

Uncle_Bjorney

博客园 首页 新随笔 联系 订阅 管理

随笔 - 24 文章 - 0 评论 - 0 阅读 - 13305

公告

昵称: Uncle_Bjorney
园龄: 4年9个月
粉丝: 2
关注: 0
[+加关注](#)

< 2022年9月 >						
日	一	二	三	四	五	六
28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	1
2	3	4	5	6	7	8

搜索

找我看

谷歌搜索

常用链接

[我的随笔](#)
[我的评论](#)
[我的参与](#)
[最新评论](#)
[我的标签](#)

随笔分类

[concurrent\(9\)](#)
[lang\(2\)](#)
[nio\(2\)](#)
[spring\(4\)](#)
[util\(5\)](#)
[zookeeper\(2\)](#)

随笔档案

[2019年2月\(5\)](#)
[2018年1月\(4\)](#)
[2017年12月\(15\)](#)

阅读排行榜

1. InetAddress之域名解析(2987)
2. 红黑树 (Red Black Tree) (2231)
3. Java NIO入门小例（短连接：客户端和服务端一问一答）(2044)
4. ConcurrentHashMap (Java8) 源码分析(1068)
5. ReentrantReadWriteLock（读写锁）源码分析(674)

红黑树 (Red Black Tree)

和AVL树一样，红黑树也是一种自平衡二叉排序树，其定义如下：

- (1) 节点有且只有两种颜色，红色和黑色。
- (2) 根节点和叶子节点必须是黑色，其中，叶子节点是虚拟存在的空节点（NULL）。
- (3) 红色节点的两个子节点必须是黑色。
- (4) 任意节点到叶子节点的路径上，必须包含相同数目的黑色节点。

从红黑树的定义可以发现，任意节点左右子树的高度差在一倍之内（最长路径为节点红黑相间，最短路径为节点全黑）。

由于红黑树对平衡性的要求没有AVL树高，因此频繁插入和删除节点时，触发平衡调整的次数更少，平衡调整的过程也更易收敛。

1. 插入节点

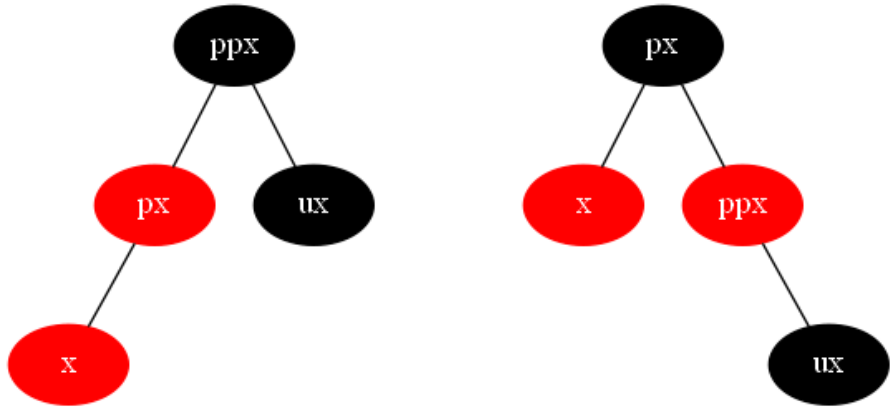
新插入节点x，x.color = RED，未违背定义（1）（2）（4），可能违背定义（3）。

若根节点ROOT== NULL，则ROOT= x，ROOT.color = BLACK，RETURN。

LOOP:

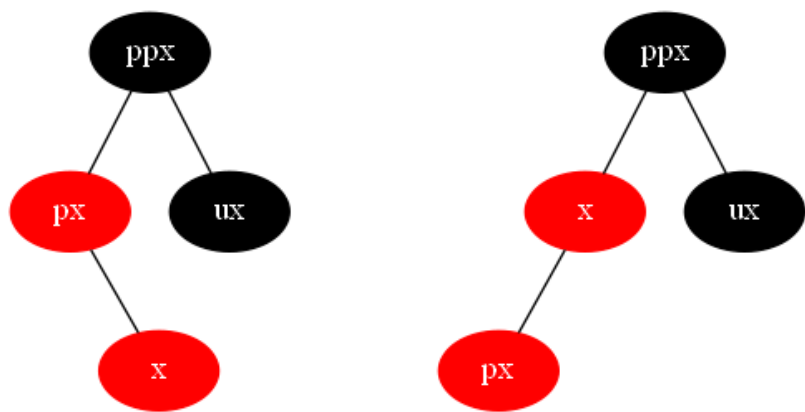
- (1) 若当前节点x == ROOT，则ROOT.color = BLACK，RETURN。
- (2) 若父节点px = x.parent，px.color == BLACK，则未违背定义（3），RETURN。
- (3) 爷爷节点ppx = px.parent（ppx一定为黑色）。假设px为ppx的左孩子（px为右孩子的情况同理），则叔叔节点ux = ppx.right:

（特殊情况一）ux为黑色，x为px的左孩子



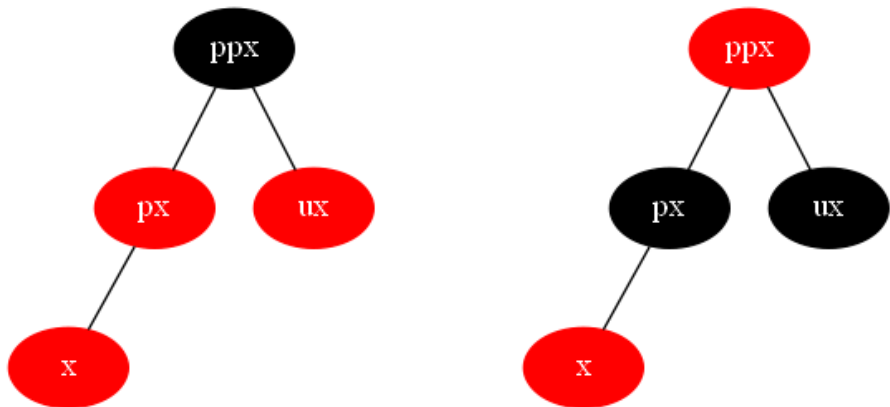
px.color = BLACK，ppx.color = RED，再以ppx为支点右旋，回到LOOP（到（2）时RETURN）。

（情况二）ux为黑色，x为px的右孩子



以px为支点左旋，**当前节点x = px**，进入（特殊情况一）。

(情况三) ux为红色



px.color = BLACK, ux.color = BLACK, ppx.color = RED, **当前节点x = ppx**，回到LOOP。

2. 删除节点

若被删除节点m的左右子树非空，则将左子树的最大节点（或右子树的最小节点）n的数据保存到节点m上，再删除节点n。

若n.color == RED，则未违背定义（1）（2）（3）（4），RETURN。

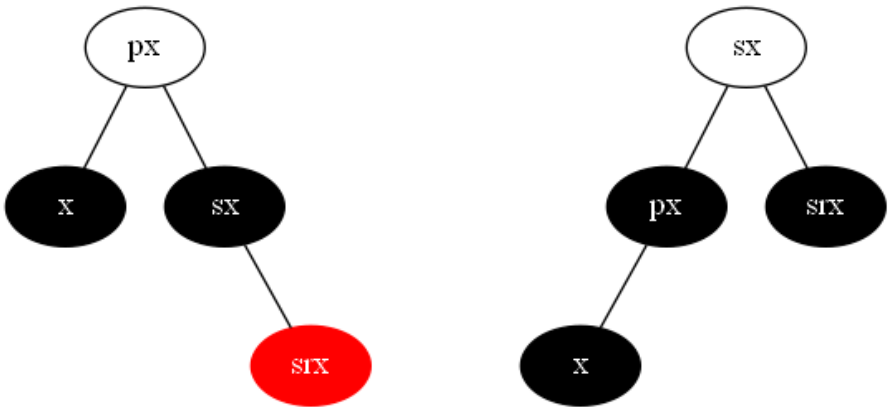
若n.color == BLACK，则未违背定义（1）（2），**可能违背定义（3）（4）**。设节点n的孩子节点为x：

- （1）若n.parent == NULL（即n == ROOT），且x == NULL，则说明n为唯一节点，**未违背定义（3）（4）**，RETURN。
- （2）若n.parent == NULL（即n == ROOT），且x != NULL，则说明x将为根节点，ROOT = x。
- （3）若n.parent != NULL，且x == NULL，则**x = n**，待平衡调整完毕后再删除x。
- （4）若n.parent != NULL，且x != NULL，则删除节点n，节点n的孩子节点x替到n的位置。

LOOP:

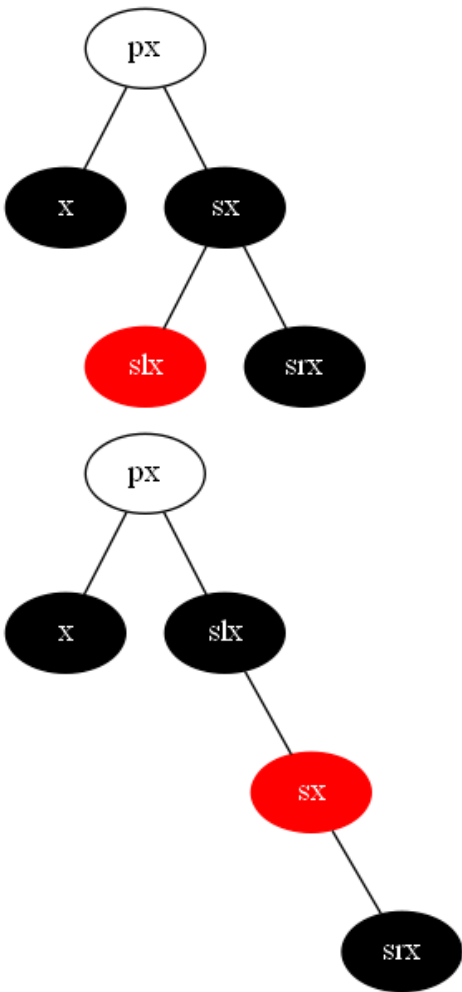
- （1）若**当前节点x** == ROOT，则x.color = BLACK，RETURN。
- （2）若x.color == RED，则x.color = BLACK，RETURN。
- （3）父节点px = x.parent，假设x为px的左孩子（x为右孩子的情况同理），则兄弟节点sx = px.right：

(特殊情况一) sx为黑色，sx的右孩子srx为红色



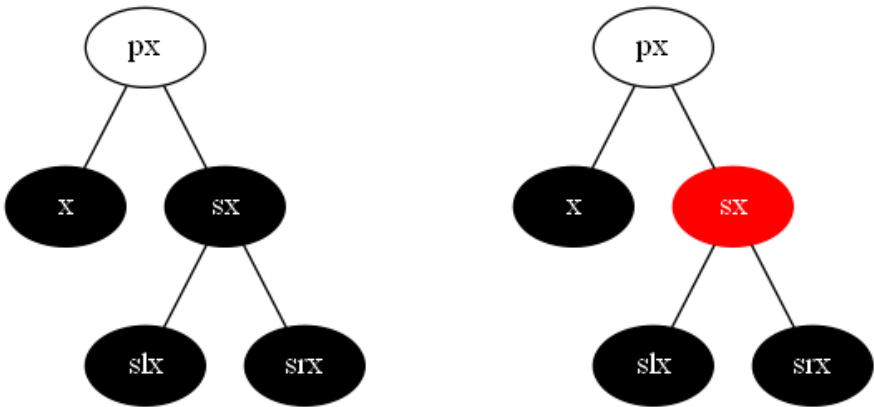
slx.color = px.color, px.color = BLACK, srx.color = BLACK, 再以px为支点左旋, **x = ROOT**, 回到LOOP (到 (1) 时RETURN)。

(情况二) sx为黑色, sx的右孩子srx为黑色, sx的左孩子slx为红色



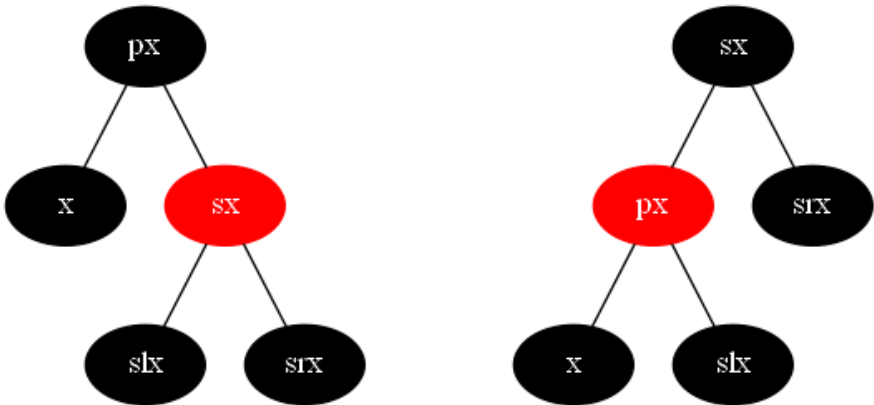
slx.color = BLACK, sx.color = RED, 再以sx为支点右旋, 进入 (特殊情况一)。

(情况三) sx为黑色, sx的右孩子srx为黑色, sx的左孩子slx为黑色



sx.color = RED, 当前节点x = px, 回到LOOP。

(情况四) sx为红色 (px一定为黑色, sx的两个孩子节点一定为黑色)



px.color = RED, sx.color = BLACK, 再以px为支点左旋, 进入 (特殊情况一) 或 (情况二) 或 (情况三)。

分类: [util](#)

好文要顶

关注我

收藏该文

Uncle Bjorney

粉丝 - 2 关注 - 0

[+加关注](#)

0 0

» 下一篇: [HashMap源码分析 \(Java8\)](#)

posted @ 2017-12-03 00:28 Uncle_Bjorney 阅读(2231) 评论(0) 编辑 收藏 举报

[刷新评论](#) [刷新页面](#) [返回顶部](#)

登录后才能查看或发表评论, 立即 [登录](#) 或者 [逛逛](#) 博客园首页

编辑推荐:

- 在 WPF 中实现融合效果
- 2>&1到底是什么意思?
- 聊聊 asp.net core 授权流程
- C# 中的那些锁, 在内核态都是怎么保证同步的?
- .NET Core Web API 类库如何内嵌运行?

最新新闻:

- 微信新增“退群可保留聊天记录”功能, 网友: 都退群还保留干什么
- 反驳黄仁勋, 英特尔CEO: 摩尔定律“依然有效”
- 传特斯拉上海工厂年底周产2.05万辆汽车, 罕见未达产
- 印媒兴奋最新苹果手机“印度制造”, 还定位印度是“下一个中国”

· 元宇宙这么能赚? 平均月薪1.8万, 深度学习月薪3万位居榜首
» 更多新闻...

Copyright © 2022 Uncle_Bjorney

Powered by .NET 6 on Kubernetes