

2016 元旦欢乐赛

Gromah

题目名称	Fibnacci	Monster	Star
可执行文件名	fibnacci	monster	star
输入文件名	fibnacci.in	monster.in	star.in
输出文件名	fibnacci.out	monster.out	star.out
单个测试点时限	2 s	1 s	1 s
内存限制	512 MB	512 MB	512 MB
测试点个数	10	10	10
单个测试点分数	10	10	10
是否有部分分	否	否	否
题目类型	传统	传统	传统
提交文件须加后缀:			
对于 C++ 语言	fibnacci.cpp	monster.cpp	star.cpp
对于 C 语言	fibnacci.c	monster.c	star.c
对于 Pascal 语言	fibnacci.pas	monster.pas	star.pas

注意事项

- 本套题考试时间为 8:00 开始到 12:00 结束，持续 4 个小时，请大家注意好考试时间。
- 数据在 Windows 下生成，评测在 Windows 7 的 Lemon 下进行，请大家注意相关细节。



1 Fibonacci

1.1 题目描述

从前，有一个长为 n 的序列并给定所有元素的初始值，以及 m 个操作，操作分为两种：

- 形如 “1 l r” 的操作，意为给 $[l, r]$ 中的每一个元素 A_i 都加上 F_{i-l+1} ，其中 F_x 表示第 x 个斐波那契数，规定 $F_0 = 0, F_1 = 1$ 。
- 形如 “2 l r” 的操作，意为查询序列在 $[l, r]$ 内的元素和并输出其对 $10^9 + 1$ 取模的结果。

1.2 输入格式

输入第一行两个正整数 n, m ，意义如题所述。

第二行 n 个非负整数，表示序列所有的初始值。

接下来 m 行，每行三个正整数 op, l, r ，表示一次操作，保证 $l \leq r$ 。

1.3 输出格式

输出若干行，每行一个非负整数，表示每次询问操作的答案对 $10^9 + 1$ 取模的结果。

1.4 样例输入输出

fibnacci.in	fibnacci.out
5 2 0 0 0 0 0 1 1 4 2 2 5	6

1.5 样例解释

序列最后是长这样的：1 1 2 3 0

所以在 $[2, 5]$ 内的元素和就是 $1 + 2 + 3 = 6$ 了。

1.6 数据范围及约定

测试点编号	n	m	特殊性质
0,1	≤ 1000	≤ 1000	无
2,3	≤ 40000	≤ 40000	无
4,5	$\leq 10^5$	$\leq 10^5$	$r - l$ 为定值
6,7,8,9	$\leq 10^5$	$\leq 10^5$	无

2 Monster

2.1 题目描述

从前，有 2^n 个点，标号从 0 到 $2^n - 1$ 。定义两个数 u, v 的距离 $dis(u, v)$ 为 u 和 v 按位异或的结果的二进制表示中数码 1 的个数，比如说：

$$dis(1, 2) = 2, dis(4, 5) = 1, dis(7, 7) = 0$$

那么在这个图中，两个点 u, v 之间有边相连当且仅当 $dis(u, v) = 1$ 。

在一开始，图中的每个点上都会有一些“大年兽”。由于“大年兽”擅长“魔法”，所以每过一秒，每只“大年兽”就会“魔法”一下。我们令：在 u 号点的一只“大年兽”“魔法”一下的效果是：所有与 u 号点有边相连的点都会增加一只新的“大年兽”。

现在给出这个图中每个点的“大年兽”的数量，问过了 t 秒后，每个点上会有多少只“大年兽”，输出对 593119681 取模的结果。

2.2 输入格式

输入第一行两个正整数 n, t ，意义如题所述。

第二行 2^n 个非负整数，第 i 个数表示标号为 $i - 1$ 的点上的“大年兽”的初始数量。

2.3 输出格式

输出仅一行 2^n 个非负整数，第 i 个数表示过了 t 秒后，标号为 $i - 1$ 的点上的“大年兽”的数量对 593119681 取模的结果。

2.4 样例输入输出

monster.in	monster.out
2 2 1 0 0 0	3 2 2 2

2.5 样例解释

第一秒后，每个点上的“大年兽”的数量分别为：1 1 1 0

第二秒后，每个点上的“大年兽”的数量分别为：3 2 2 2

2.6 数据范围及约定

测试点编号	n	t	特殊性质
0,1	≤ 6	≤ 1000	无
2,3	≤ 6	$\leq 10^7$	无
4,5	≤ 10	≤ 1000	无
6,7,8,9	≤ 18	$\leq 10^9$	无

3 Star

3.1 题目描述

从前，有一片夜空，夜空下有 n 颗流星，每颗流星都会沿着一条直线运动，保证流星的轨迹两两不重合，并且一颗流星所划过的轨迹上的所有点的亮度都会增加一点，问时间足够久之后，夜空中亮度最大的点的亮度为多少，以及有多少个这样的点，保证这样的点有限。

注意：我们考虑把夜空抽象成一个平面直角坐标系，那么流星的运动轨迹就会是一个一次函数，故给出的流星的轨迹就会是 $Ax + By + C = 0$ 这样的形式。

3.2 输入格式

输入第一行仅一个正整数 n ，表示流星的数目。

接下来 n 行，每行三个整数 A_i, B_i, C_i ，表示第 i 颗流星的轨迹方程。

3.3 输出格式

输出仅一行两个整数，分别为最大亮度，以及亮度最大的点的个数。

3.4 样例输入输出

star.in	star.out
3 0 1 0 1 0 0 1 1 -1	2 3

3.5 样例解释

最大亮度为 2，有 $(0,0), (1,0), (0,1)$ 这三个点的亮度为 2。

3.6 数据范围及约定

测试点编号	n	$ A_i , B_i $	$ C_i $
0,1	≤ 10	≤ 100	$\leq 10^6$
2,3	≤ 70	≤ 1000	$\leq 10^8$
4,5	≤ 200	≤ 1000	$\leq 10^8$
6,7,8,9	≤ 1000	≤ 1000	$\leq 10^8$