数据结构与算法

实验报告

实验题目：实验五题目一 实现二叉排序树（及交互功能）

姓 名：刘帅

学 号：2020212267

日 期：2021年6月14日

自我评分：【 A+ 】

自我评分说明：A+，A，B+，B，B-，C，D，分别对应分数95、90、85、80、75、70、60

诚信声明

本人郑重承诺：本实验程序和实验报告均是本人独立学习和工作所获得的成果。尽我所知，实验报告中除特别标注的地方外，不包含其他同学已经发表或撰写过的成果；实验程序中对代码工作的任何帮助者所作的贡献均做了明确的说明，并表达了谢意。

如有抄袭，本人原因承担因此而造成的任何后果。

特此声明。

签名：刘帅

日期：2021年6月10

日

程序引用说明

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 引用项 | 来源 | 相同代码行数 |
| 1 | 排序树类的声明 | 课本 | 7 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| 小计 | | |  |

总代码行数123; 引用占比\_\_\_\_\_\_6%\_\_\_\_\_\_\_\_\_\_

1、实验简介

本实验为二叉排序树及其功能实现的实验，实现建立、查找、插入、删除、销毁、打印的操作，为了数据存储的一致性及操作过程可视化、交互化，作者采用了交互式系统，实现排序树的功能。

2、程序框架

1. 与构造相关的函数：

bstree(int\* arr, int n);//建立

2．与操作有关函数：

int search(binode\* r, int key);//查找

void insertbst(binode\*& r,binode\*s);//插入

void deletenode(binode\*& r);//删除结点

bool deletekey(binode\*& r, int key);//删除值为key的结点

void release(binode\* r);//释放节点

void print(binode\*r);//打印

3、关键代码实现

【哪些函数是你认为最能体现自己工作成果的函数，说明函数实现基本思想（可用文字或图表示），以及具体的实验步骤（用伪代码或带注释代码）】

图形用户界面, 文本

描述已自动生成3.1 结构体binode的定义

文本

描述已自动生成3.2 void insertbst(binode\*& r,binode\*s);//插入

利用递归的方法，将数组中的元素插入到排序树的结构体内存中，遵循左子树结点小于根节点小于右子树结点的原则。

3.3 构造函数 bstree(int\* arr, int n);//建立

利用inserbst插入函数，实现整个数组的存入。文本

描述已自动生成

3.4 void deletenode(binode\*& r);//删除结点

如果删除的结点为父节点，则相应指针需要进行移动操作。文本

描述已自动生成

文本

描述已自动生成3.5int bstree::search(binode\* r, int key) {

从根结点从左至右递归的查找相应元素。

文本

描述已自动生成3.6可视化显示

4、不足

无

5、心得体会

了解了二叉排序树的原理及相关操作。

数据结构与算法

实验报告

实验题目：实验五题目二 实现平衡二叉树 （及交互功能）

姓 名：刘帅

学 号：2020212267

日 期：2021年6月10日

自我评分：【 A+ 】

自我评分说明：A+，A，B+，B，B-，C，D，分别对应分数95、90、85、80、75、70、60

诚信声明

本人郑重承诺：本实验程序和实验报告均是本人独立学习和工作所获得的成果。尽我所知，实验报告中除特别标注的地方外，不包含其他同学已经发表或撰写过的成果；实验程序中对代码工作的任何帮助者所作的贡献均做了明确的说明，并表达了谢意。

如有抄袭，本人原因承担因此而造成的任何后果。

特此声明。

签名：刘帅

日期：2021年6月10日

程序引用说明

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 引用项 | 来源 | 相同代码行数 |
| 1 | 排序树类的声明 | 课本 | 7 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| 小计 | | |  |

总代码行数236; 引用占比\_\_\_\_\_\_2.3%\_\_\_\_\_\_\_\_\_\_

1、实验简介

本实验使用二叉链表实现平衡二叉树，并实现建立、查找、插入、删除、销毁操作，同时，为满足数据的操作性及交互性，采用了交互系统。

2、程序框架

void search(bnode\* r, int x);//查找num为x的结点

avltree(int\* arr, int n);//二叉树建立

void insert(bnode\*& r, int x);//插入操作

void print(bnode\* r);

bool deletekey(bnode\*& r, int key);//关键词删除

bnode\* root;

void release(bnode\* r);//节点释放

void deletenode(bnode\*& r);//节点删除

int getheight(bnode\* r) {//获取树的高度

void update(bnode\* r) {//更新树的高度

void leftrotate(bnode\*& r) {//树的左旋操作

void rightrotate(bnode\*& r) {//树的右旋操作

int getbf(bnode\*r) {//计算平衡因子

~avltree();

3、关键代码实现

文本, 聊天或短信

描述已自动生成3.1 void leftrotate(bnode\*& r) { //左旋操作

左旋操作需要让右孩子节点的左子树成为父节点的右子树，并使父节点成为子节点的左子树，交互孩子节点与父节点的位置。

3.2 void rightrotate(bnode\*& r) {//右旋操作

文本, 聊天或短信

描述已自动生成右旋操作与左旋操作类似，是交换左孩子结点与父节点的一种操作。使左孩子的右子树成为父节点的左子树，而后让父节点成为孩子节点的左子树，交换孩子结点与父节点的位置。

图形用户界面, 文本, 应用程序

描述已自动生成3.3 bnode\* newnode(int k) {//新建结点 录入权值为k

新建结点 重置其高度，并录入权值为k。

文本

描述已自动生成3.4 void avltree::insert(bnode\*& r, int x)//插入操作

插入操作是满足排序树的前提下对于平衡因子进行调整的操作，当目标节点的左孩子的平衡因子为1时，树型为LL型，为-1时为LR型。对于LL型，将左孩子的左子树看作整体，以目标节点为根进行右旋操作。同理，对于LR型，以左孩子节点为根进行左旋操作，而后便可将树型转换成LL型。

同理，当目标结点的平衡因子为-1时为RR型，为1时为RL型。对于RR型来说，将目标节点的左子树看作整体，以目标节点为根进行左旋操作；同理对于RL型，以右孩子结点为根进行右旋操作，即可把情况转化为RR型，再按上述方法进行左旋操作。

文本

描述已自动生成3.5删除操作

在删除操作上，作者依赖了bitree.h的文件，具体想法是在建立avl的同时建立一棵排序树，起到增添、删除的拷贝效果，对于平衡树的每次删除操作，都需要先将平衡树析构，再将排序树的元素存回数组，而后通过insert函数再次添加到平衡树中，从而避免了繁琐的平衡因子调整操作，而以时间复杂度的提高作为了代价。

文本

描述已自动生成3.6可视化

4、不足

1、在平衡树的删除操作中，利用了排序树的建立，在实验之余应思考平衡树的常规删除操作，避免空间和时间的浪费。

5、心得体会

了解了平衡二叉树的概念并实现其功能。