数据结构实验报告

软件学院 软件工程一班 冯金涛 2211371

实验题目:ds-表达式求值

1. 算法说明

本次题目的名称是表达式求值，题目是给出一个中缀表达式要求将中缀表达式转换为后缀表达式并进行后缀表达式的计算。此次代码的具体实现思路如下。

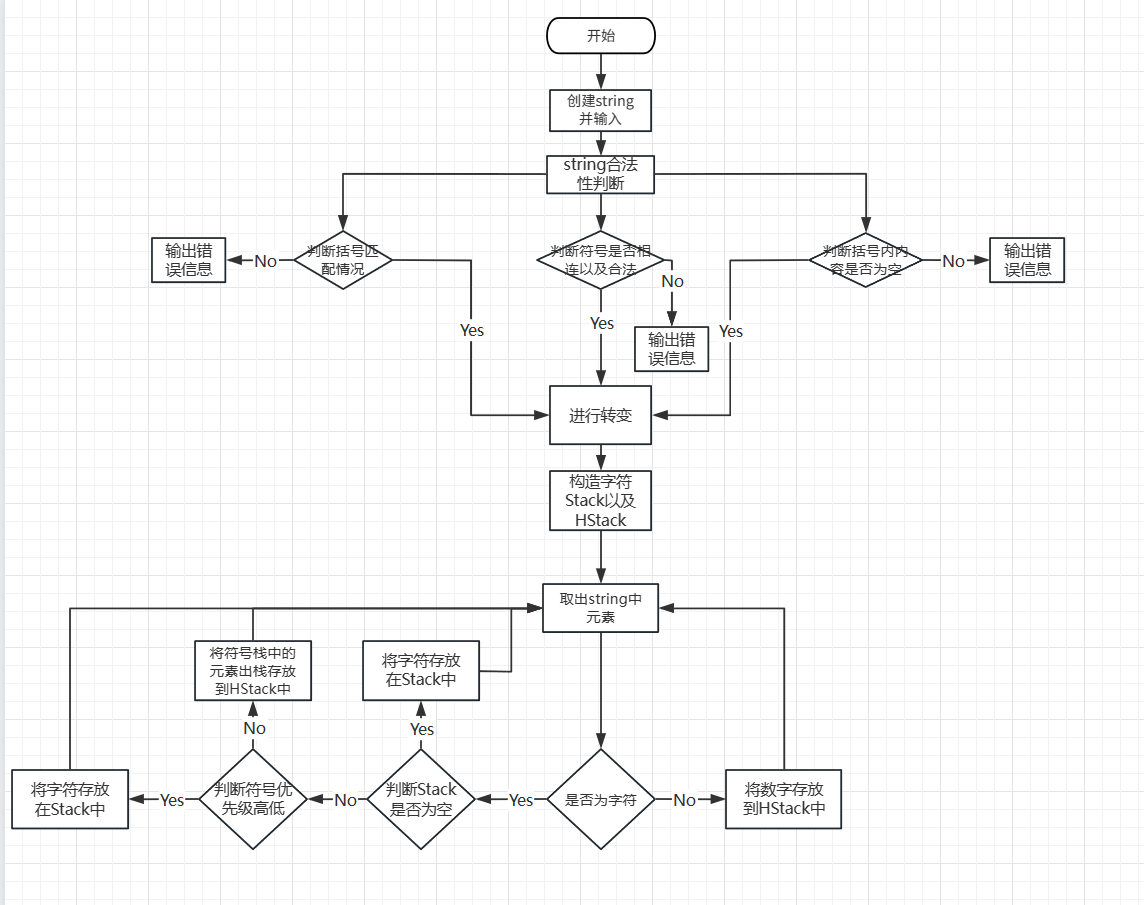
本题主要利用了栈的方法，首先自定义两个栈，分别为存储符号的符号栈以及存放数字的数字栈。以及定义了一个特定的结构体HStack（分别用三个变量，分别存放字符，数字，与字符与数字的判断变量judge），用于对后缀表达式进行存储。

在将中缀表达式转换为后缀表达式之前首先需要判断中缀表达式的输入是否合法。这里我对不同的情况进行了不同的处理方法。在用string接受字符串后，定义判断变量cjudge，若cjudge为0则停止程序运行并输出”ERROR IN INFIX NOTATION”字样，具体对字符串合法性判断的代码将会在下述内容中示出。

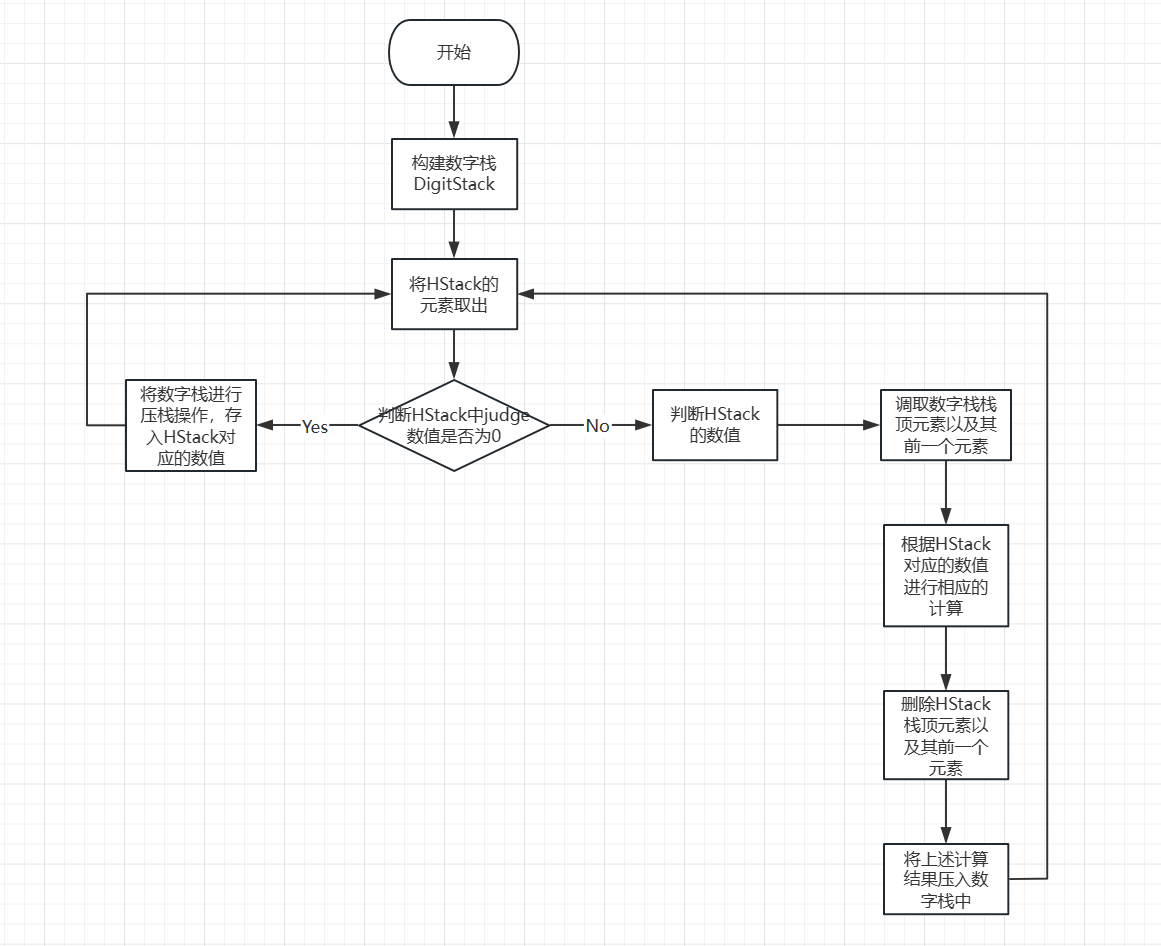
在中缀表达式合法的前提下，进行中缀表达式与后缀表达式的转换。在这里我定义了一个符号栈以及一个HStack数组，同时对string进行遍历，如果遇到数字则直接存放到HStack中，若第一次遇到字符，则直接存放到字符栈之中，在这之后的字符，则需判断此字符与上一个字符的运算优先级，在这里我定义了一个估值函数，对常见的运算符进行了不同的赋值。若进行估值后发现此字符的优先级高于原先存放的字符的优先级，则进行压栈操作，若此字符的优先级低于原字符的优先级，则进行出栈操作。并将出栈的字符存放于HStack数组中，这样就得到了后缀表达式。

在后缀表达式计算方面，首先定义一个数字栈，用于存放数字以及相应的计算结果。接下来遍历HStack数组，若遇到数字，则直接对数字栈进行压栈；若遇到字符，则进行计算函数，将数字栈栈顶位置的数字以及其前一个数字进行计算操作，并将这两个数字出栈删除，将计算结果进行压栈。这样数字栈最后将会存留一个数字，此数字就是后缀表达式的计算结果，也是输入的中缀表达式的计算结果。

1. 流程图

将中缀表达式转换为后缀表达式

后缀表达式的计算



1. 测试结果：

测试案例 测试目的 测试结果

2.5+1.2+3\*5+3^2 测试无括号计算 27.70

(2.5+1.5)+8\*9+12.5/2 测试有括号计算 82.25

()+2.5 测试括号内无输入情况 ERROR IN INFIX NOTATION

\*/+1.5+3.2\*2 测试两个运算符相连的情况 ERROR IN INFIX NOTATION

2.5+3.5+(5+3.5\*6+(6.2+3.2\*2))+6 测试多括号复杂情况 50.60

(2.5+2\*5 测试括号不匹配情况 ERROR IN INFIX NOTATION

(2.5+6.2+(7.2+3.2\*(2.5+6.3) 测试复杂情况下括号不匹配情况 ERROR IN INFIX NOTATION

15+25\*3+9\*8 测试输入数字大于10的情况 162.00

1. 总结与探讨

总结：本次实验基本实现了将中缀表达式转换为后缀表达式并计算的过程。在实验的过程中，测试了多种数据以模拟不同的情况。在代码编写阶段，最难的部分我认为是错误判断方面，在错误判断方面我总是考虑不全，从而在此处花费的时间较多；第二是将中缀转为后缀的过程，在此处我的代码较为繁琐，尤其是对多重括号的处理方面。我在此实验并没有使用stl，这也可能是造成代码过于冗长的原因之一。

在后缀表达式的存储方面，我最开始使用了符号栈作为存放后缀表达式的容器。在之后的过程中我发现若使用符号栈对于数字的判别，尤其是10以上的数字较为困难，代码也较为复杂与繁琐，后我定义了一个特殊变量类型，通过judge的数值来进行数字以及运算符的判断与存储。

在此次实验中，我认为可以改进的地方有很多，比如可以使用stl里面的栈来进行编写，可以直接使用函数，大大减少了工作量，也可以使代码变得更加简洁；其次，过多的在main函数里面编写也是导致此现象的原因之一，在今后我将会修改此代码，加入更多的函数以减少main函数的内容，同时也提升了代码的可读性。

使用栈固然对这道题非常适用，但在中缀转后缀的方面也需要不断的修改，利用遍历的方法较为麻烦，可以去寻找并发现更多更好的方法来分解表达式。