

北京邮电大学

实验报告



实验名称: 使用先序递归建立二叉树

班级 : 2019211309 姓名 : 陈悦 学号 : 2019211413 分工 : 文档

班级 : 2019211309 姓名 : 马晓亮 学号 : 2019211400 分工 : 代码

2020 年 11 月 29 日

一 需求分析

本程序将利用先序遍历所得序列，建立一棵可以唯一确定的二叉树。使用 # 标志一个输入的结束。支持输入多个数据。

程序具有判定和输出所建成的二叉树的功能。程序会判定输入是否合法，比如 a*** 可以判定为一个不合法的输入。

此时程序会输出 Input Error

当二叉树建成时，程序会使用广义表的方式输出二叉树。例如样例中的二叉树可以输出为 a(b(c, d), e)

二 概要设计

首先我们要定义一个使用二叉链表表示的结构体。

1. TreeNode* lch 储存左子树
2. TreeNode* rch 储存右子树
3. char data 储存数据

再定义一个解决问题的类

1. printTree() 输出二叉树的广义表行事
2. inputTree() 输入一个字符串并将该字符串转化为二叉树
3. buildTree() 传入指针，递归建树

三 详细设计

Algorithm 1 inputTree

```
1: function INPUTTREE(void)
2:   读入字符串, 并保存在 str 中
3:   初始化根节点为 root
4:   if buildTree(root, str, 1) == str 的长度 then
5:     printTree(root)
6:   else
7:     打印 Input Error
8:   end if
9: end function
```

Algorithm 2 buildTree

```
1: function BUILDTREE(root, str, n)
2:   if str[n] 是 * then
3:     return n
4:   else
5:     root 结点的数据设为 str[n]
6:     初始化 lch
7:     n = buildTree(lch, str, n + 1)
8:     初始化 rch
9:     n = buildTree(rch, str, n + 1)
10:    return n
11:  end if
12: end function
```

Algorithm 3 printTree

```
1: function PRINTTREE(root)
2:   if root 为空 then
3:     return
4:   else
5:     输出 root 数据
6:     if root 存在子树 then
7:       输出 (
8:         printTree(lch)
9:         输出,
10:        printTree(rch)
11:        输出)
12:     end if
13:   end if
14: end function
```

四 调试分析报告

buildTree 函数调用的次数 n 次，所以我们认为代码的复杂度为 $O(n)$

五 用户使用说明

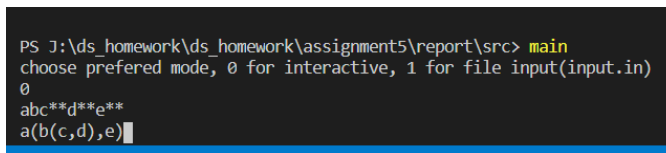
用户可以使用 IDE 或者手动编译源代码 stack.cpp，获得可执行文件。

笔者使用的 gcc 版本为 8.1.0

用户执行程序时，可以输入 0 表示交互式输入，1 表示文件输入，一行表示一次输入文件输入时，输入内容要存储在 in.txt 中，输出到 out.txt 中。

六 测试结果

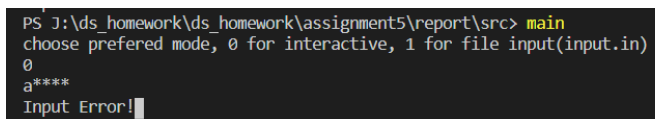
输入样例 abc**d**e**，输出为



```
PS J:\ds_homework\ds_homework\assignment5\report\src> main
choose preferred mode, 0 for interactive, 1 for file input(input.in)
0
abc**d**e**
a(b(c,d),e)
```

图 1: 样例

同时作为非法样例，还有 a****



```
PS J:\ds_homework\ds_homework\assignment5\report\src> main
choose prefered mode, 0 for interactive, 1 for file input(input.in)
0
a****
Input Error!
```

图 2: 非法样例