1. 请以“插入”操作为例，解释插入操作是如何使得红黑树依然维持以上五个性质，具体而言，请回答：

1.1 为什么规定插入的结点为红色

如果插入的节点是黑色：根据性质五“从任一结点到其每个叶子的所有路径都包含相同数目的黑色结点”，如果我们要保证性质五成立，即要保证红黑树中从根节点到任意叶子节点的路径上的黑色节点数量相同。但是如果插入的节点是黑色，那么根节点就会有一条路径上的黑色节点数量与其他路径不同，这样插入的话，就需要对红黑树进行整体上的调整，比较麻烦。

如果插入的节点是红色：此时所有路径上的黑色节点数量不变，问题转化为对红黑树局部节点的调整。这种情况下，只会出现不满足性质四“每个叶子到根的所有路径上不能有两个连续的红色结点”的情况。这时，通过对插入节点附近局部节点进行变色和旋转的调整操作就能重新满足这个性质，调整起来相对插入黑色节点来说更加方便。

因此，规定插入的结点为红色。

1.2 当插入结点的父结点为红色，如何操作帮助其恢复该性质（假设此时父节点的兄弟结点为黑色）

LL ：

将父节点染黑：防止连续两个红色结点，但导致左侧路径上黑色结点增多失衡

祖父染红：左侧路径上黑色结点数恢复，但右侧路径上黑色结点数减少失衡

右旋：右侧路径黑色结点数增加，平衡恢复

LR：

对父节点左旋：转化成 LL 的情况

将新节点染黑：防止连续两个红色结点，但导致左侧路径上黑色结点增多失衡

祖父染红：左侧路径上黑色结点数恢复，但右侧路径上黑色结点数减少失衡

右旋：右侧路径黑色结点数增加，平衡恢复

RR：

将父节点染黑：防止连续两个红色结点，但导致右侧路径上黑色结点增多失衡

祖父染红：右侧路径上黑色结点数恢复，但左侧路径上黑色结点数减少失衡

左旋：左侧路径黑色结点数增加，平衡恢复

RL：

对父节点右旋：转化成 RR 的情况

将新节点染黑：防止连续两个红色结点，但导致右侧路径上黑色结点增多失衡

祖父染红：右侧路径上黑色结点数恢复，但左侧路径上黑色结点数减少失衡

左旋：左侧路径黑色结点数增加，平衡恢复

2. 红黑树的本质是一个 2-3-4 树（4 阶 B 树），请谈一谈你对这句话的理解

4 阶 B 树一共有三种节点：2-节点：包含一个元素和两个子节点；3-节点：包含两个元素和三个子节点；4-节点：包含三个元素和四个子节点。

红黑树有两种节点：红色节点和黑色节点。如果我们将红黑树中的黑色节点看作主要节点，红色节点看作是黑色节点的附属节点，我们就可以将 4 阶 B 树种的三种节点转化为红黑树中红色节点和黑色节点的不同组合，一共有三种情况：

2-节点：包含一个元素，我们可以将它看作红黑树中的一个单独的黑色节点。

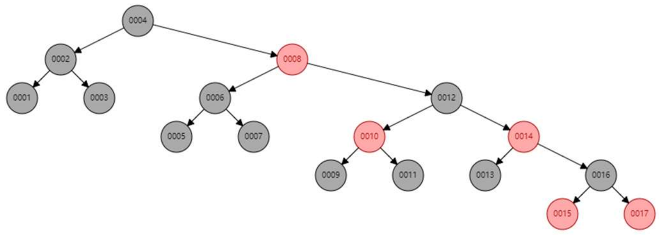
3-节点：包含两个元素，我们可以将它看作是红黑树中红色节点和黑色节点的组合（有两种排列方式，1：较大元素作为主值，为黑色节点，较小元素作为较大元素的左孩子，为红色节点；2：较小元素作为主值，为黑色节点，较大元素作为较小元素的右孩子，为红色节点）

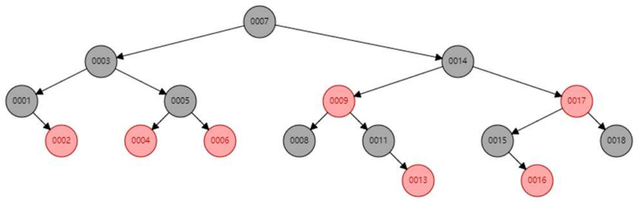
4-节点：包含三个元素，我们将其中中间大小的元素作为主值，为黑色节点，最小元素作为中间元素的左孩子，最大元素作为中间元素的右孩子，两者都为红色节点。

这样我们就完成了 4 阶 B 树到红黑树形式上的转化，我们发现这样转化后的红黑树满足其基本的五条性质。同时，如果对这样转化方式下的相同状态的红黑树和四阶 B 树做相应的增删操作，也不会改变其本质的相同性。因此红黑树的本质是一个 4 阶 B 树。

3. 红黑树顺序插入和乱序插入会有何影响?

通过简单实验，我对红黑树顺序插入和乱序插入的影响进行了初步探究。首先，分别进行从元素 1 至 17 顺序插入和乱序插入红黑树的不同结果，如下两张图所示：





可以看到，上图为顺序插入 17 个节点后的红黑树，下图为乱序插入 17 个节点后的红黑树。我们发现在顺序插入的情况下，红黑树根节点的左右子树的高度更容易发生不平衡的情况（图中为左子树高度低于右子树高度），算上红色节点的个数，高度相差一倍。因此、乱序插入后的红黑树在大部分情况下相对于顺序插入的红黑树更加平衡。