数据结构与算法

实验报告

实验题目：实验三 题目一 基础实验

姓 名：刘帅

学 号：2020212267

日 期：2021年5月30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5月30日

程序引用说明

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 引用项 | 来源 | 相同代码行数 |
| 1 | 栈的构造及功能 | 课本 | 10 |
| 小计 | | | 10 |

总代码行数134; 引用占比\_\_\_\_\_\_8%\_\_\_\_\_\_\_\_\_\_

1、实验简介

【实验内容的简要说明，具体说明实验完成的功能和性能要求】

根据二叉树的抽象数据类型的定义，使用二叉链表实现一个二叉树。从而实现

1、二叉树的建立2、前序遍历二叉树3、中序遍历二叉树4、后序遍历二叉树

5、按层序遍历二叉树6、求二叉树的深度7、求指定结点到根的路径

8、二叉树的销毁9、寻找指定结点的路径 功能

2、程序框架

1. 与构造相关的函数：

void create(node<T>\*& r, T\*data, int i, int n);//递归的建立二叉树

2．与操作有关函数：

void preorder(node<T>\* r);//前序遍历

void inorder(node<T>\* r);//中序遍历

void postorder(node<T>\* r);//后序遍历

void leverorder ( node<T>\* r,int n);//层序遍历

void getpath(int\*arr,T a, T b,T n);//找指定节点的路径

3、与删除相关函数

void release(node<T>\* r);//递归的删除节点

~bitree();//析构函数

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）】

文本

描述已自动生成3.1结点node结构体的定义

文本

描述已自动生成3.2 void bitree<T>::create(node<T>\*& r, T\*data, int i, int n) //递归的建立二叉树

3.3

void bitree<T>::preorder(node<T>\* r)//前序遍历二叉树

void bitree<T>::preorder(node<T>\* r)//中序遍历二叉树

void bitree<T>::postorder(node<T>\* r)//后序遍历二叉树

文本

描述已自动生成

文本

描述已自动生成3.4 void bitree<T>::leverorder(node<T>\* r,int n)//层序遍历

屏幕上有字

描述已自动生成3.5 void bitree<T>::getpath(int\* arr, T a, T b, T n) {//利用栈的思想，后入栈的先出

文本

描述已自动生成3.6运行结果

4、不足

实现的扩展功能较少

5、心得体会

1、若数据量非常大，如何使得构造二叉树时栈不溢出？ 答：当数据量很大时可以采用链式存储的方式，通过指针来建立动态的结构体，提高储存空间。

2、若二叉树的每一个结点具有数值，如何搜索二叉树，找到指定值的叶子结点？

可采用层序遍历的方法，与所查找的数值进行匹配，定义查找次数time与层数layer；若在第time次找到该数值，返回layer=abs(log2 time)为层数，time-2^layer为在layer层的个数.

3、若已知叶子结点的指针，如何输出从根到该叶子的路径？

见上述算法getpath（）；

数据结构与算法

实验报告

实验题目：实验三 题目2 二叉树实现哈夫曼编解码器

姓 名：刘帅

学 号：2020212267

日 期：2021年5月30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4月30日

程序引用说明

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 引用项 | 来源 | 相同代码行数 |
| 1 | 栈的构造及功能 | 课本 | 8 |
| 小计 | | | 8 |

小计

总代码行数141; 引用占比\_\_\_\_\_\_5%\_\_\_\_\_\_\_\_\_\_

1、实验简介

（1）利用二叉树结构实现哈夫曼编/解码器。实现了对字符串每个字符的频度进行统计并建立哈夫曼树。

（2）建立编码表(CreateTable)：利用已经建好的哈夫曼树进行编码，并将每个字符的编码输出。

（3）编码(Encoding)：根据编码表对输入的字符串进行编码，并将编码后的字符串输出。

（4）译码(Decoding)：利用已经建好的哈夫曼树对编码后的字符串进行译码，并输出译码结果。

2、程序框架

1. 与构造及操作有关函数：

（1）int getnum(string s)//统计字符串字母出现频度，并返回字符种类的个数。

（2）bool operator < (node x, node y)//重载运算符，返回x的权值大于y的权值的真值。

（3）void huffman::create(string s ,int n)//建立哈夫曼树

（4）void huffman::code(int i,string a)//为哈夫曼树的边赋值

（5）void huffman::create(string s ,int n)//创建编码表

（6）string huffman::encode(string str, string d) //编码

（7）string huffman::decode(string str, string d) //解码

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）】

文本

描述已自动生成3.1 int getnum(string s)//统计字符串字母出现频度，并返回字符种类的个数。

3.2 void huffman::create(string s ,int n)//创建编码表文本

描述已自动生成

首先，需要建立结构体数组getcharr，用来存储字符及其出现次数图片包含 图形用户界面

描述已自动生成

文本

描述已自动生成而后，通过getnum函数，用num表示getnum（s）的返回值，即出现字符种类的个数。

利用cmp函数，将结构体数组按字符出现频度的大小（由小到大）排序。

建立node结构体，表示哈夫曼树的结点信息，文本

描述已自动生成并将getcharr的权值、出现次数内容导入到node实例化对象tree中，利用非递归及优先队列，建立哈夫曼树。

文本

描述已自动生成3.3 void huffman::code(int i,string a)//递归的哈夫曼树的边赋值

3.4 void huffman::code(int i,string a)//递归的为哈夫曼树的边赋值文本

描述已自动生成

文本

描述已自动生成3.5 string huffman::encode(string str, string d) //编码

文本

描述已自动生成3.6 string huffman::decode(string str, string d) //解码

文本

中度可信度描述已自动生成3.7可视化结果

显示哈夫曼树的结点信息、编码表、编码及解码结果。

4、不足

未完成二进制编码选作任务

5、心得体会

本实验为较为综合的问题，在过程中学习了优先队列、operator迭代器等STL知识，并对递归有了更加深刻的认识。