



重慶大學
CHONGQING UNIVERSITY

数据结构实验

时间：2021年（春）--第八周

QQ群：673749661



4. 栈的实现及其应用

● 项目要求

编写程序：

- 1、实现顺序栈；
 - 2、用键盘输入一个包括+、-、*、/、正整数和圆括号的合法算术表达式；
 - 3、先将算术表达式转换成后缀表达式，然后对后缀表达式求值；
 - 4、要求输出：输入的算术表达式、转化后的后缀表达式、计算结果；
 - 5、最后提交完整的实验报告和可运行源程序（.c/.cpp格式）。
-



中缀转后缀



步骤一

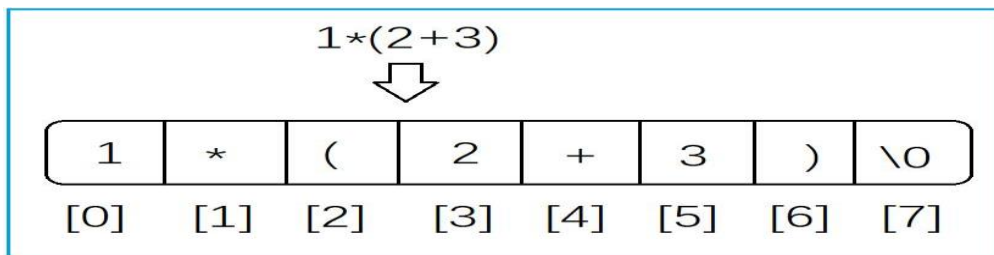
● 算法思想：

- 1、首先构造一个运算符栈，此运算符在栈内遵循越往**栈顶**优先级**越高**的原则。
- 2、读入一个用中缀表示的简单算术表达式，为方便起见，设简单算术表达式的右端多加上了优先级最低的特殊符号“#”。
- 3、从左至右扫描算术表达式，从第一个字符开始判断，如果该字符是**数字**，则分析到该**数字串**的结束并将该数字串直接输出。
- 4、如果不是数字，该字符则是**运算符**，**此时需比较优先关系**。
- 具体做法是：将该字符与运算符栈顶的运算符的优先关系相比较。如果该字符优先关系**高于**此运算符栈顶的运算符，则将该运算符**入栈**。**若不是**的话，则将**栈顶**的运算符从栈中**弹出**，直到栈顶运算符的优先级低于当前运算符，将该字符入栈。
- 5、重复步骤3~4，直至扫描完整个简单算术表达式，确定所有字符都得到正确处理，便可以将中缀式表示的简单算术表达式转化为逆波兰表示的简单算术表达式。

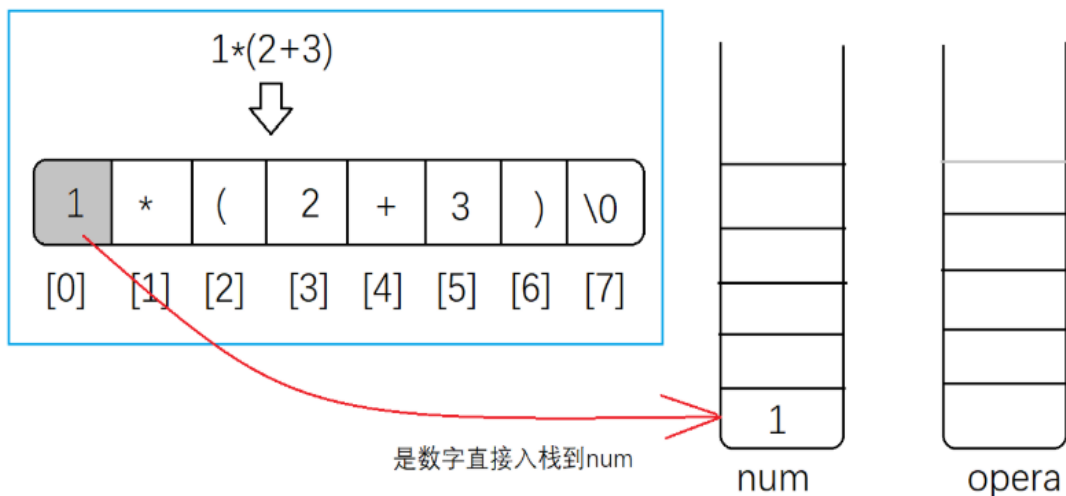


举例 $1*(2+3)$

1、从左往右扫描中缀表达式（这里我们以 $1*(2+3)$ 为例）



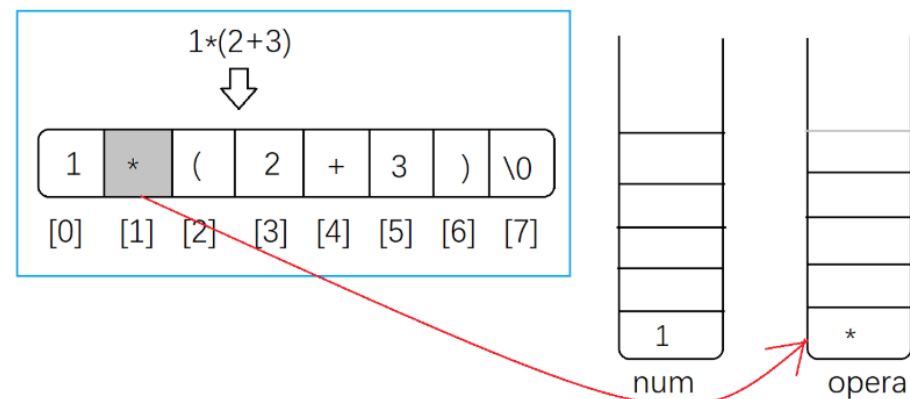
2、“1”是数字那么将数字串输出到数组num中；



3、“*”是操作数，需要进一步判断：

如果是运算符（'+', '-', '*', '/'），先判断数组**opera**的**栈顶**的操作数的优先级

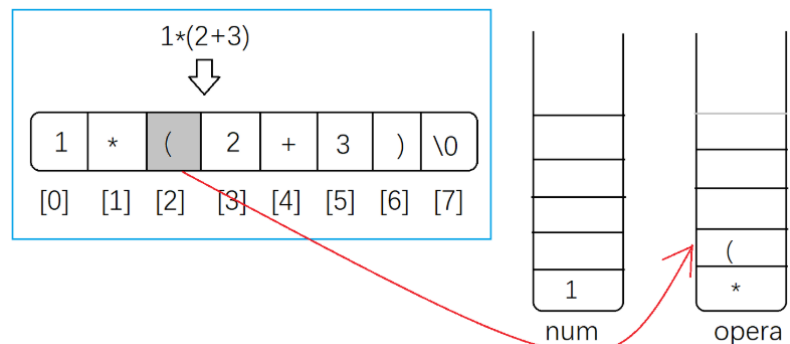
（如果是空栈那么直接入栈到数组**opera**），如果是左括号那么直接入栈到数组**opera**中，如果栈顶是运算符，且栈顶运算符的优先级大于该运算符那么将栈顶的运算符出栈，并将栈顶运算符输出，用数组num记录，重复步骤3，如果栈顶运算符优先级小于当前输入运算符，那么直接将该输入运算符入栈**opera**中。



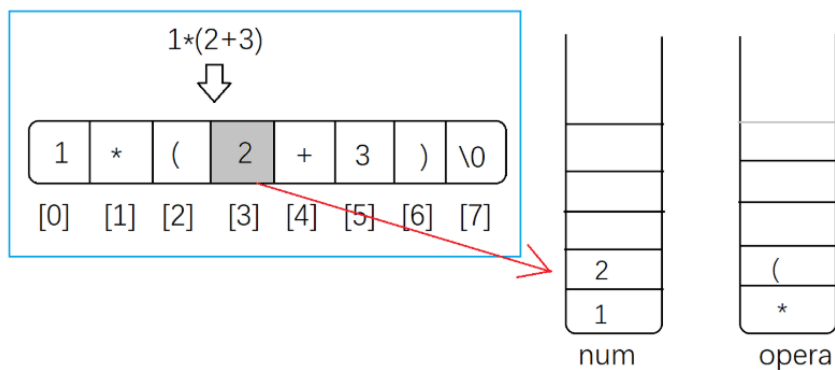


举例 $1*(2+3)$

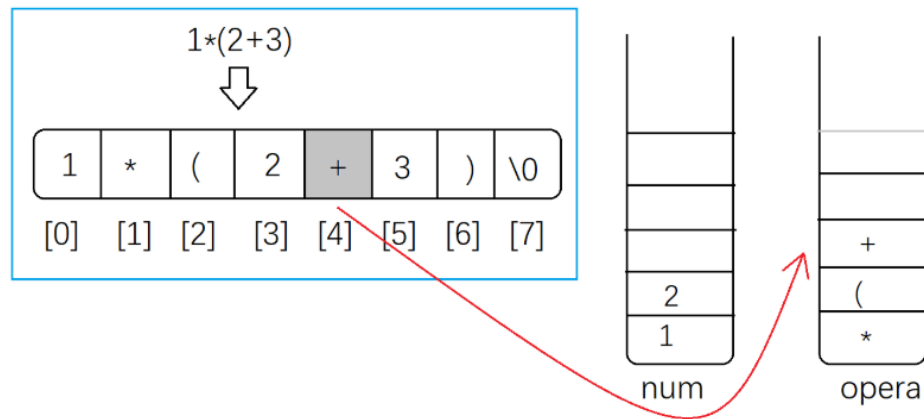
4、“(” 是操作数，需要进一步判断：
左括号“(”直接入栈到**opera**中



5、“2” 是数字那么将数字串输出，并用数组num记录；



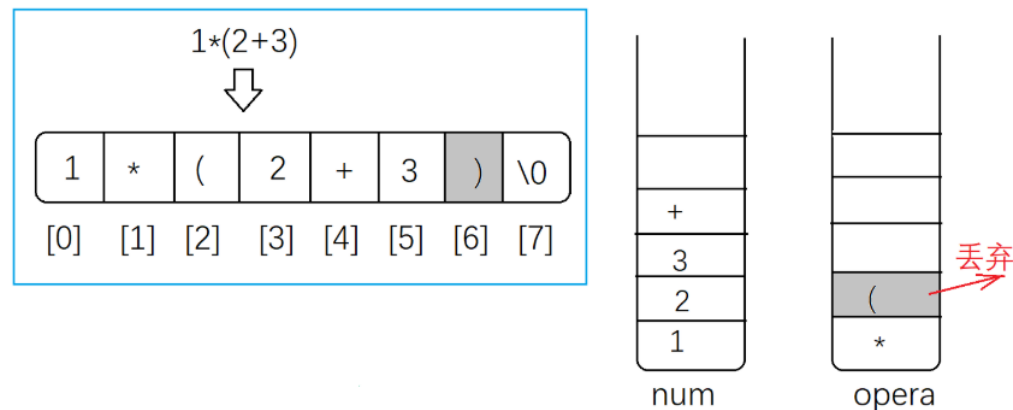
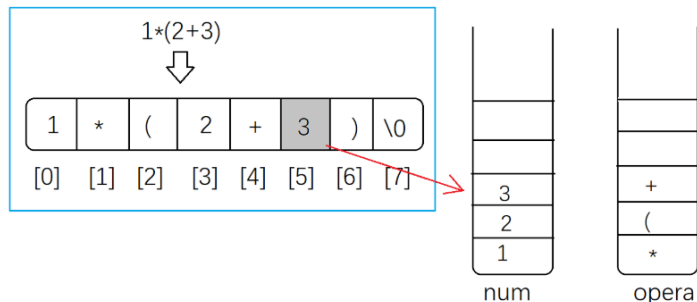
6、“+” 是操作数，需要进一步判断：
如果是运算符（'+', '-', '*', '/'），先判断
数组**opera**的**栈顶**的操作数的优先级
如果栈顶是运算符，且栈顶运算符的优先级大于该
运算符那么将栈顶的运算符出栈，并入栈到数组
num中，重复步骤3，如果栈顶运算符优先级小于该
运算符，那么直接将该运算符入栈**opera**中。此时+
入栈。





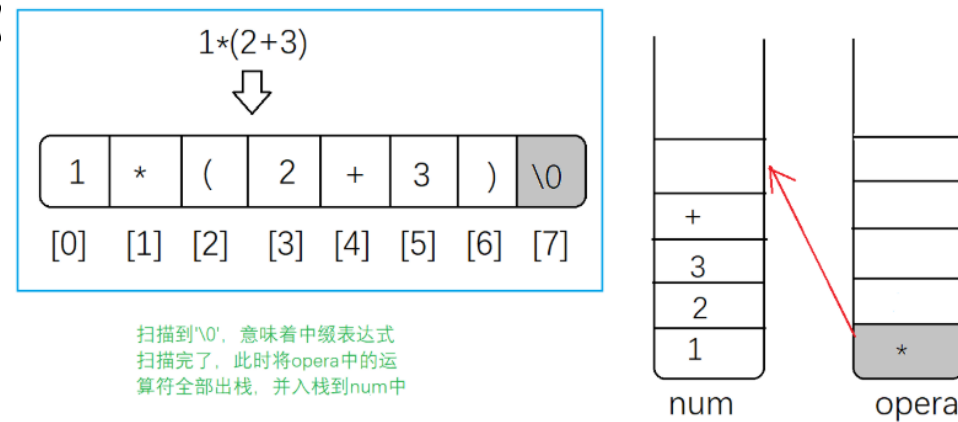
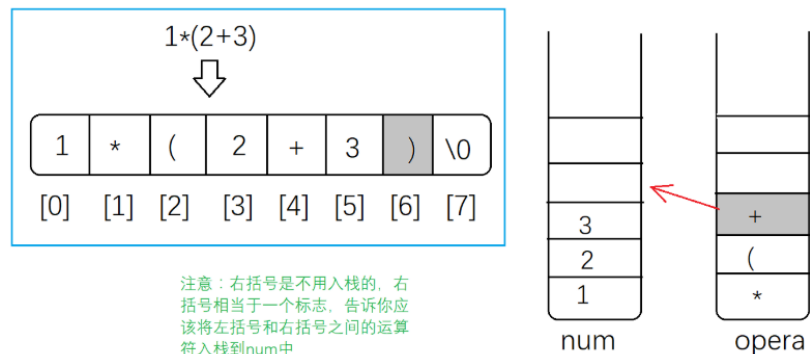
举例 $1*(2+3)$

7、 “3” 是数字那么将数字串输出到数组num中;



8、 “)” 是操作数，需要进一步判断:

如果是右括号')'，那么说明在**opera**数组中一定有一个左括号与之对应，那么将**opera**中的运算符依次出栈，并输出到**num**中，直到遇到左括号'(', 注意左括号不用输出到**num**，





后缀表达式计算



步骤二

● 算法思想：

- 1、需要使用栈的数据结构；
- 2、当遇到数值时，入栈；
- 3、当遇到运算符的时候，连续出栈两次（即取栈最上面的两个数字），对这两个数字结合运算符进行计算；
- 4、将第一次出栈元素放在操作符后面，第二次出栈元素放在操作符前面，使用中缀表达式的计算规则进行计算，然后将计算出的结果入栈；
- 5、不断循环，直到后缀表达式中的元素全部进行了操作。栈顶元素为计结果。



代码分析

```
/*中缀转后缀函数*/
char* Change(SqStackOp *S,SElemTypeOp str[]){
    char* exp; int i=0,j=0; SElemTypeOp e,top;
    InitStack(S);
    printf("后缀表达式 = ");
    while(str[i]!='#'){
        if(isdigit(str[i])){//如果是数字
            int temp;
            temp = str[i] - '0';
            exp3[j++] = str[i];
            i++;//后移一位,因为数字不一定是个位数
            while (str[i]!='#' && isdigit(str[i])){
                temp = temp * 10 + (str[i] - '0');
                exp3[j++] = str[i];
                i++;
            }
            //Push(S,str[i]);
            printf("%d ",temp);
            exp3[j++] = ' ';
        }.....}}
}
```




代码分析

```
char* Change(SqStackOp *S,SElemTypeOp str[]){ .....
    while(str[i]!='#'){ .....
        else if(In(str[i], OP)){//如果是操作符
            if(str[i] == '('){//3.遇到左括号： 将其入栈
                Push(S,str[i++]);
            }else if(str[i] == ')'){//4.遇到右括号： 执行出栈操作， 输出到后缀表达式， 直到弹出的是左括号
                while( !StackEmpty(*S) && GetTop(*S,&top) != '('){
                    exp3[j++] = top;
                    exp3[j++] = ' ';
                    printf("%c ",top);
                    Pop(S,&e);
                }
                Pop(S,&e);//弹出左括号
                i++;
            }else{//5.遇到其他运算符： 弹出所有优先加大于或等于该运算符的栈顶元素， 然后将该运算符入栈
                while (!StackEmpty(*S) && (Precede(GetTop(*S,&e), str[i]) == '>' || Precede(GetTop(*S,&e), str[i]) == '=')){
                    exp3[j++] = e;
                    printf("%c ",e);
                    exp3[j++] = ' ';
                    Pop(S,&e);}
                Push(S,str[i]);
                i++;}}}}}
```



代码分析

```
char* Change(SqStackOp *S,SElemTypeOp str[]){  
  
    .....  
  
    /*最后把栈中剩余的运算符依次弹栈打印*/  
    while(StackLength(*S)){  
        Pop(S,&e);  
        exp3[j++] = e;  
        exp3[j] = ' ';  
        printf("%c ",e);  
    }  
    printf("\n");  
    exp3[j] = '\0';  
    exp = exp3;  
    //printf("后缀表达式 = %s",exp3);  
    return exp;  
}
```



重慶大學
CHONGQING UNIVERSITY

感谢聆听!

时间: 2021年 (春) --第八周

QQ群: 673749661