

分治

题目: 241. Different Ways to Add Parentheses

语言: python3

英文版链接: <https://leetcode.com/problems/different-ways-to-add-parentheses/description/>

中文版链接: <https://leetcode-cn.com/problems/different-ways-to-add-parentheses/description/>

题目分析

分治, 即分而治之, 就是把一个复杂的问题分成两个或更多的相同或相似的子问题, 再把子问题分成更小的子问题.....直到最后子问题可以简单的直接求解, 原问题的解即子问题的解的合并。

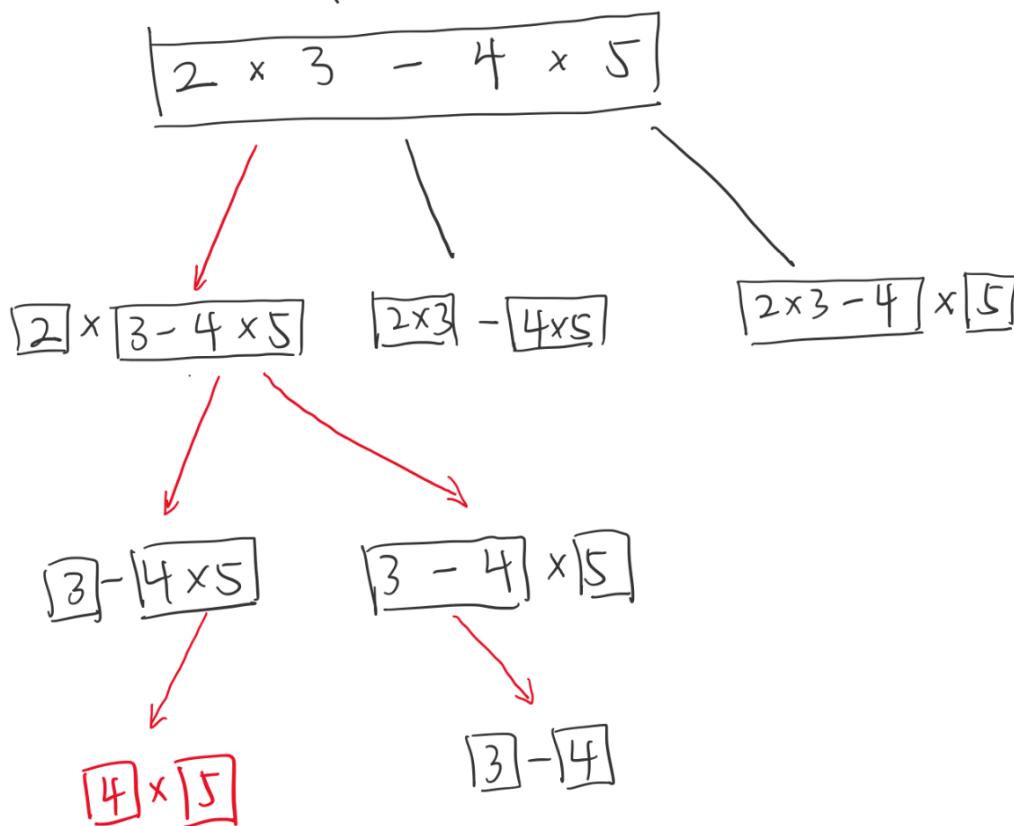
题目要求: 给定一个含有数字和运算符的字符串, 为表达式添加括号, 改变其运算优先级以求出不同的结果。你需要给出所有可能的组合的结果。有效的运算符包含+, -, *。

我们用例子来说明如何使用分治来解决这个问题:

$2 * 3 - 4 * 5$

我们可以将第一个运算符前后分为两个子问题, 即2和 $3 - 4 * 5$, 第一个子问题直接返回2, 第二个子问题继续可以分成两个子问题3和 $4 * 5$ 。然后继续往下分。

我们划个简单的图来说明这个问题 (图丑了点不要介意:)) :



一个大问题，我们可以根据运算符分成相应的小问题；第一个小问题是2，直接从字符串转成int返回，第二个小问题是 $3 - 4 \times 5$ ，又可以转化成四个小问题。这就是分而治之，分治思想。

我们常用来解决分治的方法是递归，递归需要注意递归推进和递归返回（或者叫做边界条件）。递归推进就是从大问题转化为小问题，从“情况n”变化到情况“n+1”的过程，这里我们直接将运算符两侧的字符串分别作为一个子问题，压入下一个递归即可。边界条件是当我们没有运算符的时候返回对应的整数，或者返回对应子问题的结果。需要注意，当子问题返回了结果之后，需要根据对应的运算符，得到父问题的结果，再进行返回。

比如， 4×5 的结果得到了之后，我们要算3和 4×5 两个子问题返回结果相减后的结果再返回。

答案

```
class Solution:
    def diffWaysToCompute(self, input: str) -> List[int]:
        return_list = []
        for i in range(len(input)):
            c = input[i]
            if c in ['+', '-', '*']:
                left = self.diffWaysToCompute(input[:i])
                right = self.diffWaysToCompute(input[i+1:])
                for l in left:
                    for r in right:
                        if c == '+':
                            return_list.append(l + r)
                        elif c == '-':
                            return_list.append(l - r)
```

```
        elif c == '*':  
            return_list.append(l * r)  
if not return_list:  
    return_list.append(int(input))  
return return_list
```