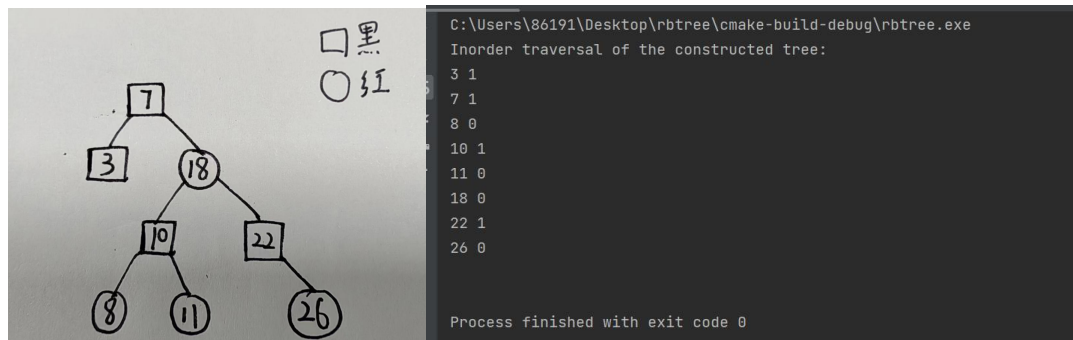


1、程序运行结果中每行第一个为节点值，第二个为颜色（红为 0，黑为 1）



2、删除后采取的失衡恢复操作只在双黑情况（被删除节点和其继承节点均为黑色）下会出现。

2.1 (BB-1) 被删除节点 (x) 的兄弟节点 (s) 为黑色节点且 s 节点有红色子节点 (t)：

将 p 和 s 节点颜色互换：被删除侧路径上少一个未知的原 p 节点颜色的节点，未被删除侧路径可能出现连续两个红色节点。

将 t 节点染黑，防止未被删除侧连续两个红色节点，但导致未被删除侧路径上黑色节点增多失衡

p 节点右旋/左旋：被删除侧路径少的原 p 节点颜色的节点恢复，删除侧路径上黑色节点数减少恢复

2.2 (BB-2R) 被删除节点 (x) 的兄弟节点 (s) 为黑色节点且 s 节点无红色子节点 (t)，且被删除节点的父节点 (p) 为红色节点：

将 p 节点染黑：被删除侧路径黑色节点数量增多恢复，未被删除侧路径对黑色节点数量增多失衡

将 s 节点染红：未被删除侧黑色节点数量减少恢复。

2.3 (BB-2B) 被删除节点 (x) 的兄弟节点 (s) 为黑色节点且 s 节点无红色子节点 (t)，且被删除节点的父节点 (p) 为黑色节点：

将 s 节点染红，使未被删除侧黑色节点减少，等于删除侧黑色节点，局部修复红黑树性质，但由于高度下降，可能出现父节点失问题

父节点作为删除后的替代节点继续递归修复：解决高度下降带来的红黑树失衡问题

2.4 (BB-3) 被删除节点 (x) 的兄弟节点 (s) 为红色节点

将 p 右旋/左旋：未被删除侧黑色节点数减少失衡

s 黑转红：未被删除侧黑色节点数增多恢复，被删除侧黑色节点数增多失衡

p 黑转红：被删除侧黑色节点数减少恢复，转化为 BB-1 或 BB-2R 的情况

根据所在情况再次进行一轮调整

3

顺序插入结果：

```
C:\Users\86191\Desktop\rbtrees\cmake-build-debug\rbtrees.exe
fixTime:19947 RotationTime:9976 ColorChangeTime:59865
Process finished with exit code 0
```

乱序插入结果（10000 次实验平均值）：

```
C:\Users\86191\Desktop\rbtree\cmake-build-debug\rbtree.exe
Average fixTime:9007      Average RotationTime:5820      Average ColorChangeTime:33141
Process finished with exit code 0
```

由实验数据可知，乱序插入的修复次数，旋转次数，染色次数均小于顺序插入，说明乱序插入能比较好地维护红黑树的形状，减少修复次数，而顺序插入会导致红黑树退化成链表，因此需要更多修复，染色，旋转来维护红黑树形状。