武汉纺织大学

数学与计算机学院

数据结构实验报告

2019 ~2020学年第一学期

|  |  |
| --- | --- |
| 专 业 班 级 | 物联网二班 |
| 学生学号 | 1802220113 |
| 学生姓名 | 梁少伟 |
| 任 课 教 师 | 叶鹏 |
| 实验指导教师 | 叶鹏 |
| 实验地点 | 实训楼 |
| 填写时间 | 2019/12/8 |
|  | |

**填写说明**

1. 报告需要学生填写的内容包括：实验要求、完整的实验结果记录、实验结果分析、实验总结。实验评价由教师填写。
2. 各部分填写内容的要求
   * **实验要求**主要填写源代码。
   * **实验结果记录**中主要反映程序运行后的效果。
   * **实验结果分析**中主要就程序运行结果是否符合预期进行分析，对于达不到预期的分析其产生的原因。
   * **实验总结**中主要阐述在实验中遇到了哪些问题，如何解决的。
3. 实验报告及源代码的提交方式
   * 报告和源代码一并发送到trustie平台。
   * 附件中应有两个文件：一是word版实验报告，**实验报告的命名规则**是：“学号\_姓名\_班级\_实验X”。另一个是压缩文件。它是将整个工程压缩成一个文件，这个**压缩文件的命名规则**是：“学号\_姓名\_班级\_实验X\_源代码”。

**实验7 二叉排序树**

**一、实验目的**

本次实验目的是掌握二叉排序树的特点，熟悉针对二叉排序树的各种操作。

**二、实验内容**

已知一个长度为 15 的线性表，其关键字序列为 {19，38，12，40，41，39，54，76，35，47，80，14，9，44，60}，编程实现如下功能：

①按各元素的顺序构造一棵二叉排序树

②中序遍历二叉排序树

③删除 54

④中序遍历二叉排序树

⑤销毁二叉排序树

**三、实验要求**  
1.认真阅读和掌握本实验的算法。  
2.写出程序并上机运行本程序。

（源程序）

“head.h”头文件

#include"stdio.h"

#include"malloc.h"

#include"iostream"

using namespace std;

typedef int elemtype;

typedef struct Node

{

elemtype data;

struct Node\* Lchild, \*Rchild;

}BISTreeNode,\*BISTree;

bool SearchBSTree(BISTree T, elemtype key, BISTree pre, BISTree& p);/////搜索树中是否有该元素

bool InsertNode(BISTree& T, elemtype KEY);//////插入元素

void Show(BISTree T);////打印树

void deletdata(BISTree& q);////删除树树中元素

void DestroyBiTree(BISTree& T);/////摧毁树

void panduan(BISTree T);////判断树是否为空

“fun.cpp”功能函数

#include"head.h"

bool SearchBSTree(BISTree T, elemtype key, BISTree pre, BISTree& p)//////////搜索元素

{

if (!T)

{

p = pre;

return false;

}

else if (T->data == key)

{

p = T;

return true;

}

else if (key < T->data)

{

pre = T;

return SearchBSTree(T->Lchild, key, pre, p);

}

else

{

pre = T;

return SearchBSTree(T->Rchild, key, pre, p);

}

}

bool InsertNode(BISTree& T, elemtype KEY)//////////////////插入元素

{

BISTree p;

BISTree s;

if (!SearchBSTree(T, KEY, NULL, p))

{

s = (BISTree)malloc(sizeof(BISTreeNode));

if (!s)exit(1);

s->data = KEY;

s->Lchild = s->Rchild = NULL;

if (!T)

T = s;

else if (KEY > p->data)

p->Rchild = s;

else

p->Lchild = s;

return true;

}

else

{

return false;

}

}

void Show(BISTree T)////打印树

{

if (T)

{

Show(T->Lchild);

cout << T->data<<" ";

Show(T->Rchild);

}

}

void panduan(BISTree T)/////判断树是否为空

{

if (T)

cout << "树不为空 " ;

else

cout<< "树为空 " << endl;

}

void deletdata(BISTree& q)//////删除树中元素

{

BISTree p;

BISTree s;

if (!q->Lchild)

{

p = q;

q = q->Rchild;

delete p;

}

if (!q->Rchild)

{

p = q;

q = q->Lchild;

delete p;

}

else

{

p = q;

s = q->Lchild;

while (s->Rchild)

{

p = s;

s = s->Rchild;

}

q->data = s->data;

if (q == p)

q->Lchild = s->Lchild;

else

p->Rchild = s->Lchild;

delete s;

}

}

void DestroyBiTree(BISTree & T)////摧毁树

{

if (T) {

if (T->Lchild) DestroyBiTree(T->Lchild);

if (T->Rchild) DestroyBiTree(T->Rchild);

free(T);

T = NULL;

}

}

“BISTree.cpp”主函数

#include"head.h"

int main()

{

int n[15] = {19,38,12,40,41,39,54,76,35,47,80,14,9,44,60};

BISTree t;

t= (BISTree)malloc((16)\*sizeof(BISTree));

t->data = n[0];

t->Lchild = t->Rchild = NULL;

for (int i = 1; i < 15; i++)

{

InsertNode(t, n[i]);

}

panduan(t);

Show(t);

cout << endl;

BISTree p;

SearchBSTree(t, 54, NULL, p);

deletdata(p);

panduan(t);

Show(t);

cout << endl;

DestroyBiTree(t);

panduan(t);

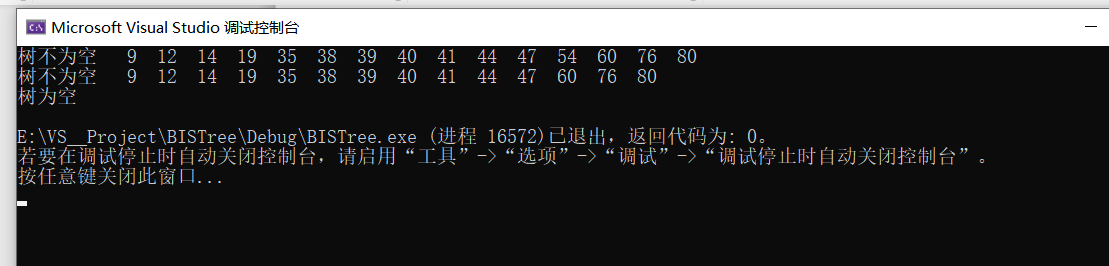
Show(t);

return 0;

}

1. **完整的实验结果记录**

**实验结果正确：**



1. **实验结果分析**

**实验结果正确**

**六、实验总结**

**这次实验相比前几次实验比较简单，没遇到太大的问题；**

**七、实验评价**（教师）