数据结构与算法（Python版）

后缀表达式求值

陈斌 北京大学 gischen@pku.edu.cn

后缀表达式求值

- 作为栈结构的结束，我们来讨论“**后缀表达式求值**”问题
- 跟中缀转换为后缀问题不同，
- 在对后缀表达式从左到右扫描的过程中，
- 由于操作符在操作数的**后面**，
- 所以要**暂存操作数**，在碰到操作符的时候，再将暂存的两个操作数进行实际的计算

  仍然是栈的特性：操作符只作用于离它最近的两个操作数

后缀表达式求值

- 如“4 5 6 * +”，我们先扫描到4、5两个操作数
- 但还不知道对这两个操作数能做什么计算，需要继续扫描后面的符号才能知道
- 继续扫描，又碰到操作数6
- 还是不能知道如何计算，继续暂存入栈
- 直到“*”，现在知道是栈顶两个操作数5、6做乘法

后缀表达式求值

- 我们弹出两个操作数，计算得到结果30

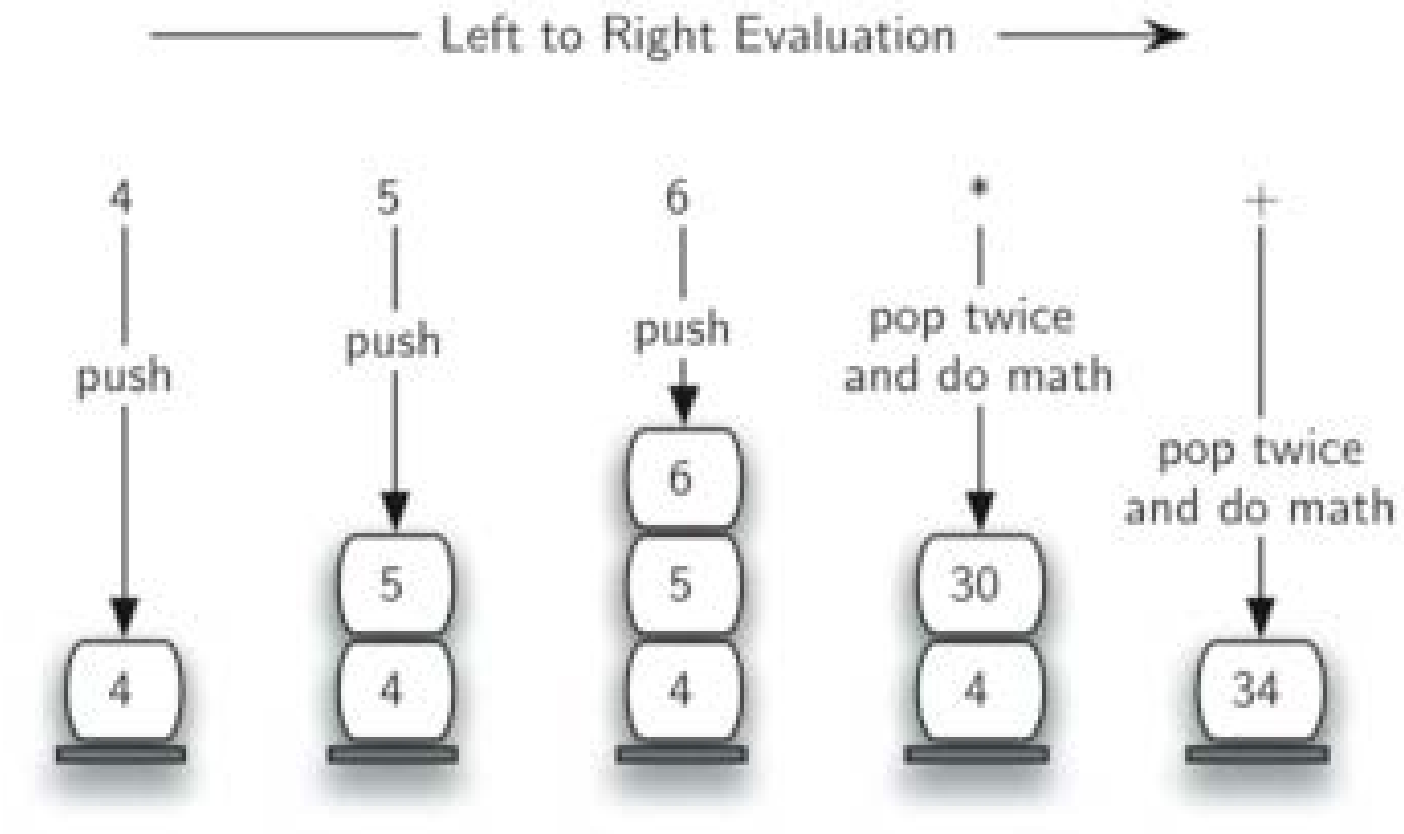
  需要注意：

  先弹出的是右操作数

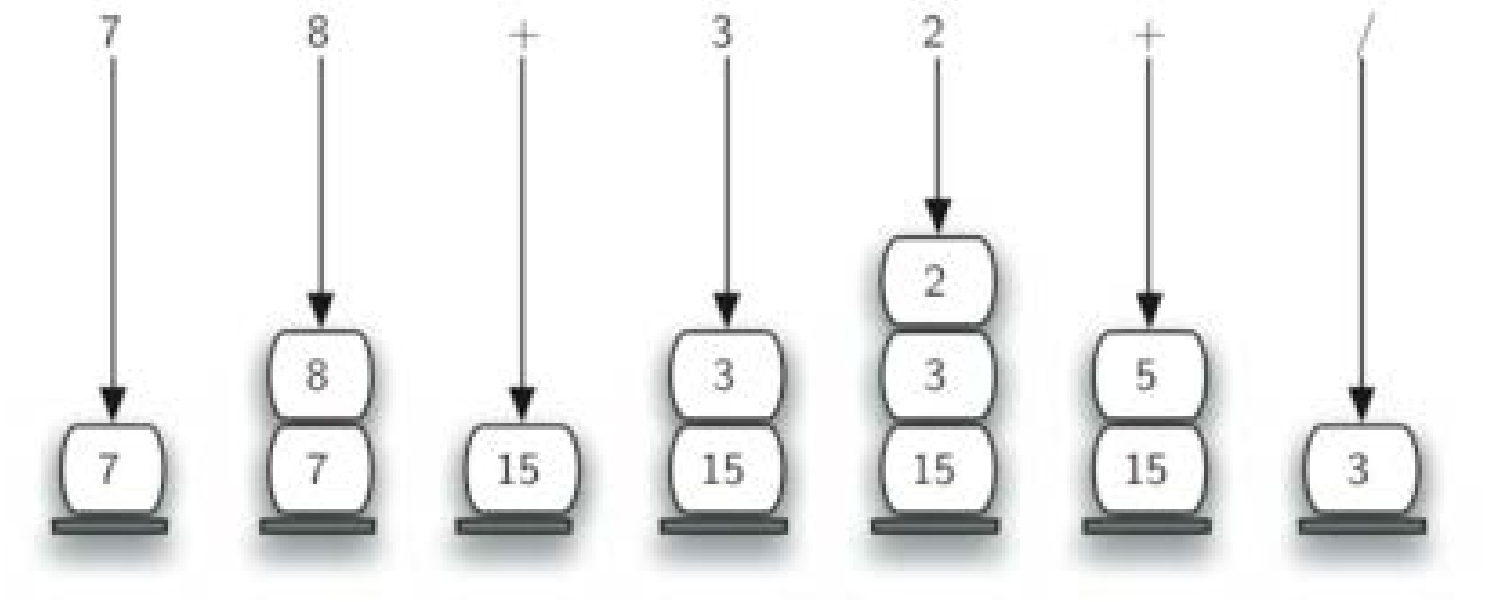
  后弹出的是左操作数，这个对于-/很重要！

- 为了继续后续的计算，需要把这个中间结果30压入栈顶
- 继续扫描后面的符号
- 当所有操作符都处理完毕，栈中只留下1个操作数，就是表达式的值

后缀表达式求值：实例


Left to Right Evaluation
4
push
4
5
push
5
4
6
push
6
5
4
*
pop twice
and do math
30
4
+
pop twice
and do math
34

后缀表达式求值：实例



后缀表达式求值：流程

- 创建空栈operandStack用于暂存操作数
- 将后缀表达式用split方法解析为单词（token）的列表
- 从左到右扫描单词列表

  如果单词是一个操作数，将单词转换为整数int，压入operandStack栈顶

  如果单词是一个操作符（*/+-），就开始求值，从栈顶弹出2个操作数，先弹出的是右操作数，后弹出的是左操作数，计算后将值重新压入栈顶
- 单词列表扫描结束后，表达式的值就在栈顶
- 弹出栈顶的值，返回。

```
def postfixEval(postfixExpr):
    operandStack = Stack()
    tokenList = postfixExpr.split()

    for token in tokenList:
        if token in "0123456789":
            operandStack.push(int(token))
        else:
            operand2 = operandStack.pop()
            operand1 = operandStack.pop()
            result = doMath(token,operand1,operand2)
            operandStack.push(result)
    return operandStack.pop()

def doMath(op, op1, op2):
    if op == "*":
        return op1 * op2
    elif op == "/":
        return op1 / op2
    elif op == "+":
        return op1 + op2
    else:
        return op1 - op2
```

操作数

操作符