14 【2017】二叉树转换为中缀表达式

24计算机考研成员一战成硕！

💡 题目描述：

请设计一个算法，将给定的表达式树（二叉树）转换为等价的中缀表达式（通过括号反映操作符的计算次序）并输出。例如，当下列两颗表达式树作为算法的输入时：

输出的等价中缀表达式分别为(a+b)*(c*(-d))和(a*b)+(-(c-d))。

要求：

（1）给出算法的基本设计思想。

（2）根据设计思想，采用C或者C++语言描述算法，关键之处给出注释

倒计时

1、知识点及难度

✍

题解人：多动症男孩

难度：中等

知识点：

1. 中缀表达式使用括号来反映操作符的计算次序，确保了正确的运算顺序。
2. 注意在C和C++中，字符串的处理方式略有不同，需要根据语言选择合适的方式处理字符串。

- 时间复杂度：O(n)
- 空间复杂度：O(n)

2、算法题

思路

1. 从根节点开始递归遍历树。
2. 如果当前节点是叶子节点，输出该节点的值。
3. 如果当前节点是操作符节点（加法、减法、乘法、除法等），则输出左括号，然后递归处理左子树和右子树。
4. 最后，输出操作符节点的值，再输出右括号。

> 基本实现-C++

```cpp
#include <iostream>
#include <string>
using namespace std;

// 定义二叉树结构体
typedef struct BTNode {
    string val;
    BTNode* left;
    BTNode* right;
}BTNode,*BTree;

// 递归将表达式转化为中缀表达式
string expressionToInfix(BTree root) {
    if(!root) return "";

    // 如果是叶子节点，直接返回值
    if(!root->left && !root->right) return root->val;

    // 递归处理左右子树
    string leftExpr = expressionToInfix(root->left);
    string rightExpr = expressionToInfix(root->right);

    // 返回中缀表达式
    return "(" + leftExpr + root->val + rightExpr + ")";
}
```

> 基本实现C

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

// 定义二叉树节点结构
```

```c
struct TreeNode {
    char val[100];
    struct TreeNode* left;
    struct TreeNode* right;
};

// 递归将表达式树转换为中缀表达式
void expressionToInfix(struct TreeNode* root) {
    if (root == NULL) {
        return;
    }
    // 如果是叶子节点，直接输出值
    if (root->left == NULL && root->right == NULL) {
        printf("%s", root->val);
    } else {
        // 否则，输出左括号，递归处理左子树和右子树，输出操作符，输出右括号
        printf("(");
        expressionToInfix(root->left);
        printf("%s", root->val);
        expressionToInfix(root->right);
        printf(")");
    }
}
```

3、总结

总结栏

蓝蓝B站首页：蓝蓝希望你上岸呀B站首页

蓝蓝公众号：算法训练营9分计划

蓝蓝知识星球介绍：关于知识星球的权益

如何在星球打卡记录：

- 计算机考研数据结构算法专项day[1/60]：
- 学习内容：最好能发出自己写的图片
- 遇到的问题：如果无就不用写了
- 小结：这部分一周写一次即可。