表达式计数

【问题描述】

对于N个变量，在可以使用括号的基础上，有多少种本质不同的运算方案？由于答案会很大，所以只需要输出模1000000007之后的答案。

两个方案本质相同即，无论如何改变这些变量的值，两种方案的结果不变(当然，要在表达式有意义的基础上)

例如：$a/b/c$和$a/(b*c)$的本质是相同的。

【试题考察点】

动态规划

【解题报告】

这道题是一道经典的Dp题。

首先，我们需要对本质不同进行深入理解。

可以从以下三个方面下手：

1. 对每一层都进行括号操作，同层内部的括号用相同优先级的符号连接，相邻两层的连接符号一定不相同，比如对于表达式$(a+b)*c+\left(\frac{d}{e}*f\right)$可以把它写成$\left(\left((a+b)*c\right)+\left(\frac{d}{e}*f\right)\right)$
2. 对于同一层的符号，规定把加和乘放到前面，比如对于$a-b-c+d$可以写成$a+d-b-c$
3. 然而这样还会有重复，比如对于两个式子$(a-b)*(c-d),(b-a)*(d-c)$，本质是一样的，所以还需要减掉这些方案（关键）。

前两方面：

如果只要考虑前面两个方面，那么很容易就能得到Dp方程：用f_i表示有i个变量的且最外层是加减层的方案数，用g_i表示有i个变量的且最外层是乘除层的方案数。然后按第二点那样分前后两半做，枚举最后一个括号的大小，然后用组合数乘一下就得到答案了。

以f_i为例，转移大概如下(上面所说的前后两半可以自己考虑)：

$$f_i=\sum_{j=1}^{i-1} f_{i-j}*g_j*C_{i-1}^{j-1}$$

by Eternal_Answer

关键的第三方面：

但是对于第三点，用上面的方法不能很有效的去重，因此要再挖掘第三点：首先，这样的重复只会出现在乘除层，而出现重复的必要条件是：存在偶数个加减层括号能够置反。

而一个括号可以置反的前提是它存在没有前导符号的相反项。例如：$a+b$是不能置反的，因为置反后为$-a-b$，无论如何交换顺序都必然存在有前导负号的项，而$a-b$是可置反的，因为它能变成$b-a$。而对于一个乘除层，他如果能置反，充要条件是至少存在一个括号(加减层括号)能置反，而对于一个加减层，如果在当前层有负号,那么一定能置反,否则需要有一个括号(乘除层括号)能置反，从而在当前层构造出一个负号。因此，通过分析可以发现，一个表达式存在置反方案的充要条件是存在负号！

现在知道了，存在置反的充要条件是存在负号，那么也就是说一个表达式不存在置反方案的充要条件为不存在负号。

那么我们可以先把表达式置反前后的方案当成一个方案，即绝对值相同的方案当成一个方案，因此对于一个N个变量的表达式，我们可以先算出绝对值不同的方案数ans_1,那么对于每个方案，假设它都存在置反方案，那么总答案就是$ans=ans_1*2$,然而，由于不存在负号的方案是没有置反方案的，所以可以在计算出不存在负号的方案数ans_2,那么答案就是$ans=ans_1*2-ans_2$。

就此，问题的基本思路已经得到了，那么接下来就是如何计算这两种方案。

我们可以定义以下几个Dp状态：

f_i：外面层是加减层的绝对值不同的方案数。

ff_i：不包含减号的，最外面层是加减层的方案数。

g_i:最外面层是乘除层的绝对值不同的方案数。

gg_i：不包含减号的，最外面层是乘除层的方案数。

$g1_i$:最外面层仅有乘法连接的绝对值不同的方案数。

$gg1_i$：不包含减号的，最外面层仅有乘法连接的方案数。

其中$g1_i$和$gg1_i$是用来帮忙转移g_i和gg_i的。

各个Dp递推的转移方程大概如下：

$$f_i=\sum_{j=1}^{i-1}\left(f_{i-j}+g_{i-j}\right)*g_j*2*C_{i-1}^{j-1}$$

$$ff_i=\sum_{j=1}^{i-1}\left(ff_{i-j}+gg_{i-j}\right)*gg_j*C_{i-1}^{j-1}$$

$$g1_i=\sum_{j=1}^{i}g1_{i-j}*f_j*C_{i-1}^{j-1}$$

$$gg1_i=\sum_{j=1}^{i}gg1_{i-j}*ggf_j*C_{i-1}^{j-1}$$

$$g_i=\sum_{j=1}^{i}\left(g1_j-f_j\right)*g1_{i-j}*C_i^j$$

$$gg_i = \sum_{j=1}^{i} (gg1_j - ff_j) * gg1_{i-j} * C_i^j$$

当然，还是有一些细节需要考虑的，比如$j = i - 1$的时候的情况。

空间复杂度：$O(N^2)$
时间复杂度：$O(N^2)$
期望得分：100分

【小结】

这道题是一道比较难的经典Dp题，需要选手在想题和写代码的时候有所细心，不能漏过任何一种情况，特别是去重方面，是这道题的难点所在，同时也需要选手对Dp有比较好的掌握。在这场互测中属于较难题。

by Eternal_Answer