天津理工大学中环信息学院

课程教案

2018 ～ 2019 学年 第 一 学期

教 学 单 位 计算机工程系

教 研 室 计算机科学与技术教研室

课 程 名 称 数据结构

主 讲 教 师 史英杰

职 称 / 职 务 讲师

使 用 教 材 数据结构教程—李春葆

课程教案32

|  |  |  |
| --- | --- | --- |
| 授课题目（教学章、节或主题）：  **查找算法应用-二分查找题目** | 课时安排 | 2学时 |
| 教学的和要求（分掌握、熟悉、了解三个层次）：  1．掌握：在实际问题背景下灵活运用二分查找  2．熟悉：二叉树的特点  3．了解： 二叉树的概念 | | |
| 教学内容（包括基本内容、重点、难点）：  1．基本内容：二叉树的概念、二叉树的特点、在实际问题背景下灵活运用二分查找  2．重点：在实际问题背景下灵活运用二分查找  3．难点：在实际问题背景下灵活运用二分查找 | | |
| 讲课进程和时间分配  １、问题描述：从键盘读入一串整数构造一棵二叉排序树，并对得到的二叉排序述进行中序遍历，得到有序序列。  ２、基本要求：（1）构造一棵二叉树；（2）判断是否为二叉树；（3）利用中序遍历算法，输出中序序列。  3、示例数据：（4,9,0,1,8,6,3,5,2,7）  4、选作内容：  　　（１）、二分查找算法，查找关键字  　　（２）、分块查找算法，查找关键字 | | |
| 讨论、思考题、作业：  练习：  作业：  思考题：  讨论： | | |
| 授课类型（请打√）：理论课√ 讨论课□ 实践课□ 练习课□ 其他□ | | |
| 教学方式（请打√）：传统讲授√ 讨论□ 翻转□ 示教□ 指导□ 其他□ | | |
| 教学资源（请打√）：多媒体√ 模型□ 实物□ 挂图□ 音像□ 其他□ | | |

#include<stdio.h>

#include<stdlib.h>

#define MAX 20

#define YES true

#define NO false

typedef int KeyType; //数据类型

typedef struct BSTNode{

KeyType data;

struct BSTNode \*LChild;

struct BSTNode \*RChild;

}BSTNode,\*BSTree;

/\*\*

\* 向下遍历查找给定结点的相邻节点,以便插入指定节点

\*\*/

void SearchBSTreeNode(BSTree &ROOT,BSTree &node){

if(ROOT == NULL)

{

return;

}

if(ROOT->data > node->data)

{

SearchBSTreeNode(ROOT->LChild,node); //递归遍历搜索

if(ROOT->LChild == NULL)

{

ROOT->LChild = node;

}

}

if(ROOT->data < node->data)

{

SearchBSTreeNode(ROOT->RChild,node);

if(ROOT->RChild == NULL){

ROOT->RChild = node;

}

}

}

/\*\*

\* 插入指定节点node

\*\*/

void Insert(BSTree &ROOT,BSTree &node){

if(ROOT==NULL)

{

ROOT = node;

}

else{

SearchBSTreeNode(ROOT,node);

}

}

bool IsBSTree(BSTree node)

{

if(node == NULL) return YES;

if( node->LChild == NULL && node->RChild == NULL ) return YES;

if( node->LChild != NULL && node->RChild == NULL )

{

if(node->LChild->data > node->data) return NO;

else return IsBSTree(node->LChild);

}

if( node->LChild == NULL && node->RChild!=NULL)

{

if( node->RChild->data < node->data ) return NO;

else return IsBSTree(node->RChild);

}

if( node->RChild != NULL && node->LChild != NULL)

{

if(( node->LChild->data > node->data ) || ( node->RChild->data < node->data ))

return NO;

else

return ( IsBSTree(node->LChild) && IsBSTree(node->RChild) );

}

}

/\*\*

\* 二叉排序树创建

\*\*/

void CreatBSTree(BSTree &ROOT,KeyType Key[],int Key\_Num)

{

for(int i=0;i<Key\_Num;i++)

{

BSTree s = (BSTree)malloc(sizeof(BSTNode));

s->data = Key[i];

s->LChild = NULL;

s->RChild = NULL;

Insert(ROOT,s);

}

}

/\*\*

\* 中序打印二叉树

\*\*/

void Inorder(BSTree &node)

{

if(node == NULL) return;

Inorder(node->LChild);

printf("%d ",node->data);

Inorder(node->RChild);

}

/\*\*

\* 树形打印哈夫曼树

\*\*/

void Tree\_Print(BSTree &node,int level)

{

static FILE \*fp=fopen("TreePrint.txt","w");

if(node!=NULL)

{

switch(level)

{

case 0:break;

case 1:printf("|\_\_\_\_\_");fprintf(fp,"|\_\_\_\_\_");break;

case 2:printf("|\_\_\_\_\_\_|\_\_\_\_\_");fprintf(fp,"|\_\_\_\_\_\_|\_\_\_\_\_");break;

case 3:printf("|\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_");fprintf(fp,"|\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_");break;

case 4:printf("|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_");fprintf(fp,"|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_");break;

case 5:printf("|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_");fprintf(fp,"|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_");break;

default:break;

}

level++;

printf("%d \n",node->data);

fprintf(fp,"%d \n",node->data);

Tree\_Print(node->LChild,level);

Tree\_Print(node->RChild,level);

}

return;

}

int main()

{

int Key\_Num;

KeyType Key[MAX];

BSTree ROOT = NULL;

printf("请输入整数集大小：\n");

scanf("%d",&Key\_Num);

for(int i=0;i<Key\_Num;i++)

{

printf("请输入第%d个整数：",i+1);

scanf("%d",&Key[i]);

}

CreatBSTree(ROOT,Key,Key\_Num);

printf("该二叉树的中序遍历为：\n ");

Inorder(ROOT);

printf("\n");

Tree\_Print(ROOT,0);

if(IsBSTree(ROOT))

{

printf("\n这是一颗二叉树排序树。\n");

return 0;

}

printf("\n这不是一颗二叉树排序树。");

printf("\n");

return 0;

}