

## 二项队列的实现—PPT 第七页

首先，二项队列就是一个二项树的数组。所以接下来表格里要讨论的是二项树 Binomial tree 的设计策略。

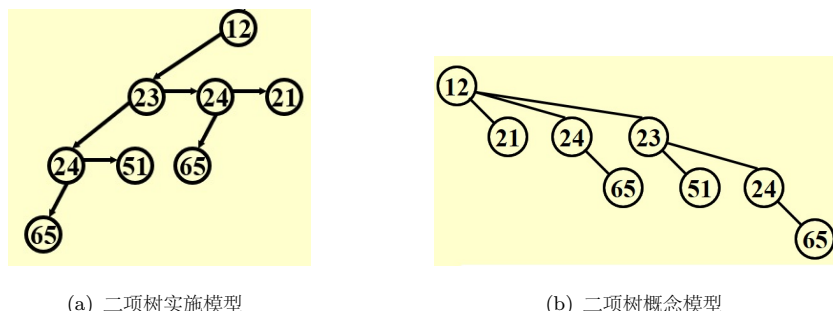


图 1: 二项树模型

在图 1 二项树实施模型和概念模型中，要想从根节点 ⑫ 开始快速遍历三个子树（⑪，⑭，⑮）就需要用“左孩右兄链相连”的实现方式。对应图 1 ⑫ → ⑮ → ⑭ → ⑪ 这条路径。

图 1 的  $B_3$  是两颗  $B_2$  的合并结果，为什么合并后要按子树大小降序排列呢？（1）二项树的子树必须是有序的，否则删除根节点后还要重新排序才能合并就慢了。问题是按树大小降序还是升序？（2）升序的话合并开销大；比如：根节点为 ⑫ 和 ⑮ 两棵树合并时，⑮ 会成为 ⑫ 的子树。这时要按“左孩右兄链相连”将子树 ⑬ 插入到子树链表，这时得先遍历所有子节点（21, 24）才能将最大这颗子树加上，多出了遍历的开支。

有个曾经把老师搞晕的神问题，每年都会有同学问：PPT 第 10 页的二项队列归并代码中，有一句两棵树归并的 for 语句：

```
for ( i=0, j=1; j<= H1->CurrentSize; i++, j*=2 ) {  
    T1 = H1->TheTrees[i]; T2 = H2->TheTrees[j]; /*current trees */  
    ..... /*对应大小子树 combine*/  
}
```

同学们曾问：那个以  $i$  作为索引的 for 循环是否意味着子树排列是由小到大？

其实是同学们弄混了二项树和二项队列，上面的 for 循环是二项队列的操作。二项队列长这个样子：

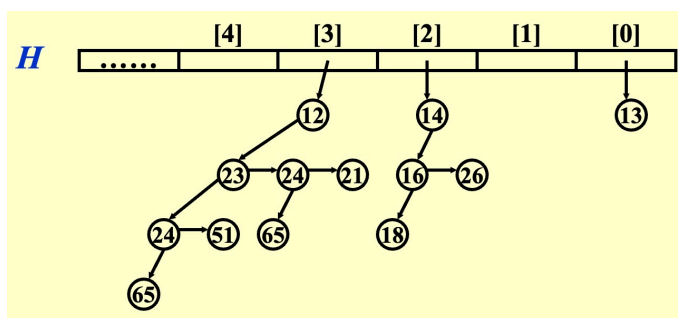


图 2: 二项队列长这个样子