

甜菜波波

[博客园](#)[首页](#)[新随笔](#)[联系](#)[订阅](#)[管理](#)

随笔 - 667 文章 - 76 评论 - 47

清晰理解红黑树的演变---红黑的含义

前言

红黑树，对不少人来说是个比较头疼的名字，在网上搜资料也很少有讲清楚其演变来源的，多数一上来就给你来五条定义，红啊黑啊与根节点距离相等之类的，然后就开始进行旋转、插入、删除这些操作。一通操作下来，连红色和黑色怎么来的，是什么含义，有什么作用都云里雾里的，能搞清楚就怪了。

本文介绍红黑树，暂时不涉及任何代码，只是帮助你理解红黑树的演变来源，树结构中红黑色具体含义，保证你理解了过后，再去看看什么旋转插入的东西，要清晰得多。换句话说，理解本文要描述的内容是从代码级理解红黑树的基础。

开始之前，我还是恳请你保持耐心，一步一步仔细看完，浮躁的话真的做不好任何事情。

正文

红黑树的起源，自然是二叉查找树了，这种树结构从根节点开始，左子节点小于它，右子节点大于它。每个节点都符合这个特性，所以易于查找，是一种很好的数据结构。但是它有一个问题，就是容易偏向某一侧，这样就像一个链表结构了，失去了树结构的优点，查找时间会变坏。

所以我们都希望树结构都是矮矮胖胖的，像这样：

公告

昵称：甜菜波波

园龄：8年3个月

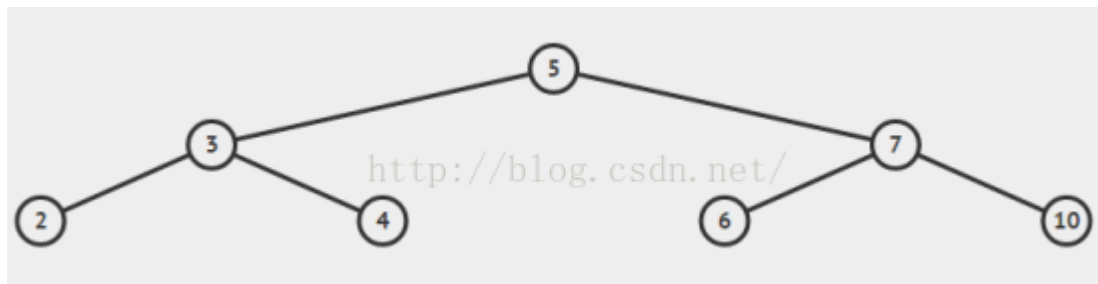
粉丝：76

关注：122

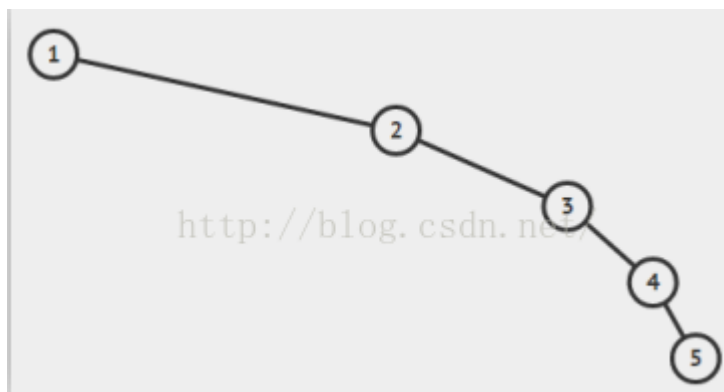
[+加关注](#)

2019年11月						
<	日	一	二	三	四	五
						六
27	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

搜索



而不是像这样：



在这种需求下，平衡树的概念就应运而生了。

红黑树就是一种平衡树，它可以保证二叉树基本符合矮矮胖胖的结构，但是理解红黑树之前，必须先了解另一种树，叫2-3树，红黑树背后的逻辑就是它。

好吧来看2-3树吧。

2-3树是二叉查找树的变种，树中的2和3代表两种节点，以下表示为2-节点和3-节点。

2-节点即普通节点：包含一个元素，两条子链接。

3-节点则是扩充版，包含2个元素和三条链接：两个元素A、B，左边的链接指向小于A的节点，中间的链接指向介于A、B值之间的节点，右边的链接指向大于B的节点。

2-节点： 3-节点：



常用链接

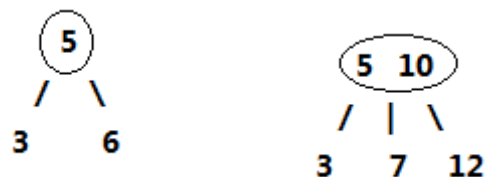
我的随笔
我的评论
我的参与
最新评论
我的标签

最新随笔

- 1.JAVA | Java对象的内存分配过程是如何保证线程安全的？
- 2.分表与分库使用场景以及设计方式
- 3.epool与select有什么区别
- 4.使用ssh-keygen生成私钥和公钥
- 5.mysql关键字冲突
- 6.MySQL 获取当前时间戳
- 7.平时常说的ThreadLocal，今天就彻底解决它
- 8.mysql和mssql最大连接数
- 9.Spring Boot实战：拦截器与过滤器
- 10.Mysql 索引问题-日期索引使用

我的标签

java(114)
jvm(29)
openfire(17)
c#(17)



在这两种节点的配合下，2-3树可以保证在插入值过程中，任意叶子节点到根节点的距离都是相同的。完全实现了矮胖矮胖的目标。怎么配合的呢，下面来看2-3树的构造过程。

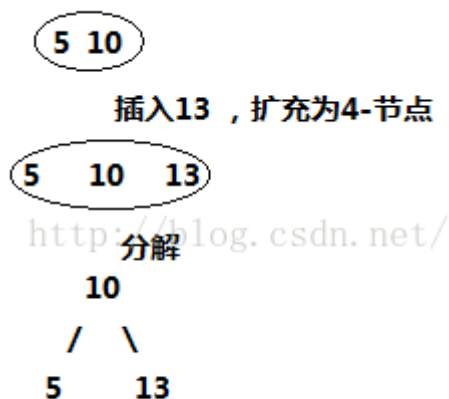
所谓构造，就是从零开始一个节点一个节点的插入。

在二叉查找树中，插入过程从根节点开始比较，小于节点值往右继续与左子节点比，大于则继续与右子节点比，直到某节点左或右子节点为空，把值插入进去。这样无法避免偏向问题。在2-3树中，插入的过程是这样的。

如果将值插入一个2-节点，则将2-节点扩充为一个3-节点。

如果将值插入一个3-节点，分为以下几种情况。

(1).3-节点没有父节点，即整棵树就只有它一个三节点。此时，将3-节点扩充为一个4-节点，即包含三个元素的节点，然后将其分解，变成一棵二叉树。



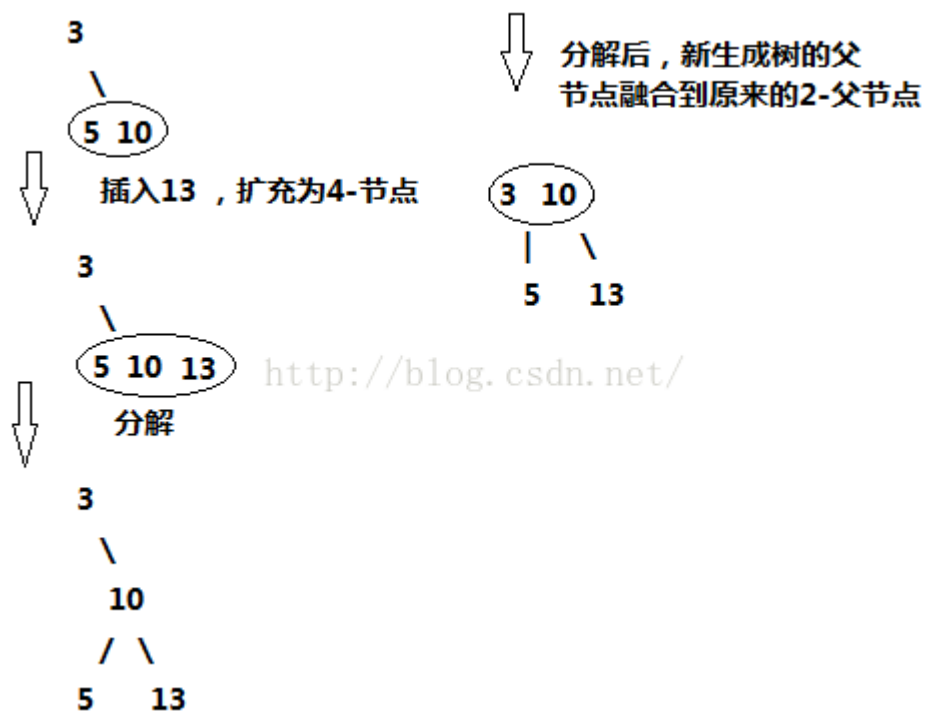
此时二叉树依然保持平衡。

mysql(14)
dubbo(13)
maven(12)
spring(12)
zookeeper(9)
linux(8)
更多

随笔分类

AngularJs(1)
Asp.NET(78)
C#(96)
DDD(5)
dubbo(13)
ElasticJob(1)
EntityFramework(8)
eureka(3)
feginclient(9)
haproxy(1)
hessian(1)
Hibernate (3)
HTML(1)
HTML5(2)
HttpClient
hystrix(8)
idea(1)
IOC(4)
Java(176)
jQuery(22)
JS(23)
jvm(36)
Linq(6)
linux(10)

(2).3-节点有一个2-节点的父节点，此时的操作是，3-节点扩充为4-节点，然后分解4-节点，然后将分解后的新树的父节点融入到2-节点的父节点中去。

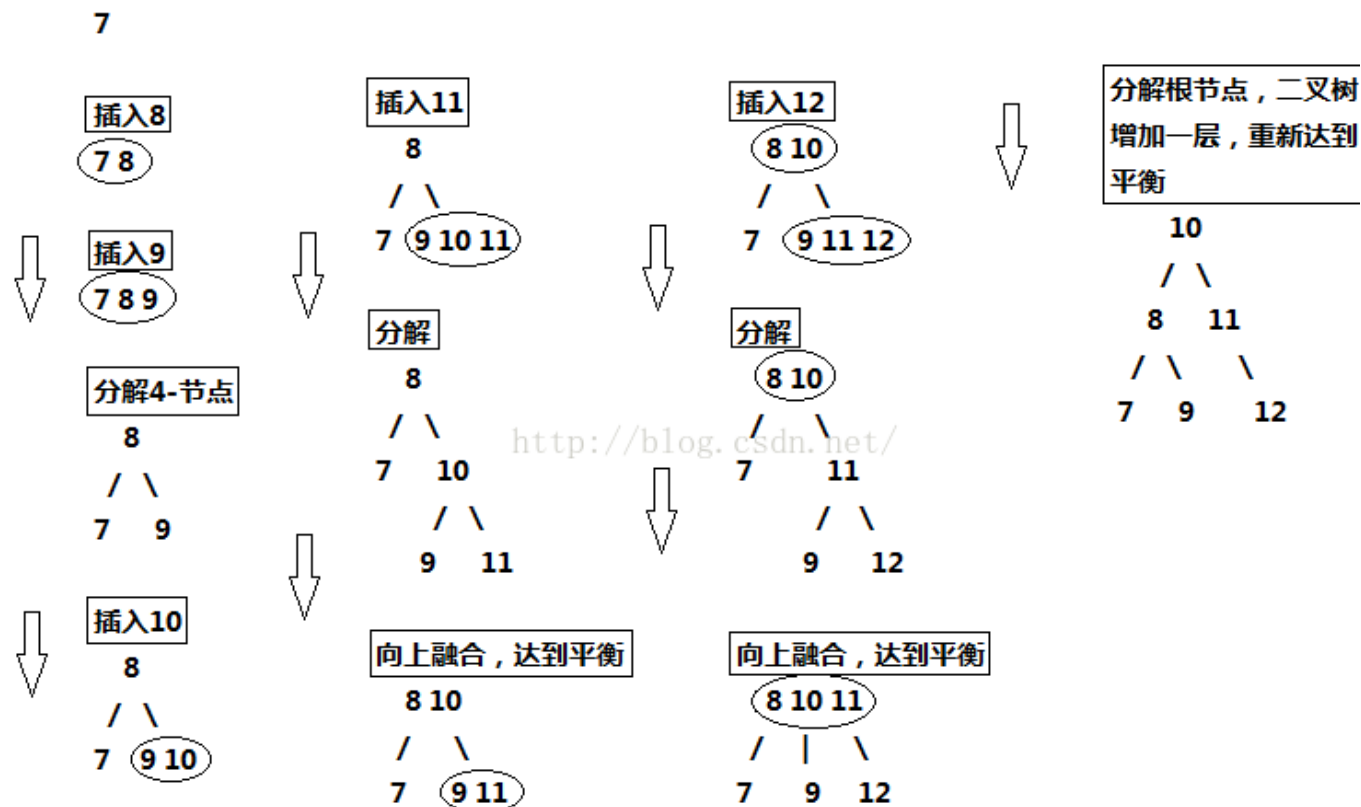


(3).3-节点有一个3-节点的父节点，此时操作是：3-节点扩充为4-节点，然后分解4-节点，新树父节点向上融合，上面的3-节点继续扩充，融合，分解，新树继续向上融合，直到父节点为2-节点为止，如果向上到根节点都是3-节点，将根节点扩充为4-节点，然后分解为新树，至此，整个树增加一层，仍然保持平衡。

第三种情况稍微复杂点，为了便于直观理解，现在我们从零开始构建2-3树，囊括上面所有的情况，看完所以步骤后，你也可以自己画一画。

我们将{7,8,9,10,11,12}中的数值依次插入2-3树，画出它的过程：

maven(10)
mina(1)
MVC(7)
mybatis(9)
mysql(15)
netty(3)
nio(2)
openfire(17)
Oracle(8)
php
RabbitMQ(4)
Redis(6)
ribbon(7)
ShardingJDBC
SOA(3)
socket(4)
Spring.NET(3)
SpringBoot And SpringCloud(31)
SpringMvc(28)
SqlServer(31)
TCP/IP(1)
tomcat(1)
TPL(2)
vue(7)
WCF(3)
WinForm(1)
zookeeper(10)
zuul(6)
操作系统(3)
常用网址(2)
反向代理(2)
分布式
分库分表(1)
高并发(2)



工具(16)

关注博客

架构(2)

面试(8)

其他(11)

设计模式(13)

算法(11)

图片识别(OCR)(2)

网络工具(2)

项目管理(6)

消息中间件(1)

随笔档案

2019年10月(4)

2019年9月(3)

2019年7月(2)

2019年1月(6)

2018年12月(33)

2018年11月(25)

2018年10月(9)

2018年9月(63)

2018年8月(22)

2018年7月(58)

2018年6月(16)

2018年5月(37)

2018年4月(20)

2018年3月(12)

2018年2月(7)

2018年1月(24)

2017年12月(7)

2017年11月(10)

2017年10月(1)

2017年9月(5)

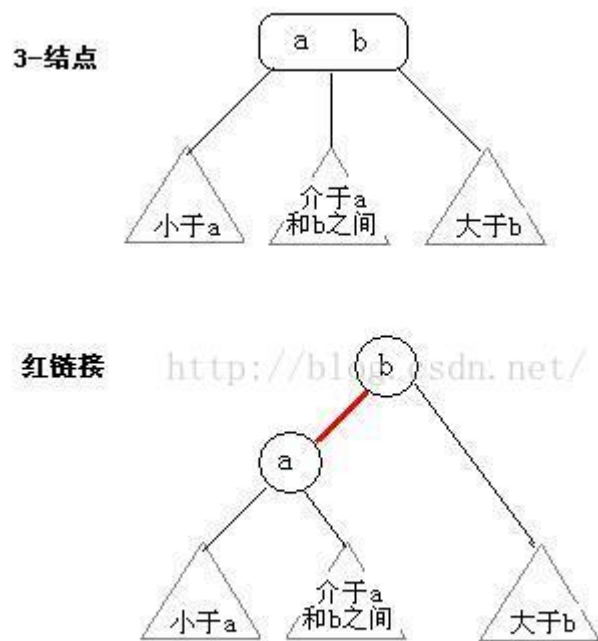
所以，2-3树的设计完全可以保证二叉树保持矮矮胖胖的状态，保持其性能良好。但是，将这种直白的表述写成代码实现起来并不方便，因为要处理的情况太多。这样需要维护两种不同类型的节点，将链接和其他信息从一个节点复制到另一个节点，将节点从一种类型转换为另一种类型等等。

因此，红黑树出现了，红黑树的背后逻辑就是2-3树的逻辑，但是由于用红黑作为标记这个小技巧，最后实现的代码量并不大。(但是，要直接理解这些代码是如何工作的以及背后的道理，就比较困难了。所以你一定要理解它的演化过程，才能真正的理解红黑树)

我们来看看红黑树和2-3树的关联，首先，最台面上的问题，红和黑的含义。红黑树中，所有的节点都是标准的2-节点，为了体现出3-节点，这里将3-节点的两个元素用左斜红色的链接连接起来，即连接了两个2-节点来表示一个3-节点。这里红色节点标记就代表指向其的链接是红链接，黑色标记的节点就是普通的节点。所以才会有那样一条定义，叫“从任一节点到其每个叶子的所有简单路径都包含相同数目的黑色节点”，因为红色节点是可

以与其父节点合并为一个3-节点的，红黑树实现的其实是一个**完美的黑色平衡**，如果你将红黑树中所有的红色链接放平，那么它所有的叶子节点到根节点的距离都是相同的。所以它并不是一个严格的平衡二叉树，但是它的综合性能已经很优秀了。

借一张别人的图来看：

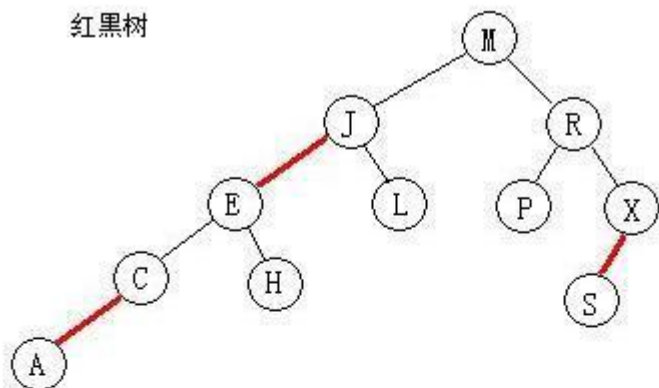


红链接只能是左链接，且由于a小于b，故b在上 为a的父结点

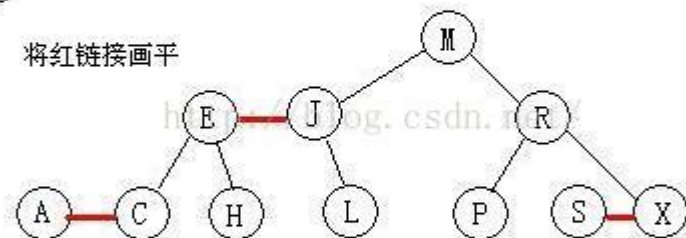
红链接放平：

2017年8月(16)
2017年7月(9)
2017年6月(15)
2017年5月(11)
2017年4月(11)
2017年3月(21)
2017年2月(7)
2017年1月(2)
2016年12月(1)
2016年11月(16)
2016年10月(2)
2016年9月(4)
2016年8月(5)
2016年7月(8)
2016年6月(4)
2016年5月(5)
2016年4月(13)
2016年3月(2)
2016年2月(1)
2016年1月(3)
2015年12月(15)
2015年11月(3)
2015年10月(10)
2015年9月(18)
2015年8月(12)
2015年7月(11)
2015年6月(6)
2015年5月(3)
2015年4月(6)
2015年1月(1)
2014年8月(1)
2014年5月(1)
2014年2月(1)
2014年1月(1)

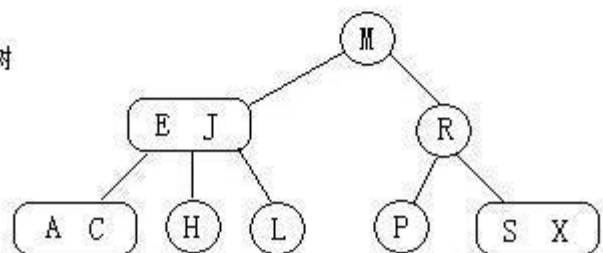
红黑树



将红链接画平



2-3树



所以，红黑树的另一种定义是满足下列条件的二叉查找树：

- (1) 红链接均为左链接。
- (2) 没有任何一个结点同时和两条红链接相连。(这样会出现4-节点)
- (3) 该树是完美黑色平衡的，即任意空链接到根结点的路径上的黑链接数量相同。

理解了这个过程以后，再去看红黑树的各种严格定义，以及其插入，删除还有旋转等操作，相信你脑子里的思路会清晰得多的。

2013年12月(4)
 2013年11月(1)
 2013年10月(3)
 2013年9月(1)
 2013年8月(3)
 2013年7月(1)
 2013年6月(2)
 2013年4月(1)
 2013年3月(4)
 2013年2月(5)
 2012年11月(4)
 2012年10月(1)
 2012年9月(1)
 2012年7月(1)
 2012年6月(1)
 2012年5月(1)
 2012年4月(1)
 2012年3月(4)
 2012年1月(1)
 2011年12月(1)
 2011年11月(7)
 2011年9月(3)
 2011年8月(7)

文章分类

Asp.NET(2)
 JQuery
 LINQ
 MVC
 php(1)
 WCF

分类: 算法

标签: 红黑树

好文要顶

关注我

收藏该文



甜菜波波

关注 - 122

粉丝 - 76

+加关注

« 上一篇: [ConcurrentHashMap的JDK1.8实现](#)» 下一篇: [红黑树从头至尾插入和删除结点的全程演示图](#)

posted @ 2018-05-22 16:41 甜菜波波 阅读(4780) 评论(4) 编辑 收藏

2

推荐

1

反对

评论列表

#1楼 2019-06-05 15:49 弱鸡徐思

为啥没有图啊，图片能发给我吗？如果可以的话发到邮箱里，非常感谢~~ 邮箱：xusinan2013@163.com

支持(0) 反对(0)

#2楼 2019-07-06 19:33 邹童

图没了？

支持(0) 反对(0)

#3楼 2019-07-17 10:01 嗯？

没图，看的我云里雾里的

支持(0) 反对(0)

最新评论

1. Re:清晰理解红黑树的演变---红黑的含义
感谢分享！

--刀尖红叶

2. Re:Synchronized修饰静态变量和普通变量的区别

看题目，我还真以为 同步的是 变量，再看内容，原来说的是 方法

--贾树丙

3. Re:为什么MySQL数据库索引选择使用B+树？

不错的总结，赞

--baronXiong

4. Re:为什么MySQL数据库索引选择使用B+树？

好文章！

--这世界很酷！

5. Re:为什么MySQL数据库索引选择使用B+树？

很透彻。感谢

我觉得，可以的话最好在红黑树的那张图补上黑色的NULL节点，这样对解释第5，6条性质更友好些。

--chen猿

阅读排行榜

1. Bootstrap TreeView使用示例(34699)

2. 为什么MySQL数据库索引选择使用B+树？ (32808)

3. 在多线程中使用静态方法是否有线程安全问题(29916)


4. \$.when().done().then()的用法(23993)

5. java通过http方式下载文件(20526)

#4楼 2019-11-03 15:37 刀尖红叶

感谢分享!

支持(0) 反对(0)

[刷新评论](#) [刷新页面](#) [返回顶部](#) 注册用户登录后才能发表评论, 请 [登录](#) 或 [注册](#), [访问](#) 网站首页。

【推荐】腾讯云海外1核2G云服务器低至2折, 半价续费券限量免费领取!

【活动】京东云服务器_云主机低于1折, 低价高性能产品备战双11

【推荐】超50万行VC++源码: 大型组态工控、电力仿真CAD与GIS源码库

【课程】马士兵老师一对一职业规划指导! 程序员突破年薪80W!

【推荐】天翼云双十一翼降到底, 云主机11.11元起, 抽奖送大礼

【课程】新鲜出炉! 阿里、腾讯、百度等大厂最新414道面试题

【推荐】流程自动化专家UiBot, 体系化教程成就高薪RPA工程师



相关博文:

- 查找(二): 彻底理解红黑树和平衡查找树
- 排序二叉树, 平衡二叉树和红黑树的概念以及相关的操作讲解
- 数据结构中的各种树浅谈
- 通过2-3-4树理解红黑树
- 红黑树-想说爱你不容易
- » 更多推荐...

评论排行榜

1. \$.when().done().then()的用法(7)
2. 为什么MySQL数据库索引选择使用B+树? (6)
3. 清晰理解红黑树的演变---红黑的含义(4)
4. Java 堆内存(3)
5. BootStrap TreeView使用示例(3)

推荐排行榜

1. 为什么MySQL数据库索引选择使用B+树? (13)
2. \$.when().done().then()的用法(11)
3. 在多线程中使用静态方法是否有线程安全问题(6)
4. 领域模型浅析(5)
5. C#中IDisposable的用法(5)



最新 IT 新闻:

- 为什么在重庆容易迷路? 大脑GPS系统呈蜂窝状, 弯路多了就“变形”
 - 像用“数据库”一样使用“大数据”! 华为宣布河图引擎开源
 - 台湾将于12月10拍卖首批5G频谱: 3.5GHz成兵家必争之地
 - 全球超算500强 美蝉联冠军 中国数量增加
 - 美航天局首次确认木卫二大气中存在水
- » 更多新闻...

历史上的今天:

2018-05-22 ConcurrentHashMap的JDK1.8实现
2018-05-22 linux常用命令
2017-05-22 c#中Monitor的使用

Copyright © 2019 甜菜波波
Powered by .NET Core 3.0.0 on Linux