中缀表达式转换为后缀表达式

一、具体步骤如下:

1、 初始化两个栈：运算符栈 s1和储存中间结果（数）的栈 s2；

2、 从左至右扫描中缀表达式；

3、 遇到操作数时，将其压 s2；

4、 遇到运算符时，比较其与 s1栈顶运算符的优先级：

1）如果 s1为空，或栈顶运算符为左括号“(”，则直接将此运算符入栈；

2）否则，若优先级比栈顶运算符的高，也将运算符压入 s1；

3）否则，将 s1栈顶的运算符弹出并压入到 s2中，再次转到(4-1)与 s1中新的栈顶运算符相比较；
ps:括号不做运算符，故遇到括号与运算符比较，直接跳到4.3)

5、遇到括号时：

1) 如果是左括号“(”，则直接压入 s1

2) 如果是右括号“)”，则依次弹出 s1栈顶的运算符，并压入 s2，直到遇到左括号为止，此时将这一对括号丢弃

6、 重复步骤 2至 5，直到表达式的最右边

7、 将 s1中剩余的运算符依次弹出并压入 s2

8、 依次弹出 s2中的元素并输出，结果的逆序即为中缀表达式对应的后缀表达式

二、举例说明:

将中缀表达式“1+((2+3)×4)-5”转换为后缀表达式的过程如下
因此结果为:"1 2 3 + 4 × + 5 - "

中缀表达式1+((2+3)×4)-5 =》后缀表达式

将s2出栈 -5+*4+321 => 123+4*+5-

中缀表达式转后缀表达式的思路步骤分析

打比方：降龙十八掌：学习->应用[层次]

算法->第一个层面：理解算法->灵活运用算法

第二层：设计算法->运用【】

1) 初始化两个栈：运算符栈s1和储存中间结果的栈s2；
2) 从左至右扫描中缀表达式；
3) 遇到操作数时，将其压s2；
4) 遇到运算符时，比较其与s1栈顶运算符的优先级：
1.如果s1为空，或栈顶运算符为左括号“(”，则直接将此运算符入栈；
2.否则，若优先级比栈顶运算符的高，也将运算符压入s1；
3.否则，将s1栈顶的运算符弹出并压入到s2中，再次转到(4.1)与s1中新的栈顶运算符相比较；
5) 遇到括号时：
(1) 如果是左括号“(”，则直接压入s1
(2) 如果是右括号“)”，则依次弹出s1栈顶的运算符，并压入s2，直到遇到左括号为止，此时将这一对括号丢弃
6) 重复步骤2至5，直到表达式的最右边
7) 将s1中剩余的运算符依次弹出并压入s2
8) 依次弹出s2中的元素并输出，**结果的逆序即为中缀表达式对应的后缀表达式**

扫描到的元素	s2(栈底->栈顶)	s1(栈底->栈顶)	说明
1	1	空	数字，直接入栈
+	1	+	s1为空，运算符直接入栈
(	1	+(	左括号，直接入栈
(	1	+((	同上
2	12	+((	数字
+	12	+((+	s1栈顶为左括号，运算符直接入栈
3	123	+((+	数字
)	123+	+(	右括号，弹出运算符直至遇到左括号
×	123+	+(×	s1栈顶为左括号，运算符直接入栈
4	123+4	+(×	数字
)	123+4×	+	右括号，弹出运算符直至遇到左括号
-	123+4×+	-	-与+优先级相同，因此弹出+，再压入-
5	123+4×+5	-	数字
到达最右端	123+4×+5-	空	s1中剩余的运算符