第04课表达式求值

为什么要使用栈?

别的方法都不容易搞定。比如：多层括号

表达式求值=中缀表达式转后缀表达式+后缀表达式求值

后缀表达式求值

也要用到栈：此栈只存放操作数

做从左向右扫描，求值过程如下：
1 遇到数字字符，转换成数值并入操作数栈
2 遇到操作符，出栈两个操作数，并计算，再将结果入栈

> 比如：1）中缀表达式 15+(24-6)/4*2+18 如何转后缀表达式 15#24#6#-4#/2#*+18#+

后缀表达式的逻辑和实现方式（逆波兰表达式求值）

1.定义

如果每个操作符跟在它的两个操作数之后，而不是两个操作数之间，那么这个表达式就是后缀表达，又称为逆波兰表达式，如：3 5 + 7 * 1 -

2.后缀表达式计算机求值

1.与前缀表达式类似，只是顺序是从左至右：
2.从左至右扫描表达式，遇到数字时，将数字压入堆栈，遇到运算符时，弹出栈顶的两个数，其中先出栈的是右操作数，后出栈的是左操作数，
3.用运算符对它们做相应的计算（次顶元素 op 栈顶元素），并将结果入栈；
4.重复上述过程直到表达式最右端，最后运算得出的值即为表达式的结果

3.例子

计算后缀表达式的值：1 2 3 + 4 × + 5 -
1）从左至右扫描，将1，2，3压入栈；
2）遇到+运算符，3和2弹出，计算2+3的值，得到5，将5压入栈；
3）遇到4，将4压入栈
4）遇到×运算符，弹出4和5，计算5×4的值，得到20，将20压入栈；
5）遇到+运算符，弹出20和1，计算1+20的值，得到21，将21压入栈；
6）遇到5，将5压入栈；
7）遇到-运算符，弹出5和21，计算21-5的值，得到16为最终结果

作业任务

任务1：表达式求值

输入一个中缀表达式，程序能输出正确的后缀表达式和求值结果

```
请输入中缀表达式：
81/3+67*(3+12)-24*5
转换的后缀表达式为：
81#3#/67#3#12#+*+24#5#*-
表达式值为：
912
```

任务2：力扣原题--逆波兰表达式求值

150. 逆波兰表达式求值

难度 中等　662　收藏　分享　切换为英文　接收动态　反馈

给你一个字符串数组 `tokens` ，表示一个根据 逆波兰表示法 表示的算术表达式。

请你计算该表达式。返回一个表示表达式值的整数。

注意：

- 有效的算符为 `'+'` 、 `'-'` 、 `'*'` 和 `'/'` 。
- 每个操作数（运算对象）都可以是一个整数或者另一个表达式。
- 两个整数之间的除法总是 **向零截断** 。
- 表达式中不含除零运算。
- 输入是一个根据逆波兰表示法表示的算术表达式。
- 答案及所有中间计算结果可以用 **32 位** 整数表示。

示例 1：

输入： tokens = ["2","1","+","3","*"]
输出： 9
解释： 该算式转化为常见的中缀算术表达式为：((2 + 1) * 3) = 9

示例 2：

输入： tokens = ["4","13","5","/","+"]
输出： 6
解释： 该算式转化为常见的中缀算术表达式为：(4 + (13 / 5)) = 6

示例 3：

输入： tokens = ["10","6","9","3","+","-11","*","/","*","17","+","5","+"]
输出： 22
解释： 该算式转化为常见的中缀算术表达式为：

```
  ((10 * (6 / ((9 + 3) * -11))) + 17) + 5
= ((10 * (6 / (12 * -11))) + 17) + 5
= ((10 * (6 / -132)) + 17) + 5
= ((10 * 0) + 17) + 5
= (0 + 17) + 5
= 17 + 5
= 22
```

提示：

- `1 <= tokens.length <= 10^4`
- `tokens[i]` 是一个算符（`"+"`、`"-"`、`"*"` 或 `"/"`），或是在范围 `[-200, 200]` 内的一个整数

逆波兰表达式：

逆波兰表达式是一种后缀表达式，所谓后缀就是指算符写在后面。

- 平常使用的算式则是一种中缀表达式，如 `( 1 + 2 ) * ( 3 + 4 )` 。
- 该算式的逆波兰表达式写法为 `( ( 1 2 + ) ( 3 4 + ) * )` 。

逆波兰表达式主要有以下两个优点：

- 去掉括号后表达式无歧义，上式即便写成 `1 2 + 3 4 + *` 也可以依据次序计算出正确结果。
- 适合用栈操作运算：遇到数字则入栈；遇到算符则取出栈顶两个数字进行计算，并将结果压入栈中