ESPC 模拟赛

第一试

时间：1926 年 8 月 17 日 08:00 ~ 12:30

题目名称	回忆录	喵了个喵 IV	分组作业	欺诈游戏
题目类型	传统型	传统型	传统型	传统型
目录	`memoir`	`meow`	`teamwork`	`prank`
可执行文件名	`memoir`	`meow`	`teamwork`	`prank`
输入文件名	`memoir.in`	`meow.in`	`teamwork.in`	`prank.in`
输出文件名	`memoir.out`	`meow.out`	`teamwork.out`	`prank.out`
每个测试点时限	1.0 秒	3.0 秒	1.0 秒	2.0 秒
内存限制	512 MiB	512 MiB	512 MiB	512 MiB
测试点数目	5	4	6	7
测试点是否等分	否	否	否	否

提交源程序文件名

对于 C++ 语言	`memoir.cpp`	`meow.cpp`	`teamwork.cpp`	`prank.cpp`

编译选项

对于 C++ 语言	`-O2 -std=c++14`

【选手须知】

1. 下发文件可在 [数据删除] 下载。
2. 下发文件的密码为：`HelloLGM`。
3. 数据范围在题面最后，请注意数据范围和题目中加粗内容。
4. 可以写点部分分。
5. 评测机配置为：11th Gen Intel(R) Core(TM) i7-1165G7 @ 2.80GHz。
6. AK 后请不要大声喧哗。
7. 本场比赛所有题目均开启子任务捆绑测试。

回忆录（memoir）

【题目背景】

我常常想，我想不出个答案。
我常常想，我想不出个答案。
再后来，我遇到了你。
我们是不同世纪的人。但是，相同的信念和追求，是可以跨越时间产生共鸣的。

【题目描述】

回忆中，Kevin 依稀记得，有一个神秘的物体，它的质量是 n 克。而现在，为了确定摆在面前的是不是那个回忆录中的物体，Kevin 需要比对这个物体的质量是否正确。

不幸的是，他手头没有什么东西，只有床边的一架天平和一些砝码。每个砝码的质量都是 4 的幂，且每种质量的砝码有**无限个**。你需要帮助 Kevin 求出，在使用的砝码**个数最少**的情况下，有多少种放物体和砝码的方式，对 10^9 取模。

Kevin 在物理课上学到了左物右码的规定，所以物体必须放在天平左盘。两种放砝码的方式不同，当且仅当存在一个砝码质量，使得在左盘和右盘中至少一个，两种方案下放置的该质量砝码个数不同。

【输入格式】

从文件 ***memoir.in*** 中读入数据。
输入的第一行包含一个正整数 n，满足 $1 \leq n \leq 10^{1000}$。

【输出格式】

输出到文件 ***memoir.out*** 中。
输出一行一个正整数，表示放物体和砝码的方案数，对 10^9 取模。

【样例 1 输入】

```
166
```

【样例 1 输出】

```
3
```

【样例 1 解释】

最少需要使用的砝码数是 7。三种方案分别是：

- 左盘放物体，右盘放质量为 $64, 64, 16, 16, 4, 1, 1$ 的砝码。
- 左盘放物体和质量为 $64, 16, 16$ 的砝码，右盘放质量为 $256, 4, 1, 1$ 的砝码。
- 左盘放物体和质量为 $64, 16, 4, 4, 1, 1$ 的砝码，右盘放质量为 256 的砝码。

【样例 2】

见选手目录下的 ***memoir/memoir2.in*** 与 ***memoir/memoir2.ans***。

【样例 2 解释】

该样例满足子任务 $3, 5$ 的限制。

【样例 3】

见选手目录下的 ***memoir/memoir3.in*** 与 ***memoir/memoir3.ans***。

【样例 3 解释】

该样例满足子任务 $3, 4, 5$ 的限制。

【样例 4】

见选手目录下的 ***memoir/memoir4.in*** 与 ***memoir/memoir4.ans***。

【样例 4 解释】

该样例满足子任务 5 的限制。

【子任务】

对于 100% 的数据，$1 \leq n \leq 10^{1000}$。
子任务 1，共 15 分：存在正整数 k，使得 $n = 4^k$。
子任务 2，共 20 分：存在正整数 k，使得 $n = 2^k$。
子任务 3，共 15 分：$1 \leq n \leq 10^5$。
子任务 4，共 25 分：存在正整数 k_1, k_2，使得 $n = 2^{k_1} + 2^{k_2}$。
子任务 5，共 25 分：无附加限制。

喵了个喵 IV（meow）

【题目背景】

搬题人因为构造写错在喵了个喵中挂分了，于是有了这个题目名字。

【题目描述】

喵了个喵的游戏规则很简单，但是这道题并不想让你研究怎么玩这个游戏，你的任务是记录玩家进行游戏的过程。

在游戏中有三个栈，每次你可以选择一个栈进行操作。游戏的状态总共有 n 个，对于状态 1 到状态 $(n-1)$，每个状态都有一个小写字母参数 c_i 和三个后继状态 a_{i1}, a_{i2}, a_{i3}。a_{ix} 表示在状态 i 选择了第 x 个栈会得到的新状态。第 n 个状态是整个游戏的结束界面，所以它不存在字母参数和后继状态。**保证一个状态的后继状态编号一定大于它自身的编号。**

小 E 是一个善良的人，为了方便玩家计算，他希望整个游戏的状态数尽量少，但是他并不想修改代码，于是他找到了你。你需要尽量的简化整个游戏的状态表示。具体地，定义一个**合法的操作序列**为一个整数序列 $(x_1, x_2, \cdots, x_k)$，满足从状态 1 开始进行 k 次操作，第 i 次选择第 x_i 个栈，最终会停留在状态 n。对于一个合法的操作序列，定义他的**参数串**为在进行操作的过程中除了结束状态的所有其余状态的字符参数按顺序拼接得到的字符串。

对于你构造的新状态表示法，设原状态的合法操作序列集合为 S，你的状态的合法操作序列集合为 T。你需要保证 $S = T$，且对于任意 S 中的操作序列，按照它进行游戏在原状态表示和新状态表示中得到的参数串相同。你需要保证新表示法的状态个数尽量少，只需输出这个状态个数即可。

【输入格式】

从文件 ***meow.in*** 中读入数据。

输入第一行包含一个正整数 n，表示游戏的状态个数。

接下来 $n-1$ 行，第 $(i+1)$ 行表示状态 i 的信息。每行首先输入一个字符 c_i，表示状态 i 的参数，接下来三个正整数 a_{i1}, a_{i2}, a_{i3}，表示状态 i 的后继状态。

【输出格式】

输出到文件 ***meow.out*** 中。

一行一个正整数 k，表示你构造的新状态表示法的状态个数。

【样例 1 输入】

```
11
N 3 5 2
Z 4 5 6
N 7 11 9
N 8 11 10
C 11 9 9
Z 11 9 10
C 11 11 11
C 11 11 11
Z 11 11 11
Z 11 11 11
```

【样例 1 输出】

```
8
```

【子任务】

对于 100% 的数据，$1 \le n \le 5 \times 10^5$，$i < a_{i1}, a_{i2}, a_{i3} \le n$。

子任务 1，共 10 分：对于所有 $1 \le i < n$，有 $a_{i1} = i + 1$。

子任务 2，共 25 分：$1 \le n \le 10$。

子任务 3，共 30 分：$1 \le n \le 2000$。

子任务 4，共 35 分：无附加限制。

分组作业（teamwork）

【题目背景】

打老年人，那赢了不是胜之不武吗。

【题目描述】

在伟大的翟家军，有 n 名 PION 参赛选手，为了让他们放平心态，他们组织了一场足球赛。球赛分为两队，**人数可以不同，也可以有一队根本没有人**。

但是不幸的是，这 n 名选手两两之间可能有**双向**敌对关系，不过由于翟家军的选手们都很善良，所以每个人**最多与三个人**有敌对关系。

你需要将这 n 名选手分成两队，并保证每个人最多和**一个**队友有敌对关系，或者判断无解。

【输入格式】

从文件 ***teamwork.in*** 中读入数据。

输入的第一行包含一个整数 n，表示选手个数。

接下来 n 行，第 $(i+1)$ 行第一个正整数 c_i 表示 i 的敌对关系个数，接下来 c_i 个数表示与 i 敌对的人。选手从 1 开始编号。

【输出格式】

输出到文件 ***teamwork.out*** 中。

第一行输出一个整数 k，表示人数较少那队的人数。若人数相等，输出 1 所在队伍的人数。

第二行输出 k 个整数，表示那队的成员，由小到大输出他们的编号。

如果无解，输出一行一个字符串 `NO SOLUTION`。

【样例 1 输入】

```
8
3 2 3 7
3 1 3 7
3 1 2 7
1 6
0
2 4 8
3 1 2 3
```

```
1 6
```

【样例 1 输出】

```
4
1 2 5 6
```

【样例 2】

见选手目录下的 ***teamwork/teamwork2.in*** 与 ***teamwork/teamwork2.ans***。

【样例 2 解释】

该样例满足子任务 5, 6 的限制。

【子任务】

对于 100% 的数据，$1 \le n \le 7000$。

子任务 1，共 10 分：$1 \le n \le 22$。

子任务 2，共 15 分：$1 \le n \le 50$。

子任务 3，共 20 分：$1 \le n \le 700$。

子任务 4，共 5 分：敌对关系构成的无向图是一张二分图。

子任务 5，共 15 分：保证对于任意 $1 \le i < n$，i 和 $i+1$ 号选手相互敌对，1 和 n 号选手相互敌对。

子任务 6，共 35 分：无附加限制。

欺诈游戏（prank）

【题目背景】

第一届粉兔杯的半决赛已经结束，进入决赛的两名选手为 jiangly 和没进省队的小丑。最终的冠军将花落谁家？点击 [数据删除]，为你支持的选手投上一票吧！

【题目描述】

有 n 个选手参加了第 154 届粉兔杯，他们会进行一次单循环赛。第 i 个人的能力值是 a_i，为了简便起见，我们假设能力值大的选手一定会战胜能力值小的选手。**保证 a_i 两两不同**。

但是按照这种假定 jiangly 就必然登顶了，为了不给他送游戏，帮助他戒网瘾，主办方 EA 决定偷偷修改一定的比赛结果。具体地，EA 会进行 m 次操作，每次操作有两个参数 $[l_i, r_i]$，表示对于两个人能力值都在范围 $[l_i, r_i]$ 中的比赛，EA 偷偷翻转这场比赛的结果。

观众们都不想看到碾压局，更希望看到不强的选手战胜较强的选手。定义「精彩值」为满足 p 赢 q，q 赢 r，r 赢 p 的数对 (p, q, r) 个数。两个数对不同当且仅当**它们包含的整数不同**，即 $(1, 2, 3)$ 和 $(2, 3, 1)$ 只计算一次。你需要求出 m 次操作**全部结束**后的「精彩值」。

【输入格式】

从文件 ***prank.in*** 中读入数据。

输入的第一行包含两个正整数 n, m，表示参赛人数和操作个数。

第二行 n 个正整数，第 i 个数 a_i 表示第 i 个人的能力值。

第 3 至 $(m+2)$ 行，第 $(j+2)$ 行两个整数 l_j, r_j，表示一次操作的参数。

【输出格式】

输出到文件 ***prank.out*** 中。

输出一行一个整数，表示「精彩值」。

【样例 1 输入】

```
3 2
1 2 3
1 2
2 3
```

【样例 1 输出】

```
1
```

【样例 1 解释】

初始时 3 赢 1，3 赢 2，2 赢 1。第一次操作翻转 1, 2 的比赛结果，第二次操作翻转 2, 3 的比赛结果，此时 (1, 2, 3) 是一个满足条件的数对。故「精彩值」为 1。

【样例 2 输入】

```
5 3
5 9 4 1 7
1 7
2 8
3 9
```

【样例 2 输出】

```
3
```

【样例 3】

见选手目录下的 ***prank/prank3.in*** 与 ***prank/prank3.ans***。

【样例 3 解释】

该样例满足子任务 1, 2, 3, 4, 5, 7 的限制。

【样例 4】

见选手目录下的 ***prank/prank4.in*** 与 ***prank/prank4.ans***。

【样例 4 解释】

该样例满足子任务 7 的限制。

【子任务】

对于 100% 的数据，$1 \le n, m \le 10^5$，$1 \le a_i, l_i, r_i \le 10^9$。保证 a_i 互不相同。

子任务 1，共 10 分：$1 \le n, m \le 500$。

子任务 2，共 10 分：$1 \le n \le 500$，$1 \le m \le 50000$。

子任务 3，共 15 分：$1 \le n, m \le 5000$。

子任务 4，共 10 分：$1 \le n \le 5000$。

子任务 5，共 15 分：$1 \le m \le 5000$。

子任务 6，共 10 分：$m = 1$。

子任务 7，共 30 分：无附加限制。