🤔👀树 - 红黑树(R-B Tree)

红黑树（Red Black Tree） 是一种自平衡二叉查找树，是在计算机科学中用到的一种数据结构，典型的用途是实现关联数组，是平衡二叉树和AVL树的折中。

为什么要有红黑树

平衡二叉树(AVLTree)，了解到AVL树的性质，其实平衡二叉树最大的作用就是查找,AVL树的查找、插入和删除在平均和最坏情况下都是O(logn)。AVL树的效率就是高在这个地方。如果在AVL树中插入或删除节点后，使得高度之差大于1。此时，AVL树的平衡状态就被破坏，它就不再是一棵二叉树；为了让它重新维持在一个平衡状态，就需要对其进行旋转处理, 那么创建一颗平衡二叉树的成本其实不小. 这个时候就有人开始思考，并且提出了红黑树的理论，那么红黑树到底比AVL树好在哪里?

红黑树与AVL树的比较:

- 1.AVL树的时间复杂度虽然优于红黑树，但是对于现在的计算机，cpu太快，可以忽略性能差异
- 2.红黑树的插入删除比AVL树更便于控制操作
- 3.红黑树整体性能略优于AVL树(红黑树旋转情况少于AVL树)

> 红黑树的性质: 红黑树是一棵二叉搜索树，它在每个节点增加了一个存储位记录节点的颜色，可以是RED,也可以是BLACK；通过任意一条从根到叶子简单路径上颜色的约束，红黑树保证最长路径不超过最短路径的二倍，因而近似平衡。

具体性质如下:

- 每个节点颜色不是黑色，就是红色
- 根节点是黑色的
- 如果一个节点是红色，那么它的两个子节点就是黑色的(没有连续的红节点)
- 对于每个节点，从该节点到其后代叶节点的简单路径上，均包含相同数目的黑色节点

应用场景

- Java ConcurrentHashMap & TreeMap
- C++ STL: map & set
- linux进程调度Completely Fair Scheduler,用红黑树管理进程控制块
- epoll在内核中的实现，用红黑树管理事件块
- nginx中，用红黑树管理timer等