**数据结构**

2022

**实 验 报 告**

|  |  |
| --- | --- |
| 实验项目名称： | 数据结构实验六 |
| 班级： | 21级9班 |
| 学号 | 2021302121185 |
| 姓名： | 陈聪睿 |
| 指导教师： | 董红斌 |
| 实验时间： | 2022.4.28 |

**实验一： 实现二叉树各种基本运算的算法**

**一、实验要求**



**二、实验环境**

**Visual studio 2019/ Windows平台**

**三、实验步骤及思路**

（1）题目分析与主要步骤

根据题目要求可知，需要构建一个二叉链存储结构以及若干实现方法。

二叉链基本存储结构可以借鉴树的孩子链存储结构，即用两个指针分别指向本结点的左孩子与右孩子。

根据题目分析可知，要满足题目条件，需要有创建二叉链CreateBTNode、释放二叉链DestoryBTNode、寻找目的结点FindNode、查询该结点的左孩子LchildNode、、查询该结点的右孩子RchildNode、得到该树的高度（深度）BTNodeDepth、打印该树BispBTNode等实现方法。

对于其中的一些实现方法，使用递归的方法实现，可以简化代码的复杂程度，使之易于其他程序员理解与维护。

（2）实验具体步骤截图

测试数据：题目已给出，结果如下图。

文本

描述已自动生成

**四、实验结果及分析**

可以看出已经按照需要打印出了正确结果。

**五、总结**

通过本实验，可以看出二叉链有如下特点：

1. 二叉链比较节省空间，其占用的存储空间与树形没有关系，只与树中结点个数有关。
2. 在二叉链中，找一个结点的孩子很容易，但找其双亲不方便。

同时，我们也可以感受到，树一类的数据结构类型，十分适合于使用递归方法来实现其基本算法以及其他特定算法，这是由其天然带有的递归关系而决定的。

**实验二： 求二叉树中的结点个数、叶子结点个数、某结点层次和二叉树宽度**

**一、实验要求**

 

**二、实验环境**

**Visual studio 2019/ Windows平台**

**三、实验步骤及思路**

（1）题目分析与主要步骤

根据题目要求，我们需要创建输出二叉树结点个数Nodes、输出二叉树叶子结点个数LeftNodes、求二叉树中指定值结点层次Level，层次遍历求二叉树宽度Width四个实现方法。

（1）Nodes函数使用遍历方法，其递归模型如下：

当b为空结点（b==NULL），说明包括其本身以下无结点，返回0；如果b是叶子结点（b->lchild == NULL && b->rchild == NULL），说明包括其本身以下只含有一个结点，返回1。

当b为非叶子结点时，说明包括其本身以下有2即以及上的结点，我们需要统计其左孩子的结点个数（n1）与右孩子的结点个数（n2），在算上当前结点的同时，加在一起作为当前结点含有的结点个数（n1+n2+1）。

（2）LeftNodes函数与Nodes函数类似，只不过其统计的时候，不需要将非叶子结点算在一起计算，即（n1+n2）。

（3）Level函数使用递归，其递归模型如下：

当b为空结点（b==NULL），说明当前层次为，返回零；当d->data==x时，证明找到了目的结点，直接打印其层次h，并返回以示结束。

当该结点不为所需结点或空结点时，我们需要先查看左孩子是不是目的结点，否则查询右孩子是不是目的结点。

（4）对于Width函数的实现，为了满足题目“层次遍历”的要求，需要使用队列存储每个结点。

在每次循环中，我们需要出队一个结点，并根据左右孩子是否存在记下当前的层次（以便之后扫描队列时，可以分清该层次有多少结点），并将孩子结点入队，直到所有结点均被考虑。

在之后，我们要根据层次编号，来统计该层次下结点的个数，将最大的数值返回，即为最终结果。

（2）实验具体步骤截图

测试数据：题目已给出，结果如下图

文本

描述已自动生成

**四、实验结果及分析**

可以看出已经按照需要打印出了正确结果。

**五、总结**

因为树具有天然的递归性，所以在之后求解树的复杂问题时，我们不妨使用递归，将困难的大问题求解分解成一个个容易处理小问题求解，这样子更方便我们理解和编写代码。

**实验三： 简单算术表达式二叉树的构建和求值**

**一、实验要求**



**二、实验环境**

**Visual studio 2019/ Windows平台**

**三、实验步骤及思路**

（1）题目分析与主要步骤

根据实验要求，我们需要编写创建算数二叉树CRTree函数以及计算二叉树对应表达式compute函数这两个实现方法来满足题目要求。

1.CRTree函数采用递归算法，我们使用i，j分别表示原字符串的头部与尾部。

如果i==j，说明其只有一个字符，直接创建一个只有根结点的二叉树并返回。

如果i！=j，则需要遍历该字符串，根据运算法则创建树：

先找’+’、’-’号，将最后的’+’、’-’号作为根结点

若没有’+’、’-’号，则寻找’\*’、’/’号，以最后一个运算符为根结点。

这样就将子串分为了两部分，然后重复上述过程，直到所有运算符都被存入树中，并将其余数值依次存放在运算符的左右孩子处，作为叶子结点，即为结束。

2.compute函数同样是一个递归函数，用于根据树存放的结构来进行计算，其过程如下：

如果为空结点，则返回0；如果为叶子结点，则返回其数值（因为ElemType为char，需要data-‘0’获取其正在数值）。

如果为非叶子结点，则先将其左孩子与右孩子的数值算出，即为v1，v2，然后将其按照其存储的的运算符进行相应的运算，并返回其结构（除法要注意除数不为0），直到所有的结点都被遍历为止，最终返回值即为计算结果。

（2）测试数据：“1+2\*3-4/5”“2/0+5”“5”具体步骤截图

测试数据：，结果如下图

图片包含 文本

描述已自动生成

图形用户界面, 文本, 应用程序

描述已自动生成

文本

描述已自动生成

**四、实验结果及分析**

可以看出已经按照需要打印出了正确结果。

**五、总结**

通过将中序表达式改写为二叉树，我们很容易就得到了前序表达式，这无疑比以前使用栈编写简单了许多。