

表达式求值相关内容

中缀表达式转化为前缀表达式

1. 算法描述

1. 首先构造一个运算符栈 $s1$ 和一个存储中间结果的栈 $s2$ 。
2. 从右至左扫描中缀表达式。
3. 如果是操作数，将其压入 $s2$ 。
4. 如果是运算符，则将 $s1$ 栈顶元素与该运算符比较优先级：
 - a. 如果 $s1$ 为空，或栈顶运算符为右括号 $)$ ，则直接将此运算符入栈；
 - b. 否则，若该运算符优先级比栈顶运算符的优先级高或相等，则将运算符压入 $s1$ ；
 - c. 否则，将 $s1$ 栈顶的运算符弹出并压入到 $s2$ 中，再与 $s1$ 中新的栈顶运算符相比较。
5. 遇到括号时：
 - a. 如果是右括号 $)$ ，则直接压入 $s1$ ；
 - b. 如果是左括号 $($ ，则依次弹出 $s1$ 栈顶的运算符，并压入 $s2$ ，直到遇到右括号为止，此时将这一对括号丢弃；
6. 重复步骤 2 至 5，直到表达式的最大长度；
7. 若表达式还未扫描完，将 $s1$ 中剩余的运算符依次弹出 $s2$ ；
8. 最后将 $s2$ 中的元素逆序，结果即为对应的前缀表达式。

中缀表达式转化为后缀表达式

1. 算法描述

1. 首先构造一个运算符栈 $s1$ 和一个存储中间结果的栈 $s2$ 。
2. 从左至右扫描中缀表达式。
3. 如果是操作数，将其压入 $s2$ 。
4. 如果是运算符，需比较 $s1$ 栈顶元素的优先级：
 - a. 如果 $s1$ 为空，或栈顶元素为左括号 $($ ，则直接将此运算符入栈；
 - b. 否则，若该运算符优先级比栈顶运算符的优先级高或相等，则将运算符压入 $s1$ ；
 - c. 否则，将 $s1$ 栈顶的运算符弹出并压入到 $s2$ 中，再与 $s1$ 中新的栈顶运算符相比较。
5. 遇到括号时：

- a. 如果是右括号 $)$ ，则依次弹出 s_1 栈顶的运算符并压入到 s_2 ，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，直到表达式的扫描完成；
 7. 若表达式还未扫描完，将 s_1 中剩余的运算符依次弹入 s_2 ；
 8. s_2 中的元素即为对应的后缀表达式。

前缀表达式的规则与计算机求值

1. 前缀表达式的计算机值

1. 从右至左扫描表达式，遇到数字时，将数字压入栈；
2. 遇到运算符时，弹出栈顶的两个数字，用运算符对它们做相应的计算（栈顶元素 op 次顶元素）；
3. 并将结果入栈；重复上述过程直到表达式最左端，最后运算得出的值即为表达式的结果。

2. 例子

计算前缀表达式的值： $- + 1 * + 2 3 4 5$

1. 从右至左扫描，将 5、4、3、2 压入栈；
2. 遇到 $+$ 运算符，弹出 2 和 3（2 为栈顶元素，3 为次顶元素），计算 $2 + 3$ 的值，得到 5，将 5 压入栈；
3. 遇到 $*$ 运算符，弹出 5 和 4，计算 $5 * 4$ 的值，得到 20，将 20 压入栈；
4. 遇到 $+$ 运算符，弹出 1 和 20，计算 $1 + 20$ 的值，得到 21，将 21 压入栈；
5. 遇到 $-$ 运算符，弹出 21 和 5，计算 $21 - 5$ 的值，得到 16，为最终结果。

中缀表达式的规则

1. 规则

1. 算符优先级：
 - 先计算算符内的，后计算算符外的。
2. 无括号或层次括号的优先级：
 - 类似于运算，运算符的优先级高于加减运算的优先级。
3. 相同优先级运算：
 - 从左向右依次进行。

2. 中缀表达式的计算机求值

1. 设置数据栈：

- 一个数字栈用来存储中缀表达式中的数字。
- 另一个操作符栈用于存储操作符。

2. 扫描表达式：

- 如果遇到数字，直接入栈。
- 如果遇到算符，则判断其优先级，并根据优先级进行操作。
- 对于底层运算符，应考虑栈中元素的优先级和顺序。

3. 数字与算符的处理：

- 在处理过程中，若发现算符栈为空，则直接将数字压入栈。
- 若有运算符，则按照优先级进行操作。最终的结果在数字栈的顶部。

后缀表达式求值

1. 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. 从左至右扫描后缀表达式，遇到数字时，将数字压入栈，遇到运算符时，弹出栈顶的两个数，其中先弹出的为操作数，后弹出的的是左操作数；
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. 重复上述过程直到达到表达式最右端，最后运算得出的值即为表达式的结果。

2. 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，弹出 3 和 2（3 为栈顶元素，2 为次顶元素），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 21 和 5，计算 $21 - 5$ 的值，得到 16，为最终结果。

参考来源 [csdn博客](#)

@title: 表达式求值

@date: 2025-01-08 19:00:00

@version: 1.0.0

@copyright: Copyright (c) 2025 数据结构期末复习

@author: 软件工程宋浩元