

NOI练习赛

276f280b19d8e63bd6f992d88bbf96b2

中文题目名称	一氯取代	不开心	打字机
英文题目名称	chem	unh	typewriter
输入文件名	chem.in	unh.in	typewriter.in
输出文件名	chem.out	unh.out	typewriter.out
每个测试点时限	1s	1s	1s
测试点数目	10	10	20
每个测试点分值	10	10	5
内存限制	256MB	32MB	256MB
是否有部分分	否	否	否
题目类型	传统	传统	传统

1 化学题

1.1 题目描述

小P和小Z是一对非常好的朋友，今天他们在做一道化学题。

这道题目也非常简单，就是求一个链烷烃的一氯取代物的种数。

众所周知，一个链烷烃可以看作一个由H和C构成的树，其中所有H的度数为1，C的度数为4。

对于一个链烷烃，我们可以从它的任何一个H出发进行深度优先搜索，得到的节点序列称为DFS序列。例如甲烷的DFS序列是HCHHH，乙烷是HCHHCHHH（注意，也有其他表示方法例如HCCHHHHH等）。已知了DFS序列，那么链烷烃的结构就确定了。

我们就输入一个链烷烃的DFS序列，求它的不同的一氯取代物的种数。注意我们这里只考虑官能团位置异构，不考虑价键异构、立体异构等。

1.2 输入格式

输入文件为chem.in。

只有一行，表示这个链烷烃的DFS序列。

1.3 输出格式

输出文件为chem.out。

只有一个整数，表示它的不同的一氯取代物的种数。

1.4 样例输入1

HCHHH

1.5 样例输出1

1

1.6 样例输入2

HCCHHHCHHHCHHH

1.7 样例输出2

2

1.8 样例输入3

HCCCHHHCHHHCHHHHH

1.9 样例输出3

1

1.10 样例解释

三个样例分别是甲烷、异丁烷、新戊烷。

1.11 数据规模与约定

对于所有数据，输入文件是随机生成的。令 n 为输入串的长度，则：

测试点编号	n
1	14
2	17
3	17
4	302
5	302
6	3002
7	3002
8	300002
9	300002
10	300002

2 不开心

2.1 题目描述

某部门有 n 个人，从1到 n 编号，他们之间有着复杂的上下级关系。

1号是这个部门的总管，也就是说他没有上司。除了他以外，每个人都有且只有一个直接的上司， i 号的直接的上司的编号为 $boss_i$ 。而且，不会出现某个人的若干级上司是他自己的情况。

对应地，一些人会有直接的下属，并且可能会有多个，我们称直接的下属为1级下属，而直接的下属的直接的下属为2级下属。同样地，我们可以定义一个人的 k 级下属。

一开始，所有人都不满。然而现在，由于这个部门不太和谐，一些人可能会产生不满。同时，为了自身的利益，有些人想知道，自己的下属的不满意度之和。由于过于高级或低级的下属对自己的影响都不是很大，他这次只会关心他的某些级别的下属。

2.2 输入格式

输入文件为unh.in。

第一行为两个整数 n q ，代表人数和操作数。

第二行为 $(n - 1)$ 个整数，分别代表2号到 n 号的直接的上司的编号，即第 i 个数表示 $boss_{i+1}$ 。

接下来 q 行，每行有两种可能的格式：

1 x y ，表示：编号为 x 的人产生了 y 点不满意；

2 x l r ，表示：求编号为 x 的人的 l 到 r 级下属的不满意度之和。

2.3 输出格式

输出文件为unh.out。

一共输出 q 行。

如果对应的是操作1，则输出一行-1。

如果对应的是操作2，则输出询问的答案。

2.4 样例输入

```
6 5
1 2 2 1 4
2 2 1 2
1 6 1086
2 2 1 2
2 2 1 1
2 2 1 5
```

2.5 样例输出

```
0
-1
```

1086

0

1086

2.6 数据规模与约定

除样例外, $n, q = 40,000, 1 \leq boss_i < i$ 。

对于每次操作, $1 \leq x \leq n, 0 \leq y \leq 10^4, 1 \leq l \leq r \leq n$ 。

设①每个人至多有1个直接的下属; ② $l = 1, r = n$; ③ $boss_i$ 是随机生成的。

测试点1满足①②, 测试点2满足②③;

测试点3满足①, 测试点4满足②, 测试点5,6满足③。

3 打字机

3.1 题目描述

小P和小Z有一种神奇的打字机, 说它神奇是因为它支持可持久化。这种打字机是这样工作的:

打字机时刻维护着一个字符串, 初始为空串。一共有 q 次操作, 第 i 次操作可以是:

- 1 c , 其中 c 是一个小写字母。接到这个命令后, 打字机会在当前字符串后添加这个字符 c 。
- 2 x , 其中 x 是小于 i 的自然数 (可以为0)。接到这个命令后, 打字机会把当前字符串变成进行 x 次操作后那时的字符串。

例如, 考虑这个操作序列: $q = 3$, 操作为: 1 $a, 2\ 0, 1\ b$, 那么产生的3个字符串就分别为 a , 空串, b 。

现在我们有两台这样的打字机，称为 A 和 B 。我们用 A_i 表示打字机 A 产生的第 i 个字符串， B_j 表示打字机 B 产生的第 j 个字符串。如果 A_i 在 B_j 中出现了 k 次，那么我们获得 $i \times j \times k$ 的得分。

求总得分对1000000007取模的结果。

注意：对于两个字符串 s, t ，如果 s 是空串，则我们认为 s 在 t 中出现的次数为 t 的长度+1。

3.2 输入格式

输入文件为typewriter.in。

第一行有一个整数 q ，表示打字机 A 的操作次数。

接下来 q 行，每行格式为1 c 或2 x ，表示第一个打字机的每一次操作。

接下来用相同的方式描述了打字机 B 的操作。

3.3 输出格式

输出文件为typewriter.out。

输出一个整数，即总得分对1000000007取模的结果。

3.4 样例输入

```
3
1 a
2 0
1 b
2
1 a
1 b
```

3.5 样例输出

25

3.6 样例解释

$$1 \times 1 \times 1 + 1 \times 2 \times 1 + 2 \times 1 \times 2 + 2 \times 2 \times 3 + 3 \times 1 \times 0 + 3 \times 2 \times 1 = 25。$$

3.7 数据规模与约定

令 n 为两个打字机的操作序列长度的较大值：

对于20%的数据， $n \leq 100$ 。

对于40%的数据， $n \leq 500$ 。

对于70%的数据， $n \leq 3000$ 。

对于100%的数据， $n \leq 300000$ 。

均匀分布着一半的数据满足所有出现的小写字母均为a。

所有数据都是用某种方法随机生成的。