

<<高级数据结构>>

图书基本信息

书名：<<高级数据结构>>

13位ISBN编号：9787564136512

10位ISBN编号：7564136510

出版时间：2012-7

出版时间：林厚从、沈军、李立新、王晓敏 东南大学出版社 (2012-07出版)

作者：林厚从 著

页数：444

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

<<高级数据结构>>

内容概要

《高级数据结构》在基本数据结构的基础上，围绕一些常用的高级数据结构，结合大量实战例题，深入分析“数据结构是如何服务于算法的”。

本书主要内容包括：哈希表、树与二叉树、优先队列与堆、并查集、线段树、树状数组、伸展树、Treap、AVL树、红—黑树、SBT、块状链表与块状树、后缀树与后缀数组、树链剖分与动态树等。

《高级数据结构》的适用对象包括：中学信息学竞赛选手及辅导老师、大学ACM比赛选手及教练、高等院校计算机专业的师生、程序设计爱好者等。

书籍目录

第1章 哈希表 1.1哈希表的基本原理 1.2哈希表的基本概念 1.3哈希函数的构造 1.4哈希表的基本操作 1.5冲突的处理 1.6哈希表的性能分析 1.7哈希表的应用举例 1.8本章习题 第2章 树与二叉树 2.1树 2.1.1树的存储结构 2.1.2树的遍历 2.2二叉树 2.2.1普通树转换成二叉树 2.2.2二叉树的遍历 2.2.3二叉树的其他操作 2.2.4二叉树的形态 2.3二叉排序树 2.4哈夫曼二叉树 2.5字典树 2.6本章习题 第3章 优先队列与二叉堆 3.1优先队列 3.2二叉堆 3.2.1Put操作 3.2.2Get操作 3.3可并堆 3.3.1左偏树的定义 3.3.2左偏树的基本操作 3.4本章习题 第4章 并查集 4.1并查集的主要操作 4.2并查集的实现 4.2.1并查集的数组实现 4.2.2并查集的链表实现 4.2.3并查集的树实现 4.3并查集的应用举例 4.4本章习题 第5章 线段树 5.1线段树的应用背景 5.2线段树的初步实现 5.2.1线段树的结构 5.2.2线段树的性质 5.2.3线段树的存储 5.2.4线段树的常用操作 5.2.4.1线段树的构造 5.2.4.2线段树的查询 5.2.4.3线段树的修改 5.2.4.4线段树的延迟修改 5.3线段树在一些经典问题中的应用 5.3.1逆序对问题 5.3.2矩形覆盖问题 5.4线段树的扩展 5.4.1用线段树优化动态规划 5.4.2将线段树扩展到高维 5.4.3线段树与平衡树的结合 5.5线段树与其他数据结构的比较 5.6线段树的应用举例 5.7本章习题 第6章 树状数组 6.1树状数组的问题模型 6.2树状数组的基本思想 6.3树状数组的实现 6.3.1子集的划分方法 6.3.2查询前缀和 6.3.3修改子集和 6.4树状数组的常用技巧 6.4.1查询任意区间和 6.4.2利用SHLL数组求出原数组a的某个元素值 6.4.3找到某个前缀和对应的前缀下标index 6.4.4成倍扩张/缩减 6.4.5初始化树状数组 6.5树状数组与线段树的比较 6.6树状数组扩展到高维的情形 6.7树状数组的应用举例 6.8本章习题 第7章 伸展树 7.1伸展树的主要操作 7.1.1伸展操作 7.1.2伸展树的基本操作 7.2伸展树的算法实现 7.3伸展树的效率分析 7.4伸展树的应用举例 7.5本章习题 第8章 Treap 8.1Treap的基本操作 8.2Treap的算法实现 8.3Treap的应用举例 8.4本章习题 第9章 平衡树 9.1AVL树 9.2红—黑树 9.3SBT 9.3.1SBT的基本操作 9.3.2SBT的效率分析 9.3.3SBT的算法实现 9.4本章习题 第10章 块状链表与块状树 10.1块状链表的基本思想 10.2块状链表的基本操作 10.3块状链表的扩张 10.3.1维护区间和以及区间最值 10.3.2维护局部数据有序化 10.3.3维护区间翻转 10.4块状链表与其他数据结构的比较 10.5分块思想在树上的应用——块状树 10.6块状链表的应用举例 10.7本章习题 第11章 后缀树与后缀数组 11.1后缀树的简介 11.2后缀树的定义 11.3后缀树的构建 11.3.1后缀树的朴素构建算法 11.3.2后缀树的线性时间构建算法 11.3.2.1隐式树的朴素构建 11.3.2.2扩展规则约定 11.3.2.3后缀链加速 11.3.2.4进一步加速 11.3.2.5后缀树拓展到多串的形式 11.3.2.6代码实现 11.3.2.7相关证明 11.4后缀树的应用 11.4.1字符串（集合）的精确匹配 11.4.1.1情形一 11.4.1.2情形二 11.4.1.3情形三 11.4.1.4情形四 11.4.2公共子串问题 11.4.2.1情形五 11.4.2.2情形六 11.4.2.3情形七 11.4.2.4情形八 11.4.2.5情形九 11.4.3重复子串问题 11.4.3.1情形十 11.4.3.2情形十一 11.4.3.3情形十二 11.5后缀数组的简介 11.6后缀数组的定义 11.7后缀数组的构建 11.7.1一种直接的构建算法 11.7.2倍增算法 11.7.2.1倍增算法描述 11.7.2.2倍增算法代码 11.7.3由后缀树得到后缀数组 11.7.4DC3算法和DC算法 11.7.4.1DC3算法 11.7.4.2DC算法 11.8LCP的引入 11.9后缀数组的应用 11.9.1后缀排序的直接应用 11.9.1.1Burrows—Wheeler变换 11.9.1.2多模式串的匹配 11.9.2通过引入LCP优化 11.9.2.1多模式串的匹配 11.9.2.2重复子串问题 11.9.2.3最长回文子串 11.9.2.4最长公共子串 11.9.3后缀数组的应用举例 11.10本章习题 第12章 树链剖分与动态树 12.1树链剖分的思想和性质 12.2树链剖分的实现及应用 12.3动态树的初探 12.3.1动态树的常用功能 12.3.2动态树的简单情形 12.4动态树的实现 12.4.1动态树的基本操作及其实现 12.4.1.1动态树的问题模型 12.4.1.2用Splay维护实路径 12.4.2动态树操作的时间复杂度分析 12.4.2.1动态树操作的次数 12.4.2.2Splay操作的平摊时间 12.5动态树的经典应用 12.5.1求最近公共祖先 12.5.2并查集操作 12.5.3求最大流 12.5.4求生成树 12.6动态树的应用举例 12.7本章习题 致谢

章节摘录

版权页：插图：并查集（Union Find Set）是一种用于处理分离集合的抽象数据类型。

当给出两个元素的一个无序对（ a, b ）时，需要快速合并 a 和 b 分别所在的集合，这期间需要反复查找某元素所在的集合，“并”、“查”和“集”三字由此而来。

也就是说，并查集的作用是动态地维护和处理集合元素之间的复杂关系。

在并查集中， n 个不同的元素被分为若干组，每组是一个集合，这种集合就叫做“分离集合（Disjoint Set）”。

并查集支持查找一个元素所属的集合以及两个元素各自所属集合的合并操作。

例如，有这样一个问题：一个城镇里居住着 n 个市民，已知一些人互为朋友，而且朋友的朋友也是朋友，也就是说，如果 A 和 B 是朋友， C 和 B 是朋友，则 A 和 C 也是朋友，请你根据给出的若干组朋友关系，求出最大的一个朋友圈的人数。

这就有了并查集的用武之地了；一开始我们把所有人都各自放在一个集合中，然后根据依次给出的朋友关系，查找判断两个人是否属于同一个集合（是否已经是朋友），如果不在同一个集合，则将这两个集合合并成一个集合（形成一个朋友圈），最后看哪个集合的元素最多并输出个数即可。

4.1并查集的主要操作 使用并查集首先要记录一组分离的动态集合 $S=\{S_1, S_2, \dots, S_k\}$ ，每个集合还要设置一个代表来识别，代表只要选择该集合中的某个元素（成员）即可，哪一个元素被选作代表是无所谓的，重要的是，如果请求某一动态集合的代表两次，且在两次请求间不修改集合，则两次得到的答案应该是相同的。

并查集主要有三种操作：初始化、查找与合并。

（1）初始化：MAKE—SET（ x ）建立一个新的集合，其仅有的成员是 x （同时就是代表）。

由于各集合是分离的，所以要求 x 没有在其他集合中出现过。

使用并查集前都需要执行一次初始化操作，无论采用何种实现方式，其时间复杂度都是 $O(n)$ 。

（2）查找：FIND—SET（ x ）查找一个元素所在的集合，本操作返回一个包含 x 的集合的代表。

查找是并查集的核心操作，也是优化并查集效率的重点。

（3）合并：UNION（ x, y ）将包含 x 和 y 的动态集合（假设为 S_x 和 S_y ）合并成一个新的集合 S ，本操作返回集合 $S_x \cup S_y$ 的代表。

一般来说，在不同的实现中通常都以 S_x 或者 S_y 的代表作为新集合的代表。

合并之前一般要先判断两个元素是否属于同一集合，这可以通过查找操作来实现。

<<高级数据结构>>

编辑推荐

《青少年信息学奥林匹克竞赛实战辅导丛书:高级数据结构》的适用对象包括：中学信息学竞赛选手及辅导老师、大学ACM比赛选手及教练、高等院校计算机专业的师生、程序设计爱好者等。

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>