

基于 C# 的二叉排序树图形显示系统研究与实现

杨政安

(兰州职业技术学院,甘肃 兰州 730070)

摘要:介绍了二叉树在数据结构中的应用,并结合 C#语言实现了二叉树的可视化功能。对二叉树的构建和输出实现进行了说明,同时对数据结构的教学方法进行了讨论。

关键词:计算机软件;二叉树;图形显示;遍历

中图分类号:TP29

1 背景介绍

二叉树是树形结构的一个重要类型。许多实际问题抽象出来的数据结构往往是二叉树的形式,即使是一般的树也能简单地转换为二叉树,而且二叉树的存储结构及其算法都较为简单,因此二叉树显得特别重要。

二叉排序树(Binary Sort Tree)又称二叉查找(搜索)树(Binary Search Tree)。其定义为:二叉排序树或者是空树,或者是满足如下性质的二叉树:

- 1)若它的左子树非空,则左子树上所有结点的值均小于根结点的值;
- 2)若它的右子树非空,则右子树上所有结点的值均大于根结点的值;
- 3)左、右子树本身又各是一棵二叉排序树。

上述性质简称二叉排序树性质(BST性质),故二叉排序树实际上是满足BST性质的二叉树。

二叉排序树在实际应用中,经常用来实现提高数据的查寻效率,但由于概念理解上的困难,在教学过程中,学生往往不能掌握二叉树的基本操作,不能在脑海中形成对应的模型。所以本文试图从概念入手,用可视化的方法来对数据结构中二叉树的教学进行改进。

2 实现工具

系统采用 C#编程语言进行开发,C#是微软公司发布的一种面向对象的、运行于 .NET Framework 之上的高级程序设计语言,通过 C#可以让开发人员快速地建立大范围的基于 MS 网络平台的应用,并且提供大量的开发工具和服务帮助开发人员开发基于计算和通信的各种应用。基于这个特点,我们采用 C#来开发二叉排序树图形显示系统。

3 具体设计

3.1 界面设计

由于只是一个图形显示系统,所以就没有繁琐的窗口和页面,只设计了一个主界面,在主界面上完成二叉排序树的显示,实现二叉树各种数据操作。

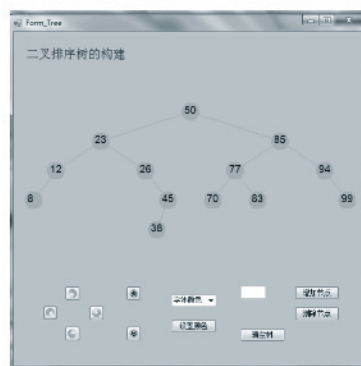


图1 界面设计

3.2 功能设计

本系统在功能上,除了可以进行二叉排序树基本操作,初始化、节点的增加、删除、清空所有节点外,还对显示功能进行了强化,增加了颜色设置可以设置界面背景颜色、节点的颜色和节点字体的颜色,还能进行二叉树节点间宽度的增大和缩小。

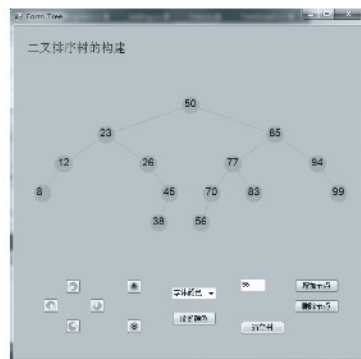


图2 功能设计

4 实现

实现二叉树的显示,主要要进行这样几个操作,首先是初始化二叉树,对初始的输入序列依次进行增加节点操作,在增加节点时先要找到节点要插入的位置,然后在窗口上显示出这个结点,直到所有输入序列都插入到二叉树中为止,这些操作都是递归进行的。

4.1 插入位置查找

```
private bool Search(TreeNode node,int data)
{
    if (node.data == data)
    {
        found = true;
    }
    else
    {
        if (node.data > data) ///条件成立则查左子
        树,否则查右子树。
        {
            if (node.LeftNode != null)
            {
                Search(node.LeftNode,data);
            }
            else
            {
                if (node.RightNode != null)
                {
                    Search(node.RightNode,data);
                }
            }
        }
        return found;
    }
}
```

4.2 插入操作

```
private void Insert(TreeNode node,int data)
{
    if (node.data > data)
    {
        if (node.LeftNode == null)
        {
            TreeNode temp = new TreeNode();
            temp.data = data;
```

```
node.LeftNode = temp;
        }
        else
        {
            Insert(node.LeftNode,data);
        }
    }
    else
    {
        if (node.RightNode == null)
        {
            TreeNode temp = new TreeNode();
            temp.data = data;
            node.RightNode = temp;
        }
        else
        {
            Insert(node.RightNode,data);
        }
    }
}
```

4.3 显示模块

```
private void Draw (TreeNode parent, Point cur,
Point last,int deep)
{
    if (parent == null)
    {
        return;
    }
    /* 画圆圈 */
    Size size = new Size (nodeRadius * 2, nodeRadi-
us * 2);
    Point center = new Point();
    center.X = cur.X - nodeRadius;
    center.Y = cur.Y - nodeRadius;
    Rectangle rec = new Rectangle(center,size);
    myGraphics.FillEllipse(cirBrush,rec);
    /* 画线 */
    myGraphics.DrawLine(myPen,cur,last);
    /* 绘制左右子树 */
    Point left = new Point();
    int len = (int) Math.Pow(2.0,(double)deep);
    left.X = cur.X - (nodeRadius + distance-
Width) * len;
    (下转 87 第页)
```

(上接第 28 页)

```
left. Y = cur. Y + distanceHeight;
Draw( parent. LeftNode, left, cur, deep - 1);
Point right = new Point();
right. X = cur. X + (nodeRadius + distance-
Width) * len;
right. Y = cur. Y + distanceHeight;
Draw( parent. RightNode, right, cur, deep - 1);
/* 写字 */
center. X = cur. X - nodeRadius * 9 / 10;
center. Y = cur. Y - nodeRadius * 9 / 10;
myGraphics. DrawString ( parent. data. ToString
(), myFont, fontBrush, center);
|
```

5 结论

在对计算机应用专业和软件技术专业的学生进行数据结构课程的教学中,通过使用二叉排序树图形显示系统,发现学生对学习数据结构这门课更加

有兴趣,学生在学习树这一部分内容的时候,对树的概念和基本操作都有了深刻的理解,对掌握二叉树这一数据结构中的重要概念也起到了很好的帮助作用。

今后将在这个系统的基础上,完善系统功能,增加诸如平衡的二叉排序树的显示实现、遍历树的过程显示等功能,并对数据结构中的各种经典算法比如图中最小生成树、最短路径、排序算法等,进行显示系统的构建,使这个系统更加丰富起来,真正成为学习数据结构课程的辅助工具。

参考文献:

- [1] 吴福英,谭罗生,王明文. 顺序存储的满二叉树中序遍历的非递归算法[J]. 江西师范大学学报(自然科学版),2003,27(4).
- [2] 范时平. 基于满二叉树的原地快速排序[J]. 重庆邮电学院学报(自然科学版),2006,18(6).
- [3] 周志华. 二叉树的绘制算法计算机应用研究[J].
- [4] 储美芳. 浅谈《数据结构》课程教改[J]. 成才之路,2010(29).