

我们使用**符号表**这个词来描述一张抽象的表格,我们会将信息(值)存储在其中,然后按照指定的键来搜索并获取这些信息。键和值的具体意义取决于不同的应用。

符号表中可能会保存很多键和很多信息,因此实现一张高效的符号表也是一项很有挑战性的任务。

我们会用三种经典的数据类型来实现高效的符号表:二叉查找数、红黑树、散列表。

# 二分查找

我们使用有序数组存储键,经典的二分查找能够根据数组的索引大大减少每次查找所需的比较次数。

在查找时,我们先将被查找的键和子数组的中间键比较。如果被查找的键小于中间键,我们就在左子数组中继续查找,如果大于我们就在右子数组中继续查找,否则中间键就是我们要找的键。

一般情况下二分查找都比顺序查找快的多,它也是众多实际应用程序的最佳选择。对于一个静态表(不允许插入)来说,将其在初始 化时就排序是值得的。



## 2018/7/31 史上最简单清晰的红黑树讲解 - CSDN博客 红黑树的介绍和实现 ② 1158 一、红黑树 红黑树 (Red-Black Tree) 是二叉搜索树 (Binary Search Tree) 的一种。二叉搜索树在最坏的情.. 红黑树 从入门到精通 2404 写评论 红黑树简介红黑树是一种非常优秀的二叉搜索树,诞生于1972年,并被优化于1978年,因其效率高而被广泛... 目录 红黑树系列一:旋转 - CSDN博客 6-12 红黑树系列一:旋转红黑树一棵在每个结点上增加了一个存储位来表示结点颜色(红色RED或黑色BLACK)的二叉搜索树。二叉搜索树简单的说就是:对树中任何结点x其左 学习算法导论——红黑树旋转插入和删除 - CSDN博客 7-25 微信 红黑树是一棵二叉搜索树,每个节点有一个标志位表示颜色,该颜色可以是红(RED)或黑(BLACK)。通过对任何一条从根到叶子的简单路径上各点的颜色进行约束,就能确保没 微博

彻底搞懂红黑树 (二)

其实关于红黑树,STL源码剖析--红黑树原理详解已经写得非常好了。但套用新警察故事里的谢霆锋说的一句...

### 红黑树的性质及旋转 - CSDN博客

6-13 制,...

1、 红黑树是一种二叉查找树,但在每个节点上增加一个存储位表示结点的颜色,可以是红色或者黑色。通过对任何一条从跟到叶子的路径上各个结点着色方式的限制,...

**红黑树右旋转 - CSDN博客** 5-23

红黑树的平衡 红黑树首先是一棵二叉查找树,它每个结点都被标上了颜色(红色或黑色),红黑树满足以下5个性质:1、每个结点的颜色只能是红色或黑色。2、根结点是...

### 【数据结构】红黑树 (如何实现及怎样判断)

红黑树是一颗二叉搜索树,它在每个节点上增加了一个存储位来表示节点的颜色,可以是red或black。通过对...

介绍红黑树是一个平衡的二叉树,但不是一个完美的平衡二叉树。虽然我们希望一个所有查找都能在~lgN次比...

### 红黑树笔记——RBT简介和旋转操作 - CSDN博客

6-5

一、红黑树简介做ACM的都用过STL里面的map和set,这在STL源码里面就是用红黑树来实现的红黑树是一种自平衡二叉查找树和上篇说到的Treap不同红黑树通过对任何…

### 彻底搞懂红黑树(二) - CSDN博客

7-19

其实关于红黑树,STL源码剖析—红黑树原理详解 已经写得非常好了。但套用新警察故事里的谢霆锋说的一句话:自己查印象深一点。这里也是一样,在自己写,印象深...

### 红黑树讲解(逻辑思路透彻清晰)

网上看到的关于红黑树的讲解,思路最清楚的一个. 原文地址: http://blog.csdn.net/yang\_yulei/article/details/2...

#### 红黑树 删除某节点后 旋转3次 举例 - CSDN博客

6-17

博客转自自己的新浪博客,属于自己原创下面对红黑树删除某节点后,旋转三次的例子做出详解:先利用代码建立树如下:修改颜色分布为下列树:修改颜色分布后利用 tree...

### 学习《算法导论》第13章 <u>红黑树</u> 总结一 - CSDN博客

6-6

学习《算法导论》第13章 红黑树 总结一前面学习了二叉查找树,我们知道一棵高度为h的二叉查找树实现的任何一种动态集合操作,其时间都是O(h);但若二叉查找树…

### 【数据结构和算法05】 红-黑树 (看完包懂~)

❷ 4.2万

从第4节的分析中可以看出,二叉搜索树是个很好的数据结构,可以快速地找到一个给定关键字的数据项,并...

太松 / 八市 L目然台注帐码行面针出机

登录 注册 ×

## 2018/7/31 史上最简单清晰的红黑树讲解 - CSDN博客 红黑树的旋转、查找和删除(附源代码) - CSDN博客 6-21 Red Black Tree Basic 红黑树的节点声明,其中Parent指针是指向某一节点的父节点的指针: typedef struct TreeNode \*PtrRBTNode; typedef struct TreeNode RBTNode; ... ◎ 7659 写评论 红黑树原理解析以及Java实现 红黑树本文的主要内容: 1、红黑树的基本概念以及最重要的5点规则。 2、红黑树的左旋转、右旋转、重新着... 目录 红黑树 ◎ 6.6万 介绍另一种平衡二叉树: 红黑树 (Red Black Tree), 红黑树由Rudolf Bayer于1972年发明, 当时被称为平衡... 收藏 一步一图一代码,一定要让你真正彻底明白红黑树 ◎ 14.4. 微信 一步一图一代码,一定要让你真正彻底明白红黑树 作者: July 二零——年一月九日 ----微博 清晰理解红黑树的演变---红黑的含义 ⊚ 9062 QQ 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... 彻底搞懂红黑树 (一) @ ②21万 红黑树和c++ 虚拟继承内存分布 几乎成了我的死敌,因为完全没用过,所以导致每次看懂了之后都忘了(也许... 红黑树性质的彻底理解 ⊚ 301 一、红黑树的介绍 先来看下算法导论对R-B Tree的介绍: 红黑树,一种二叉查找树,但在每个结点上增加一个... 红黑树的删除-10张图学会 1848 如果要学红黑树,那么就必须知道二叉树,毕竟红黑树本身就是一个二叉搜索树。只不过红黑树比一般的二叉... 教你初步了解红黑树 ◎ 30.9万 教你透彻了解红黑树作者: July、saturnman 2010年12月29日本文参考: Google、算法导论、STL源码剖析... 迭代器失效的几种情况总结 ⊚ 6443 1. 对于序列式容器(如vector,deque),序列式容器就是数组式容器,删除当前的iterator会使后面所有元素的iter... (第12讲) 234树和2-3树 ⊚ 1265

2-3-4树: 2-3-4 树在计算机科学中是阶为4的B树。它可以 2-3-4树 在O(log n)时间内查找、插入和删除。 2-3-4 ...

算法之红黑树 ◎ 1.6万

红黑树(一) 原理和算法详细介 1 R-B Tree简介 R-B Tree, 全称是Red-Black Tree, 又称为"红黑树", 它一种...

二叉树-详解二叉排序树 ◎ 2.2万

一、二叉搜索树首先二叉排序树也是一棵二叉树,所谓二叉树,就是"任何节点最多只允许两个子节点",这两...



登录 注册 积分: 2390 排名: 2万+

勋章: 📵

最新文章

OpenGL

opengl 库的说明和简单使用

C++重载++如何区分"++"重载的前置与后置

虚析构函数的作用

浅谈构造函数与析构函数的调用顺序

个人分类

 stl
 8篇

 dp
 11篇

 前端
 10篇

 机器学习
 32篇

归档

2017年3月 3篇 2017年1月 3篇 2016年12月 3篇 2016年11月 2篇 2016年10月 22篇

热门文章

欧拉函数求法与应用

阅读量: 17633

python学习之 12306的一个小爬虫

阅读量: 8858

python学习之 beautifulsoup选择器

阅读量: 7925

python学习之 requests爬虫导致的中文乱码

阅读量: 6164

强连通分量及缩点tarjan算法解析

阅读量: 4481

最新评论

欧拉函数求法与应用

u011526967: [reply]c\_circle[/reply] 原来如此,我

就说怎么都算不对

欧拉函数求法与应用

finalcsdn:妙啊,感谢博主的分享

Python学习之 PrettyT...

qq\_42303465: for item in soup('p',class\_='wt\_fc\_

c0\_i\_date'): ...

强连通分量及缩点tarjan算法解析

sdsy191553:大佬,能否借用下这篇文章,写在自己的心得体会中去?其实就是拉个链接。

欧拉函数求法与应用

qq\_21436421: 博主, HDOJ2588 "由于b>=d且 b, d互质, 所以这个数目就是φ (b) -1" ...

联系我们

0

写评论

目录

收藏

微信

微博

QQ

登录 注册 ×



▲ QQ客服 ● 客服论坛

关于 招聘 广告服务 网站地图 ©2018 CSDN版权所有京ICP证09002463号

☆ 百度提供支持

经营性网站备案信息

网络110报警服务

中国互联网举报中心

北京互联网违法和不良信息举报中心

0

写评论

目录

收藏

微信

微博

QQ

登录 注册 ×