一、什么是2-3-4树

2-3-4树和红黑树一样，也是平衡树。只不过不是二叉树，它的子节点数目可以达到4个。

每个节点存储的数据项可以达到3个。名字中的2,3,4是指节点可能包含的子节点数目。具体而言：

1、若父节点中存有1个数据项，则必有2个子节点。

2、若父节点中存有2个数据项，则必有3个子节点。

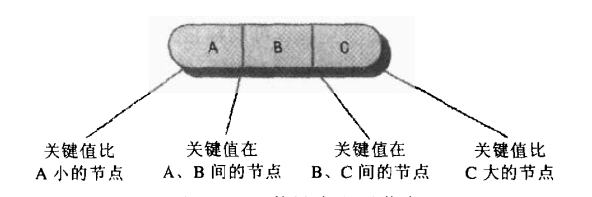
3、若父节点中存有3个数据项，则必有4个子节点。

也就是说子节点的数目是父节点中数据项的数目加一。因为以上三个规则，使得除了叶结点外，其他节点必有2到4个子节点，不可能只有一个子节点。所以不叫1-2-3-4树。而且2-3-4树中所有叶结点总是在同一层。

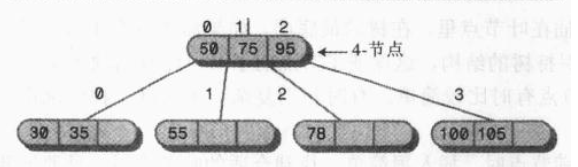
二、如何构建2-3-4树。

首先需要明白二叉树一般不允许出现重复的关键字。分析问题时就不考虑这种情况。

构建原则：



实例：



对于包含1或2个数据项的节点，构造原则相同。

搜索2-3-4树：

与二叉树方法类似，只不过比较过程较为复杂，处于不同的数据段之间转向不同的子树。结合上面的构造方法更好理解。

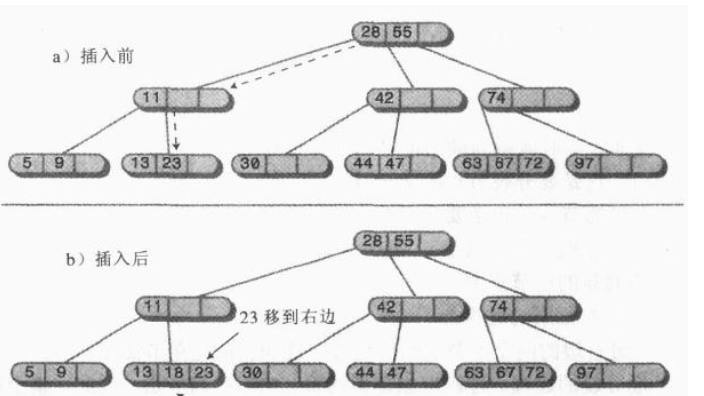
插入操作：

新的数据项总是插入在叶结点中，在树的最底层。这时可能有两种情况：

1.未遇到满节点

这种情况较为简单，只需找到相应的位置插入数据即可。

如下例：插入数据18



2、遇到满节点情况：

为了保证树的平衡和2-3-4树的结构，需要进行分裂操作。从上到下寻找插入位置时遇到的任何一个满节点都要进行分裂操作。

假设满节点中的数据项为A，B，C。根据节点是不是根又分为两种情况。

（1）满节点不是根

分裂方法：

·创建一个新的节点，与要分裂的节点是兄弟，且放在其右侧。

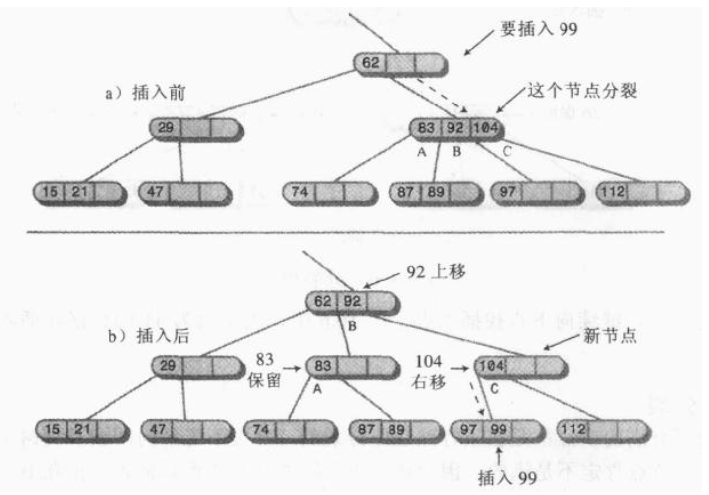
·把数据项C移动到新节点中。

·将数据项B移到父节点中的相应位置。

·将数据项A保留在原节点中。

·把原节点最右边的两个子节点（原节点为满节点，则一定有四个子节点或者是叶结点）从要分裂的节点上断开，连接到新的节点上。

实例：插入99



（2）满节点是根节点

分裂方法：

·创建一个新的根，作为要分裂节点的父节点。

·再创建一个新的节点，作为要分裂节点的兄弟节点，位于其右侧。

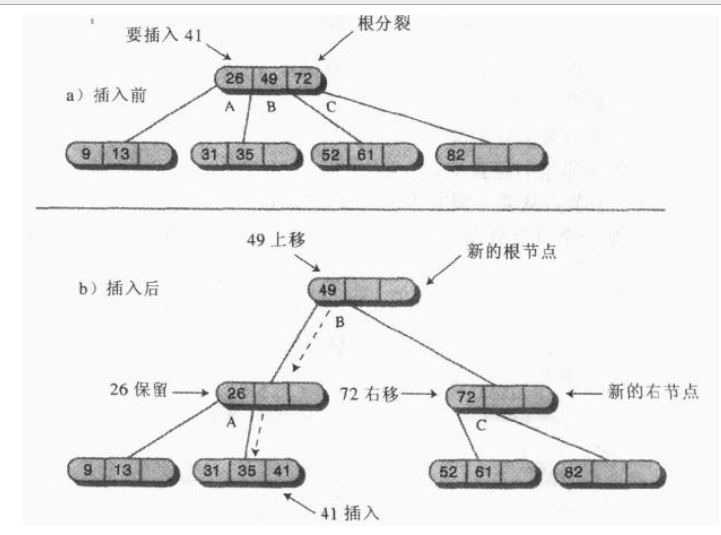
·数据项C移动到兄弟节点中。

·数据项B移动到父节点中。

·数据项A保留在原节点。

·把要分裂节点的最右边的两个子节点断开连接，重新连接到兄弟节点上。

实例：插入41



查找是从上到下的，所以分裂也是从最上方的满节点开始。这也保证了要分裂的节点的父节点一定是不满的，否则应该先分裂父节点。