
武汉纺织大学计算机与人工智能学院

专业综合训练 I 课程报告

训练选题： 表达式及其应用

专业班级： 计算机类 12101 班

学 号： 2105251027

学生姓名： 陈宇凡

指导教师： 专业综合训练 I 课程组

2022 年 7 月 5 日

目录

1、项目介绍	1
1.1 训练内容	1
1.2 训练要求	1
2、项目分析	1
2.1 功能列表	1
2.2 实施方案	2
3、项目设计与实现	2
3.1 数据模型	2
3.2 模块设计	2
3.3 算法设计	3
3.4 函数实现	4
3.5 功能测试	6
4.1 团队合作	7
4.2 项目成果	8
4.3 项目展望	8
4.4 收获和感悟	8
5 参考文献	10
6.致谢	10

1、项目介绍

1.1 训练内容

本专题为表达式求值及其应用，主要有三个任务。

1. 交互式求表达式值：支持更丰富运算符，支持变量和常量操作数（标识符），以交互的方式求值；支持 C 语言中大多数运算符，符合运算符的语法与语义，存储会话期变量的取值。

2. 真值表及其应用：根据实际逻辑问题设计命题公式，通过输入命题公式到真值表程序的输出，完成真值表的应用。

3. 二元树与表达式：通过中缀表达式建立对应二元树（二叉链表），输出波兰和逆波兰表达式，仅带必要括号的中缀表达式，通过对二叉树进行递归操作，求表达式的值。

1.2 训练要求

1. 预备任务：学习使用 C++STL，在实训前三节课听三位老师所讲的内容并理解老师分发的关于简单的中缀表达式的求值的示例代码，对计算机计算表达式的几种算法进行理解，并熟悉数据结构中栈的基本操作和实际应用。

2. 任务前期：应该多理解理论知识，在机房实训过程中，需要了解表达式可以以一颗二叉树无歧义表示，通过对树的不同顺序遍历可以获得前中和后缀表达式，通过对树的递归操作可以求得表达式的值其次，结合离散数学课程中学习的基础知识了解连结词的优先级、结合律。

3. 任务中期：应多增加动手能力，实际编程能力，在理解上述理论知识后，还需要通过查阅资料、询问他人等方式学习 C++部分 STL 的基本使用方式，掌握数据结构中某些代码的实现，加之要具有一定规模程序的编写和调试解决问题的能力、与组内其他成员进行沟通协商的合作能力。

4. 任务末期：在该报告的编写阶段，还要能撰写软件开发过程中的关键环节，分析方案的合理性与选择性，并在自己的理解基础上较规范且完整的完成报告任务。

2、项目分析

2.1 功能列表

实现了专题一选题的扩展任务三：二元树与表达式。输入中缀算术表达式，生成二元

树，输出表达式的波兰和逆波兰表示，输出表达式的中缀表达式，适当的带上括号，并对二叉树进行递归操作，求表达式的值。

对于上述任务，我已经基本实现，能够输入中缀算数表达式并生产二元树，但是运算符的支持还不够完善，单目元素符暂不支持。在输入的表达式正确的情况下，能够正确的建立所对应的二叉链表形式的二元树并输出前、中、后序序列，中序遍历时当其还有子表达式时才输出括号，一定程度上降低了括号的冗余率，但部分情况还是会存在少量冗余括号。

2.2 实施方案

对于本人实现的任务，选择的实施方案为先将中缀表达式转化为后缀表达式，然后再通过栈将后缀表达式生成二元树。其次，在通过类内的方法完成树的三序遍历，在中序遍历时，只有节点非叶子才会生成括号，一定程度上避免了冗余括号的出现。对于求值，则使用了递归操作的方法，借助库函数 `atof`，来实现递归求值。为了类的使用简单，将部分方法封装在类内，并提供更加方便使用的接口。

对于实现方案，在前期的准备工作中了解到有直接由中缀表达式生成二元树以及先转为后缀表达式，再生成二元树。为了能利用上任务一的所写的部分内容，提高代码的复用性，故选择了第二种方式，并且将该任务与任务一集成进了一个程序中。

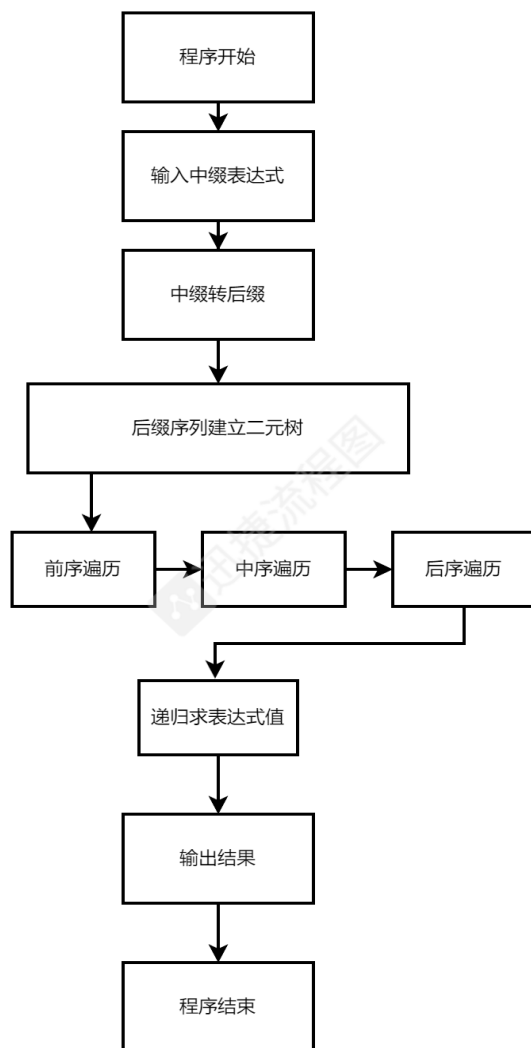
3、项目设计与实现

3.1 数据模型

主要的数据结构设计为对存储表达式的二元树的设计与实现。设计了 `BiNode` 和 `BiTree` 两个类。其中 `BiNode` 为二叉树的单个节点，包含了操作数字符串 `n`、运算符 `op` 两个成员变量，并用 `status` 区分当前节点是运算符还是操作数。其中宏 `OP`（值为 0）代表运算符，`NUM`（值为 1）代表操作数。此外，还有该节点的左右孩子两个指针。其构造函数有两个构造函数，创建新节点时只需将运算符或操作数直接传入即可。而 `BiTree` 即为二叉表达式树类。封装了根节点的指针，并对外提供了建立表达式二元树，获得二元树的前中后遍历序列的以及通过递归操作对成员函数。为了避免内存泄漏，还设计了删除二叉树的方法，在对象发生析构时会删除该树的所有节点。

3.2 模块设计

如下：



3.3 算法设计

1. 中缀表达式转后缀表达式算法：

遍历经过表达式替换后的中缀表达式串 `infixExp`, 并创建后缀表达式 `suffixExp`。遍历时, 若该字符是空格则跳过; 若是操作数则继续向后遍历得到整个数字随后直接输出到后缀表达式, 取得数字后更新遍历指针到数字的最后一位, 特别地, 若识别到负号则也直接输出到表达式中; 若是运算符, 则当栈顶运算符的优先级小于当前运算符时可以继续压栈, 否则则需要先弹栈输出直至栈顶运算符优先级小于当前运算符时再将当前运算符压栈。若遇到右括号则无条件压栈, 遇到左括号则无条件弹栈输出并输出运算符直至右括号出栈 (不输出到后缀表达式), 且若栈顶为右括号, 则可以无条件压栈。遍历结束时将栈内所有运算符全部输出, 并为字符串加上休止符 ‘\0’。另外, 为了保证后缀表达式的标准性, 便于后续处理, 会在输出一个数字或是一个运算符时输出一个空格表示分隔。

2. 后缀表达式建立二元树算法:

后缀表达式的特点是：一定以两个操作数开始，且以操作符结尾。如同后缀表达式求值一样，逐次读取后缀表达式的每一个符号。如果符号是操作数，那么就建立一个单节点树，并将一个指向它的指针推入栈中；则从栈中弹出两颗树 T1 和 T2，并形成一颗以操作符为根的树，其中 T1 为右儿子，T2 为左孩子，然后将新的树压入栈中，继续上述过程即可。

3. 树的遍历输出前、中、后缀表达式算法:

先序遍历：首先访问 T 的根，然后递归的访问子树的根。如果这棵树是有序的，则根据孩子的顺序遍历子树。中序遍历：通过递归遍历左右子树区访问一个位置。对于一个位置 p，p 将其左子树之后及其右子树之前被中序访问。后序遍历：看作相反的先序遍历，优先遍历子树，再访问根。在输入的表达式正确的情况下，能够正确的建立所对应的二叉链表形式的二元树并输出前、中、后序序列，中序遍历时当其还有子表达式时才输出括号。

3.4 函数实现

按照后序遍历序列构造表达式树（如图 3.4-1）：

```
void BiTree::buildTree(char* suffixExp)
{
    stack<BiNode*> s;
    int len = strlen(suffixExp);
    for (int i = 0; i < len; ++i)
    {
        if (suffixExp[i] == ' ')
        {
            continue;
        }
        if (suffixExp[i] == '-' && isNumber(suffixExp[i + 1]) || isNumber(suffixExp[i])) //是数字
        {
            string num;
            num.push_back(suffixExp[i]);
            int j = i + 1;
            while (suffixExp[j] != ' ')
            {
                num.push_back(suffixExp[j]);
                j++;
            }
            i = j; //i指针无需回退，因为数字后面必为空格
            auto newNode = new BiNode(num);
            s.push(newNode);
        }
        else //是运算符
        {
            auto newNode = new BiNode(suffixExp[i]);
            newNode->rchild = s.top();
            s.pop();
            newNode->lchild = s.top();
            s.pop();
            s.push(newNode);
        }
    }
    this->root = s.top();
}
```

(图 3.4-1)

```

LDouble BiTree::calcTree(void)
{
    if (!this->root)
    {
        return 0;
    }
    else
    {
        return m_calcTree(this->root);
    }
}

LDouble BiTree::m_calcTree(BiNode* T)
{
    if (!T) return 0;
    if (!T->lchild && !T->rchild)
    {
        LDouble v = (LDouble)atof(T->n.c_str());
        return v;
    }
    else
    {
        LDouble lv = 0, rv = 0;
        lv = m_calcTree(T->lchild);
        rv = m_calcTree(T->rchild);
        return calc(T->op, rv, lv);
    }
}

```

(图 3.4-2)

遍历求值（如图 3.4-3）：

```

LDouble BiTree::m_calcTree(BiNode* T)
{
    if (!T) return 0;
    if (!T->lchild && !T->rchild)
    {
        LDouble v = (LDouble)atof(T->n.c_str());
        return v;
    }
    else
    {
        LDouble lv = 0, rv = 0;
        lv = m_calcTree(T->lchild);
        rv = m_calcTree(T->rchild);
        return calc(T->op, rv, lv);
    }
}

```

(图 3.4-3)

3.5 功能测试

带有括号、小数、带负号的四则运算的表达式（如图 3.5-1、图 3.5-2、图 3.5-3）：

```
Input command >>> tree ((20+30)*5)%50+(60-50.4)/2.14
先序遍历结果为: + % * + 20 30 5 50 / - 60 50.4 2.14
中序遍历结果为: ( ( (20+30)*5)%50) + ( (60-50.4)/2.14)
后序遍历结果为: 20 30 + 5 * 50 % 60 50.4 - 2.14 / +
遍历求值结果为 4.49
```

(图 3.5-1)

```
Input command >>> tree (3.14+(-9.87))*5+60
先序遍历结果为: + * + 3.14 -9.87 5 60
中序遍历结果为: ( (3.14+-9.87)*5)+60
后序遍历结果为: 3.14 -9.87 + 5 * 60 +
遍历求值结果为 26.35
```

(图 3.5-2)

```
Input command >>> tree ((-60.3)+50.69)*90+(-60.23)/8.4
先序遍历结果为: + * + -60.3 50.69 90 / -60.23 8.4
中序遍历结果为: ( (-60.3+50.69)*90) + (-60.23/8.4)
后序遍历结果为: -60.3 50.69 + 90 * -60.23 8.4 / +
遍历求值结果为 -872.07
```

(图 3.5-3)

输出简单中序遍历表达式的三种遍历序列（如图 3.5-4、图 3.5-5）：

```
Input command >>> tree 1+2+3
先序遍历结果为: + + 1 2 3
中序遍历结果为: (1+2)+3
后序遍历结果为: 1 2 + 3 +
遍历求值结果为 6.00
```

(图 3.5-4)

```
Input command >>> tree 2*9-9.2
先序遍历结果为: - * 2 9 9.2
中序遍历结果为: (2*9)-9.2
后序遍历结果为: 2 9 * 9.2 -
遍历求值结果为 8.80
```

(图 3.5-5)

含关系运算符、逻辑运算符的表达式（如图 3.5-6）：


```

Input command >>>tree (3|4)&&(5&&6)==(4<1)
先序遍历结果为: & | 3 4 = & 5 6 < 4 1
中序遍历结果为: (3|4)& ( (5&6)= (4<1))
后序遍历结果为: 3 4 | 5 6 & 4 1 < = &

遍历求值结果为 0.00

```

(图 3.5-6)

4、总结

4.1 团队合作

1. 在本次实训中，小组成员有刘家麒、杨成希与陈宇凡。在实训过程中，我们有进行合理明确且有目的的分工，在分工后经常通过 QQ 聊天和语音通话交流各自对任务的理解以及自己的任务进程和现阶段中遇到的技术困难，交流沟通时三个人相互协作、相互讨论交流、相互分享任务中可以用到的的知识和技术、虚心听取其他组员的意见，最后将任务中出现的错误进行合理的改正。

2. 具体的组员分工为：刘家麒负责拓展任务一（较复杂表达式的交互式求值）；杨成希负责拓展任务二（求真值表及应用）；陈宇凡负责拓展任务三（二元树与表达式）。由于我组认为任务一和三的联系紧密，所以我组将拓展任务一和三合并到一个项目中，通过刘家麒和陈宇凡定期同步代码文件的方式完成代码的编写，拓展任务二则单独作为另一个项目，由杨成希负责最终代码的编写。

3. 具体地说，本人在此次综合训练中具体任务主要是：完成拓展训练三的所有任务，通过中缀表达式建立对应二元树（二叉链表），输出波兰和逆波兰表达式，仅带必要括号的中缀表达式，通过对二叉树进行递归操作，求表达式的值。其次，因为我组将任务一与任务三做成了一个整个项目的方案，故主函数和一些需要共用的辅助性函数是由我和刘家麒同学共同完成的。本人在团队中积极参与，认真理解知识点，与组员之间交流紧密，虚心请教，并及时更正代码中出现的一些问题，在任务交流中勇于发表自己的看法，对组员的代码提出自己的疑问，请教代码中不懂不理解的地方，希望将这次合作做到最好，将自己的价值发挥到最大，力争将本次实训完成的更合理更完美更规范。

4. 评价：我组三名成员每一个人都积极配合，合理分配任务，各尽其职，在自己能力范围之内，尽全力的帮助他人完成，互相找寻对方代码中可以更加完善的地方。改进之处：网络交流还是会出现一定的交流隔阂，希望面对面的交流可能会更好一些。

4.2 项目成果

在本次实训中，本人主要负责第三个任务，即通过给定的表达式生成该表达式的二元树，主要的贡献为设计了二元树的数据结构以及一些必要的算法，包括树的建立、遍历、递归求值、删除等都是由本人实现的。此外，主程序中关于生成二叉树的部分代码也是由我编写。

新的认识：我本来对于二叉树的代码运用不是非常的熟练，但是通过查找书籍、上网查找的方法，学会了运用二叉树以及相关的代码，更清楚的了解到了递归的原理以及递归调用时的执行顺序，对函数调用栈的理解更加的深入。

4.3 项目展望

该拓展任务还有虽已经完成了基本的要求，但还是有些许不太完美的地方。主要有在对运算符的支持上，由于不了解单目运算符如何在二叉树中存储，在网上查阅资料、翻阅教材时也未发现有相关的解决方案，由于时间限制只得放弃。另外便是对中缀表达式何时才是必要的括号的条件不太明晰，部分情况还是会出现冗余的括号。希望在未来的学习过程中，随着自身理论知识水平的提高和编码实践能力的提升，本人能够解决上述的问题。

4.4 收获和感悟

总结：我在本次的实训过程中收获很多，既将原来 C 语言、数据结构中的知识点重新复习了，又学到了很多老师在实训中教的新的知识点，将我的知识库更完善了，还提升了自己的代码编写能力。反省：个人在理论上如鱼得水，在动手编程上能力还是较为欠缺，上手能力不是很强，对代码的消化能力还是不足。

首先，在本次的实训练习中，我深入的学到了关于栈的知识点，STL 的部分容器的使用、C++类和对象的基本知识和语法。本次练习中，对 C++有了更加深刻的理解，虽然还算不上很透彻，但学习的过程中，我发现了 C++会比 C 语言更加的灵活。虽然在本学期数据结构的学习中，我内心非常抗拒对代码进行编写，但是一次次的练习，让我对代码有了更加深刻的认识。

其次，我也在本次练习中积累了许多的经验。从对离散数学的基础知识联系到数据结构，在二叉树与中序遍历、波兰和逆波兰表达式里有了更好的理解，明白了计算机是如何将它们之间进行转换的，使我的算法能力更上一层楼。

最后，经历了本次实训，我学到了很多，也从这一次小的实训中，看到了很多以后长远的事情。从原来的一个人慢慢吞吞的敲代码、改 bug 到如今的团队合作，互相交流讨论，相互学习，以后也不会是我一个人单枪匹马的在计算机这条道路上行走，而是一个团队的合作力量，只有大家都参与，每个人都尽自己的一份力量，才能在未来的路上走的更远。在任务中显然要更注重代码的规范性、正式性。以往写程序是总是所有代码都放在一个文件内、甚至直接写在 main 函数中，这样的写法很难完成本次实训较大体量的代码，也不利于团队的合作交流。所以目前我也开始学着将一个长长的代码分成多个文件，一个模块

的代码放在一个文件中、一个原子功能写成一个函数。这样既能提升可读性，也能提升代码的复用性。

综上所述，目前我还应该继续努力，才能在未来的道路上发光发热，才能达到自己想要的目的。在计算机这条道路上，活到老学到老，每天都要学习新知识，每天更好的完善自己，在这一领域尽自己最大的力量，发光发热。

5 参考文献

- [1]严蔚敏 / 吴伟民. 数据结构 (C 语言版) [M]. :清华大学出版社, 2012. -.
- [2][美] 史蒂芬·普拉达 (Stephen Prata). C++ Primer Plus 第 6 版中文版[M]. :人民邮电出版社, 2020. -.
- [3]周晓聪/乔海燕. 离散数学基础[M]. :清华大学出版社, 2021. -.

6. 致谢

“逝者如斯夫，不舍昼夜。”转眼间，我的大一生活匆匆逝去，从一脸懵懵懂懂的进入大一，到现在能够自己独挡一面，我收获了很多，认识了新的事物，失去了旧的事物。我感谢那些出现在我生命中的那些人。

首先，我要感谢我的家人，我的父母将我带到了这个世界上，他们在我灰心的时候给予我鼓励和帮助，在我困难的时候永远陪在我的身旁，是他们让我明白了生命中所谓的意义就是为自己所努力。是家人无微不至的关心，才让我健康茁壮的成长，让我成为了一名积极向上、热爱生活的人。

其次我要感谢我的老师，特别是我大一的班主任——魏雄老师，他是一名严厉中带着一点点幽默细胞的老师，让我在计算机学习中，没有感受到枯燥乏味，更多的是开心快乐，让我觉得学习编程学习计算机是有意义的，让我更乐意去学习以后的知识点。同时也要感谢在课堂上孜孜不倦教导我们知识的老师，课堂上传授知识，课堂下耐心的解答我们学习中不懂的疑问，在我们的学习中起到了很大很好的领导带头作用。还有要感谢我的辅导员，是他在生活中处处关心我们的学习生活健康，让我们有了一个更健康的成长空间。

最后我要感谢我的队友——刘家麒和杨成希，这是我的第一次小组作业，他们的配合以及耐心的帮助我，让我非常感谢，本次实训可以完美结束，这与他们都脱不了关系，每一个人的努力，都是对这个团队的认可。我不仅谢谢我本次合作的队友，我还感谢我下一次合作的队友，感谢他们，因为有他们，我才能成为更好的我。

综上，感谢互联网的发达，让我能在这一行业继续生存下去，让我能学到更多的东西，我谢谢我身处的这一个和平的世界。

成绩记录

一、评语（根据软件成果质量，报告质量，答辩情况综合写出）。

二、评分

评分项目 及分值	软件成果			课程报告		陈述与答辩		合计 (100分)
	任务达成 (20分)	系统与 算法设计 (20分)	代码质量 (10分)	文献学习 方案分析 (20分)	规范表达 (10分)	团队协作 (10分)	表述与回答 (10分)	
得 分								