红黑树

预备知识：二叉查找树、B树（3阶B树，每个节点中最多有两个关键字）

红黑树的定义

将树中的链接分成两种：

红链接将两个普通的2-结点连接起来构成一个具有两个关键字的3-结点（k-结点表示结点有k-1个关键字，k个链接），黑链接则是2-3数中的普通链接

将3-结点表示为由一条左斜的红色链接项链的两个2-结点

等价定义

红黑树的另一种定义是含有红黑链接并满足下列条件的二叉查找树

- 红链接均为左链接
- 没有一个结点同时和两条红色链接相连
- 该树是完美平衡的，即任意空链接到根节点的路径上的黑链接的数量相同

如果将一棵红黑树中的红链接全部画平，那么所有空白空链接到根节点的距离都是相同的，如果将红链接相连的两个节点合并，得到的就是一棵2-3树，红黑树既是二叉查找树，也是2-3树， 它具有二叉查找树中简洁高效的查找方法和2-3树中高效的平衡插入算法

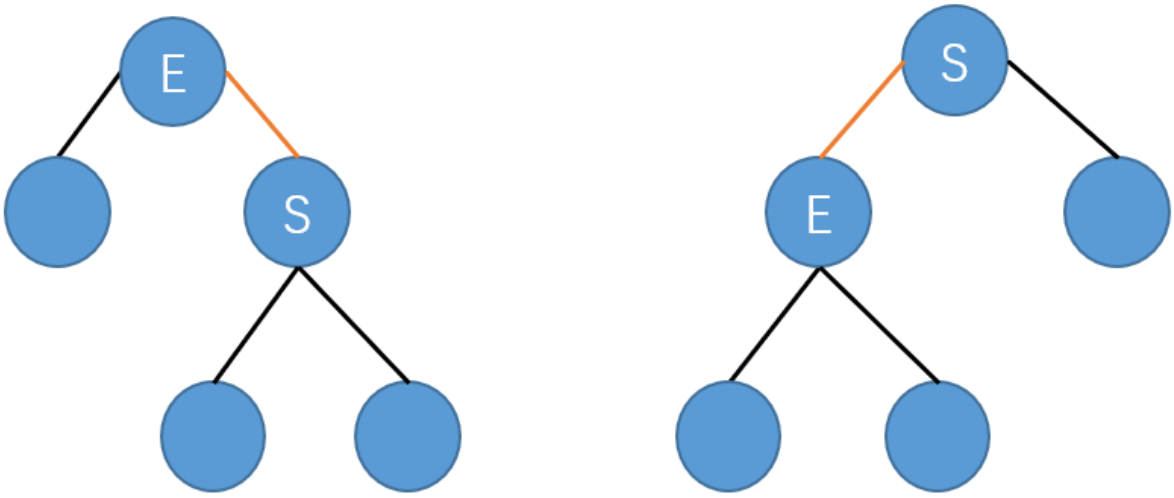
颜色定义

对于每个结点而言，都有一个父节点指向本身的链接（根节点除外），用TreeNode.color来表示指向当前节点链接的颜色

```
//Definition for a red_balck tree node.
struct TreeNode {
    int val;
    boolean isRed;  //指向当前节点的链接是否是红色
    TreeNode *left;
    TreeNode *right;
    TreeNode() : val(0), left(nullptr), right(nullptr) {}
    TreeNode(int x) : val(x), left(nullptr), right(nullptr) {}
    TreeNode(int x, TreeNode *left, TreeNode *right) : val(x), left(left), right(right) {}
};
```

旋转

在实现过程中，可能会出现连续两条红色链接或者红色右链接，在这时候需要进行旋转操作



假设现在有一条红色的右链接，需要通过旋转将右链接转换为左链接，**左旋**

```
TreeNode* rotateLeft(TreeNode *root){   //根节点的右链接是红色的
    TreeNode* p = root->right;
    root->right = p->left;
    p->left = root;
    p.isRed = root.isRed;   //颜色转换
    root.isRed = true;
    return p;
}
```

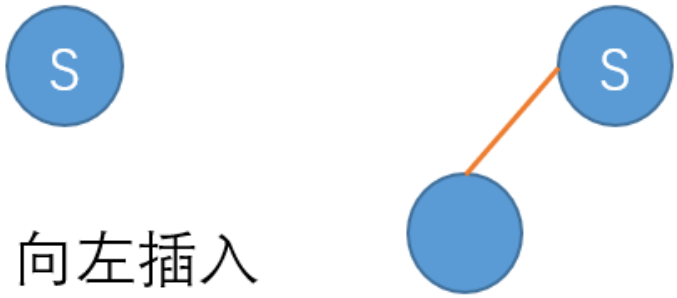
右旋

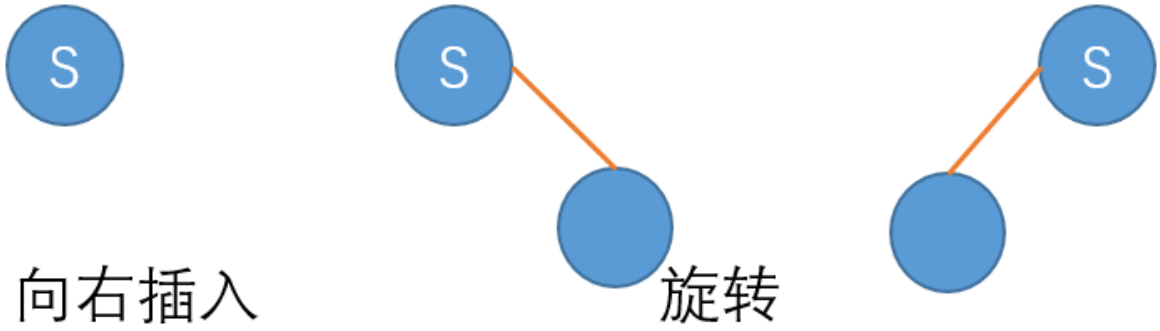
```
TreeNode* rotateRight(TreeNode* root){
    TreeNode* p = root->left;
    root->left = p->right;
    p->right = root;
    p.isRed = root.isRed;
    root.isRed = true;
    return p;
}
```

向单个2-节点中插入新键

在2-节点中插入一个新键，要么在这个节点的左边进行插入，要么在这个节点的右边进行插入。

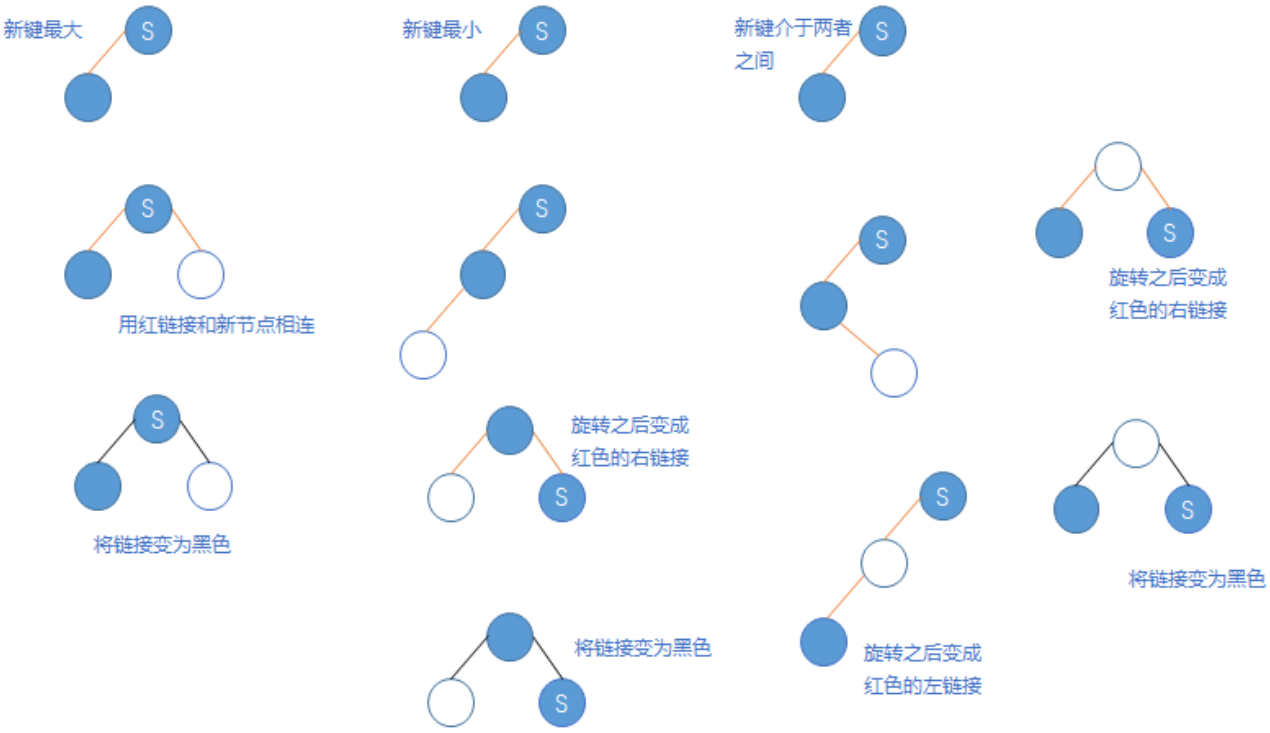
- 在左边进行插入，只需要新增一个红色的结点
- 在右边进行插入，新增的结点是红色的，利用上面的rotateLeft进行一次左旋即可





向一个3-结点中插入新键

- 新插入的键值最大，它被连接到3-结点的右链接。此时，树是平衡的，根节点为中间大小的键。如果能够将两条链接的颜色由红变黑，那么就得到了一棵由三个节点组成，高度为2的平衡树。它正好对应一棵2-3树
- 新插入的键最小，它被连接到2-3结点的左链接。这样就产生了两条连续的红链接，此时将上层的红链接进行右旋，就可以转换成第一种情况
- 新插入的键介于两者之间，它被连接到2-3结点的中间，将下层的红链接进行左旋，将情况转换成第一种情况



颜色转换

使用flipColors()来转换一个节点两个子节点的颜色，同时将父结点的颜色由黑变红

```
void filpColors(TreeNode* root){
    h.isRed = true;
    h.left.color = h.right.color = false;
}
```

将红链接在树中向上传递

在2-3树中我们需要分解3-结点，将中间键插入到父节点中，直到遇到一个2-结点或者根节点。

每次旋转之后都会改变节点的颜色，相当于将该节点插入到它的父节点当中。

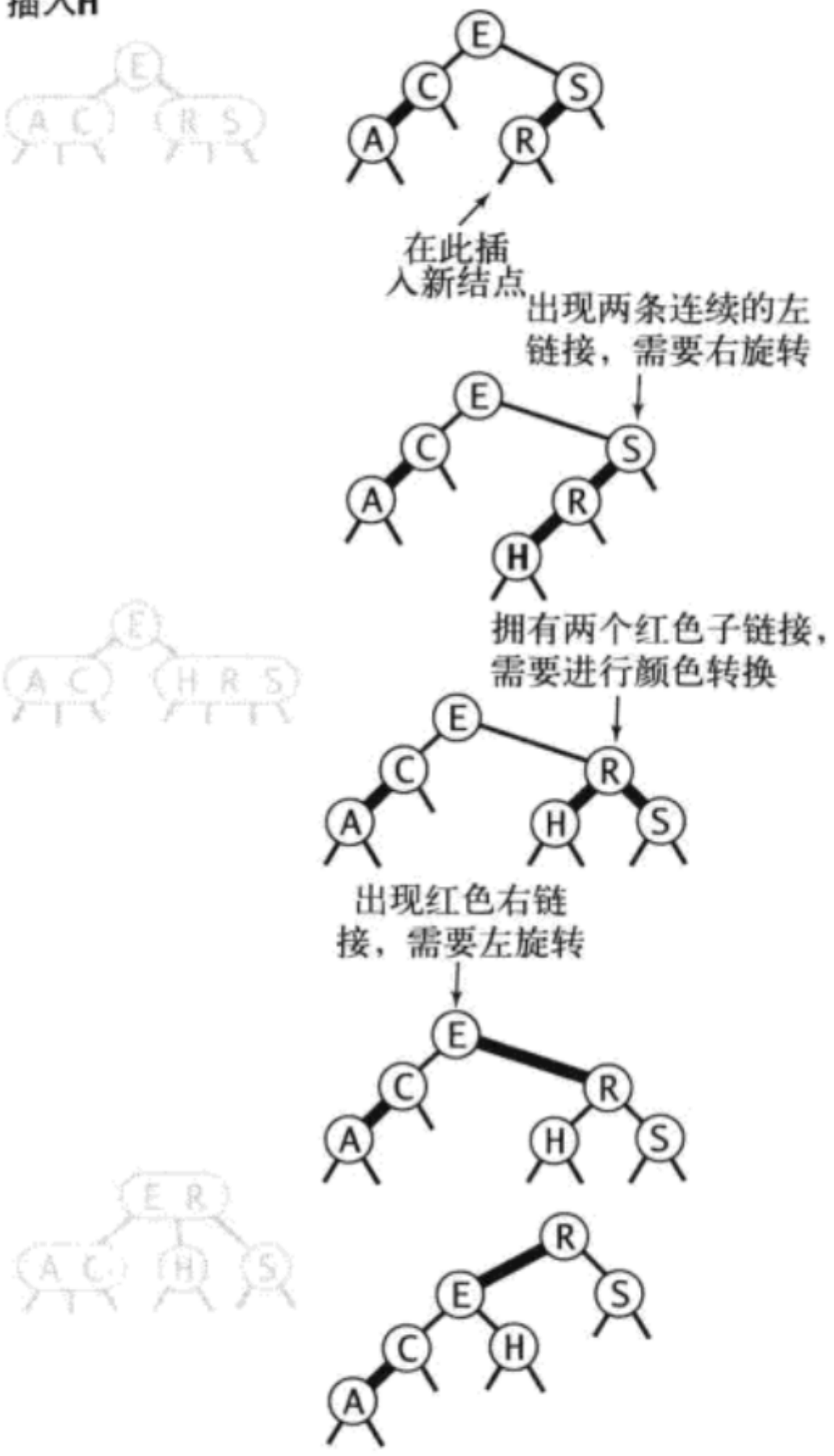


图 3.3.22　向树底部的 3- 结点插入一个新键（另见彩插）

只需要使用左旋、右旋、颜色转换这3种操作，就能够保证插入操作后红黑树和2-3树一一对应的关系。

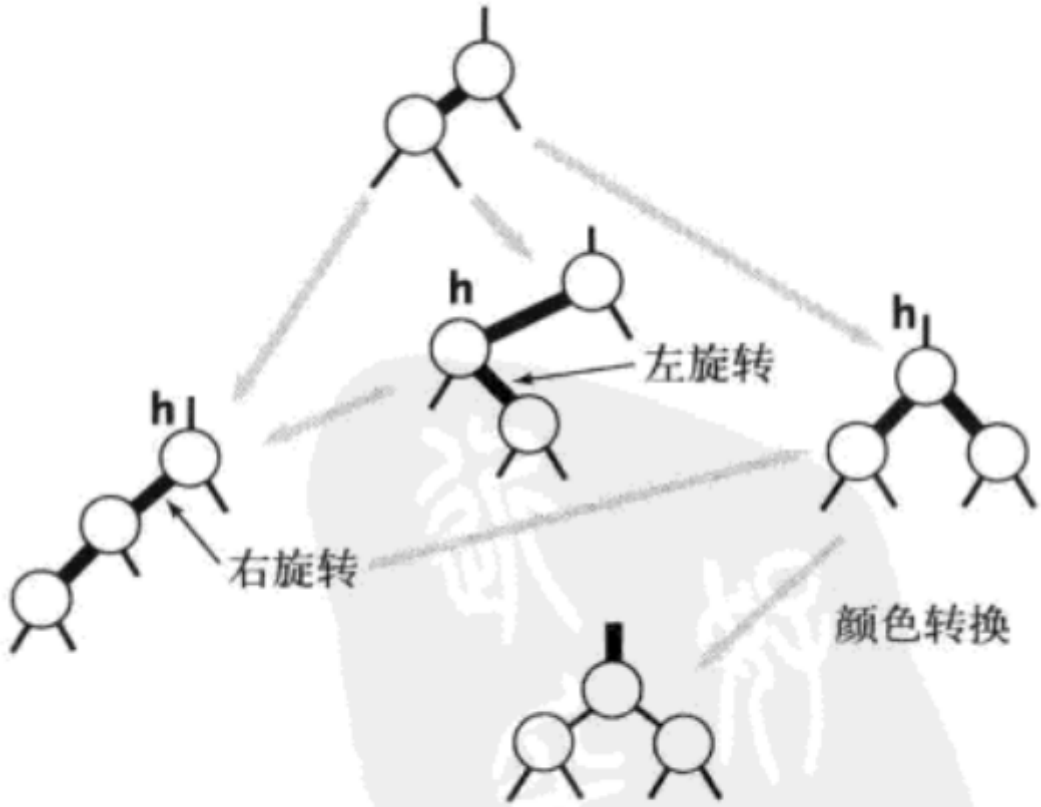


图 3.3.23　红黑树中红链接向上传递（另见彩插）