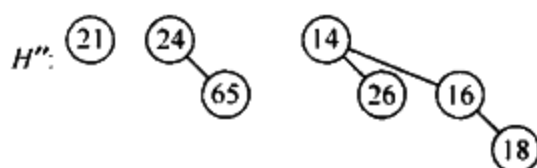


图 6-48 二项队列 H'' : 除去 12 后的 B_3

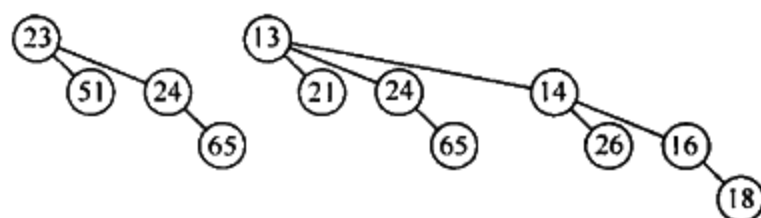


图 6-49 `deleteMin` 应用到 H_3 的结果

为了分析, 首先注意, `deleteMin` 操作将原二项队列一分为二。找出含有最小元素的树并创建队列 H' 和 H'' 花费时间 $O(\log N)$ 。合并这两个队列又花费 $O(\log N)$ 时间, 因此, 整个 `deleteMin` 操作花费时间 $O(\log N)$ 。

6.8.3 二项队列的实现

`deleteMin` 操作需要快速找出根的所有子树的能力, 因此, 需要一般树的标准表示方法: 每个节点的儿子都在一个链表中, 而且每个节点都有一个对它的第一个儿子(如果有的话)的引用。该操作还要求各个儿子按照它们的子树的大小排序。我们还需要保证合并两棵树容易。当两棵树被合并时, 其中的一棵树作为儿子被加到另一棵树上。由于这棵新树将是最大的子树, 因此, 以大小递减的方式保持这些子树是有意义的。只有这时我们才能够有效地合并两棵二项树从而合并两个二项队列。二项队列将是二项树的数组。

总而言之, 二项树的每一个节点将包含数据、第一个儿子以及右兄弟。二项树中的各个儿子以降秩次序排列。

图 6-51 解释了如何表示图 6-50 中的二项队列。图 6-52 显示二项树中的节点的类型声明以及二项队列的类架构。

为了合并两个二项队列, 我们需要一个例程来合并两个同样大小的二项树。图 6-53 表明两个二项树合并时链是如何变化的。合并同样大小的两棵二项树的程序很简单, 见图 6-54。