浙江水利水电学院

实验报告

（2019-2020学年 第1学期）

|  |  |
| --- | --- |
| 课程名称： | 数据结构与算法 |
| 班 级： | 软工S19-1 |
| 学 号： | 2019b31039 |
| 姓 名： | 肖民浪 |
| 实验室（中心）名 称： | 中心机房 |
| 教学单位： | 信息工程与艺术设计学院 |

2019 年 11 月 21 日

实 验 名 称：创建一棵二叉树并进行遍历 指导教师：寿焕君

实 验 日 期：2019.11.21 地 点：现北506

同组学生姓名：无

|  |
| --- |
| 实验内容及要求：  实验内容：   1. 递归建立二叉树 2. 二叉树的非递归先序遍历 3. 求任意结点的层次 4. 交换左右孩子 5. 编写主函数进行测试   实验要求：   1. 掌握二叉树的动态存储结构及表示方法 2. 掌握二叉树的递归建立算法 3. 掌握二叉树的先序、中序和后序遍历递归和非递归算法 4. 运用二叉树的遍历算法求解基于二叉树的有关问题 5. 掌握调试程序的方式，培养编程的耐心 |
| 主要仪器名称及型号：   1. 计算机：Windows7，i5 3Ghz,4GB内存 2. 开发工具：Cfree |
| 实验过程：（可附页）  **任务1：建立二叉树**  （1）打开库文件“bintree.h”，熟悉库中的的相关函数。  （2）在“bintree.h”文件中编写函数bnode \*creat()，根据扩充二叉树的先序序列建立二叉树p。  扩充二叉树：给定一棵二叉树，对树中不足两个孩子的结点（包括叶子结点）都添上附加结点，使每个结点都有两个孩子结点。输入时附加结点用'**#**'表示。如"ABC##D#E##F##"  **任务2：二叉树的遍历（保存文件为：学号-1.c）**  编写算法函数void fstorder1(bnode \*p)实现二叉树p的非递归先序遍历。  **任务3：求结点的层次（保存文件为：学号-2.c）**  假设二叉树采用链式方式存储，编写一个函数int Depth(bnode \*p, char x)，求值为x的结点在二叉树中的层次。  **任务4：交换子女（保存文件为：学号-3.c）**  编写一个函数void change(bnode \*p)，将一棵给定二叉树中所有结点的左、右子女互换。  **任务5：成果提交**  实验报告保存文件为：**学号实验报告-二叉树.docx**，连同3个C源程序文件上传到<ftp://192.168.106.99>上。（不要打包） |
| 实验数据记录或图片：（可附页）   1. 任务1：  |  | | --- | | 函数bnode \*creat()的完整代码：  bnode\* creat(){  bnode \*tree;  datatype c;  scanf("%c",&c);  if(c =='#'){  tree = NULL;  }else{  tree = (bnode\*)malloc(sizeof(bnode));  tree->data = c;  tree->lch = creat();  tree->rch = creat();  }  return tree;  } | | 根据输入序列画出相应的二叉树：（要求不少于5个结点）  输入的字符串：ABD##E##CF###  二叉树： |  1. 任务2：  |  | | --- | | 函数void fstorder1(bnode \*p)的完整代码：  void fstorder1(bnode \*p){  sqstack s;  init(&s);  push(&s,p);  while(!empty(&s))  {  p=pop(&s);  printf("%3c",p->data);  if(p->rch){  push(&s,p->rch);  }  if(p->lch){  push(&s,p->lch);  }  }  } | | 实验结果：（要有输入情况） |  1. 任务3：  |  | | --- | | 函数int Depth(bnode \*p, char x)的完整代码：  int Depth(bnode \*p, char x){  int k=1,flag=1,point\_not\_in\_tree=0;  bnode \*q;  sqstack L1,L2;  init(&L1);  init(&L2);  push(&L1,p);  while(1)  {  if(flag){  while(!empty(&L1))  {  point\_not\_in\_tree = 0;  q=pop(&L1);  if(q->data==x) return k;  if(q->lch!=NULL) push(&L2,q->lch);  if(q->rch!=NULL) push(&L2,q->rch);  }  k++;  flag=0;  point\_not\_in\_tree++;  }else {  while(!empty(&L2))  {  point\_not\_in\_tree = 0;  q=pop(&L2);  if(q->data==x) return k;  if(q->lch!=NULL) push(&L1,q->lch);  if(q->rch!=NULL) push(&L1,q->rch);  }  k++;  flag=1;  point\_not\_in\_tree ++;  }  if(point\_not\_in\_tree == 2){  return 0;  }  }  return 0;  } | | 实验结果：（要有输入情况）  1.当二叉树中存在用户输入的结点时，输出该结点所在的层次    2.当二叉树中不存在用户输入的结点时，提示用户该二叉树中不存在该结点 |  1. 任务4：  |  | | --- | | 函数void change(bnode \*p)的完整代码：  void change(bnode \*p){  if(p){  bnode \*t;  t = p->lch;  p->lch = p->rch;  p->rch =t;  change(p->lch);  change(p->rch);  }  } | | 实验结果：（要有输入情况） | |
| 实验结论：  本实验共有（ 4 ）小题，已完成（ 4 ）小题，完成的题目经运行测试，结果正确。 |
| 教师评语：  该同学在本次实验过程中学习态度认真，具有很强的观察、分析和解决问题的实验能力，能按时完成本次实验的全部任务。经程序运行测试，实验结果正确。  实验完成情况：优；代码规范：优-；实验报告表达能力：好。  成绩：优 批阅教师：寿焕君  2019年11月21日 |