

Problem A. A

Time limit 1000 ms
Mem limit 524288 kB

题目描述

给定一个包含 n 个自然数的数组 $A = [a_1, a_2, \dots, a_n]$ 。

需要将数组元素划分为两种颜色，使得不存在两个同色的元素 x 和 y 满足 x 可以整除 y 且 $\frac{x}{y} = p$ ，其中 p 是一个质数。保证这样的划分是存在的。

输入格式

第一行包含一个整数 n ($1 \leq n \leq 10^5$)，表示数组的元素数量。

第二行包含 n 个整数 a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^6$)，表示数组的元素。

输出格式

输出数组划分为两种颜色的描述。

输出 n 个整数，如果第 i 个元素 a_i 被划分为第一种颜色，则输出 1；如果被划分为第二种颜色，则输出 2。

如果存在多种合适的划分方案，可以输出任意一种。

样例 1

Input	Output
4 1 2 3 4	2 1 1 2

在第一个样例中，第一种颜色的元素有 2 和 3，第二种颜色的元素有 1 和 4。第一种颜色的元素之间不能整除。4 可以整除 1，但它们的比值不是质数。

样例 2

Input	Output
1 20	1

数据范围与提示

详细子任务附加限制及分值如下表所示。

子任务	分值	附加限制	子任务依赖
1	9	所有 $a_i \leq 2$	
2	19	保证所有 a_i 是某个质数 p 的幂	
3	12	所有 $a_i \leq 3$	1
4	13	所有 $a_i \leq 4$	1, 3
5	21	$n \leq 10$	
6	26	无附加限制	1 ~ 5

Problem B. B

Time limit 1000 ms

Mem limit 524288 kB

题目描述

你已抵达维尔纽斯，并希望游览立陶宛的一些城市。

立陶宛的城市位于一条直线上，从 1 到 N 依次编号。维尔纽斯的编号为 1。

每个城市都有一个火车站，从该火车站出发的列车只有一条线路。你只能在线路的起点上车，但可以在任何站点下车。以第 i 个城市为起点的列车每隔 d_i 个城市停靠一次，其线路包括 x_i 个站点（不包括起点城市）。如果 $d_i = 0$ ，则从第 i 个城市出发的列车当前停运，因此你无法上车。

更准确地说，如果你在第 i 个城市上车，你可以在编号为 $i + t \cdot d_i$ 的任何城市下车，其中 $1 \leq t \leq x_i$ 。请注意，由于你只想游览立陶宛的城市，因此即使列车路线上有更多的站点，你也不会乘坐列车到达超过第 N 个城市的城市。

你想参观一些城市，所以你会乘坐火车往返于这些城市之间。在计划行程时，你想知道有多少种从维尔纽斯出发的不同的旅程。如果两段旅程按不同的城市序列游览，那么这两段旅程不同。

计算这个值，输出这个值对 $10^9 + 7$ 取模后的结果。

输入格式

第一行一个整数 N ，表示城市个数。

接下来 N 行，第 i 行包含两个整数 d_i 和 x_i ，表示从城市 i 出发的线路。

输出格式

输出一个整数，表示你游览 N 个城市中的一些可以采用的旅程数模 $10^9 + 7$ 的值。

样例

Input	Output
5 1 3 2 1 1 3 0 10 3 5	7

你可以进行 7 种可能的旅程：

- 1
- 1 → 2
- 1 → 2 → 4
- 1 → 3
- 1 → 3 → 4
- 1 → 3 → 5
- 1 → 4

数据范围与提示

对于所有数据，满足：

- $1 \leq N \leq 10^5$
- $0 \leq d_i, x_i \leq 10^9$ (对于所有 $1 \leq i \leq N$)

详细子任务附加限制及分值如下表所示。

子任务编号	附加限制	分值
1	$N \leq 15$	8
2	$N \leq 10^4$	13
3	对于所有列车， $d_i = 1$	16
4	对于所有列车， $x_i = 10^9$	34
5	无附加限制	29

Problem C. C

Time limit 1000 ms

Mem limit 524288 kB

题目描述

你拥有一家通过为客户完成工作来赚钱的企业。目前，你可以从 N 个一次性工作中进行选择，编号从 1 到 N 。

完成第 i 个工作会带来 x_i 欧元的利润。利润也可能为负数 ($x_i < 0$)。

有些工作依赖于另一个工作。也就是说，可能存在一个编号为 p_i 的工作必须先完成，然后才能开始第 i 个工作。因此，一个利润很高的工作如果它依赖于一个利润为负的工作，就可能比看起来的吸引力小。如果 $p_i = 0$ ，第 i 个工作没有任何依赖关系。

你目前拥有 s 欧元。只要满足依赖关系，你可以决定完成哪些工作以及按什么顺序完成它们。此外，你拥有的钱数在任何时候都不能变为负数。

请计算通过选择完成部分（可能没有） N 个工作并按选定的顺序完成，可以获得的最大利润。

输入格式

第一行两个整数 N 和 s ，分别表示工作数和你最初拥有的钱数。

接下来 N 行，第 i 行包含两个整数 x_i 和 p_i ，分别表示利润和完成工作 i 之前需要完成的工作编号。如果 $p_i = 0$ ，则表示工作 i 没有工作依赖。

输出格式

输出一个整数，表示你能获得的最大利润。

样例

Input	Output
6 1 3 0 -3 1 -5 0 2 1 6 3 -4 5	6

为了最大化利润，你应该选择工作 1, 4, 3 和 5，并按如下顺序完成：

- 工作 1：钱 1 \rightarrow 4
- 工作 4（前置工作 1 已完成）：钱 4 \rightarrow 6
- 工作 3：钱 6 \rightarrow 1
- 工作 5（前置工作 3 已完成）：钱 1 \rightarrow 7

综上，总利润为 $7 - 1 = 6$ （目前的钱数减去开始的钱数）。

数据范围与提示

对于所有数据，满足：

- $1 \leq N \leq 3 \cdot 10^5$
- $0 \leq s \leq 10^{18}$
- $-10^9 \leq x_i \leq 10^9$ （对于所有 $1 \leq i \leq N$ ）
- $0 \leq p_i < i$ （对于所有 $1 \leq i \leq N$ ）

详细子任务附加限制及分值如下表所示。

子任务编号	附加限制	分值
1	$s = 10^{18}$	11
2	$N \leq 2000$ 且对于所有工作，要么 $p_i = 0$ ，要么 $p_i = i - 1$	14
3	对于所有工作，要么 $p_i = 0$ ，要么 $p_i = i - 1$	15
4	$N \leq 2000$	29
5	无附加限制	31

Problem D. D

Time limit 1000 ms

Mem limit 524288 kB

题目描述

你觉得欺骗你最好的朋友会很有趣，你把他放在一个无限网格的 $(0, 0)$ 单元格上。然后你的朋友会在这个网格上无限地移动，每次移动一步，并且总是移动到相邻的四个单元格之一。

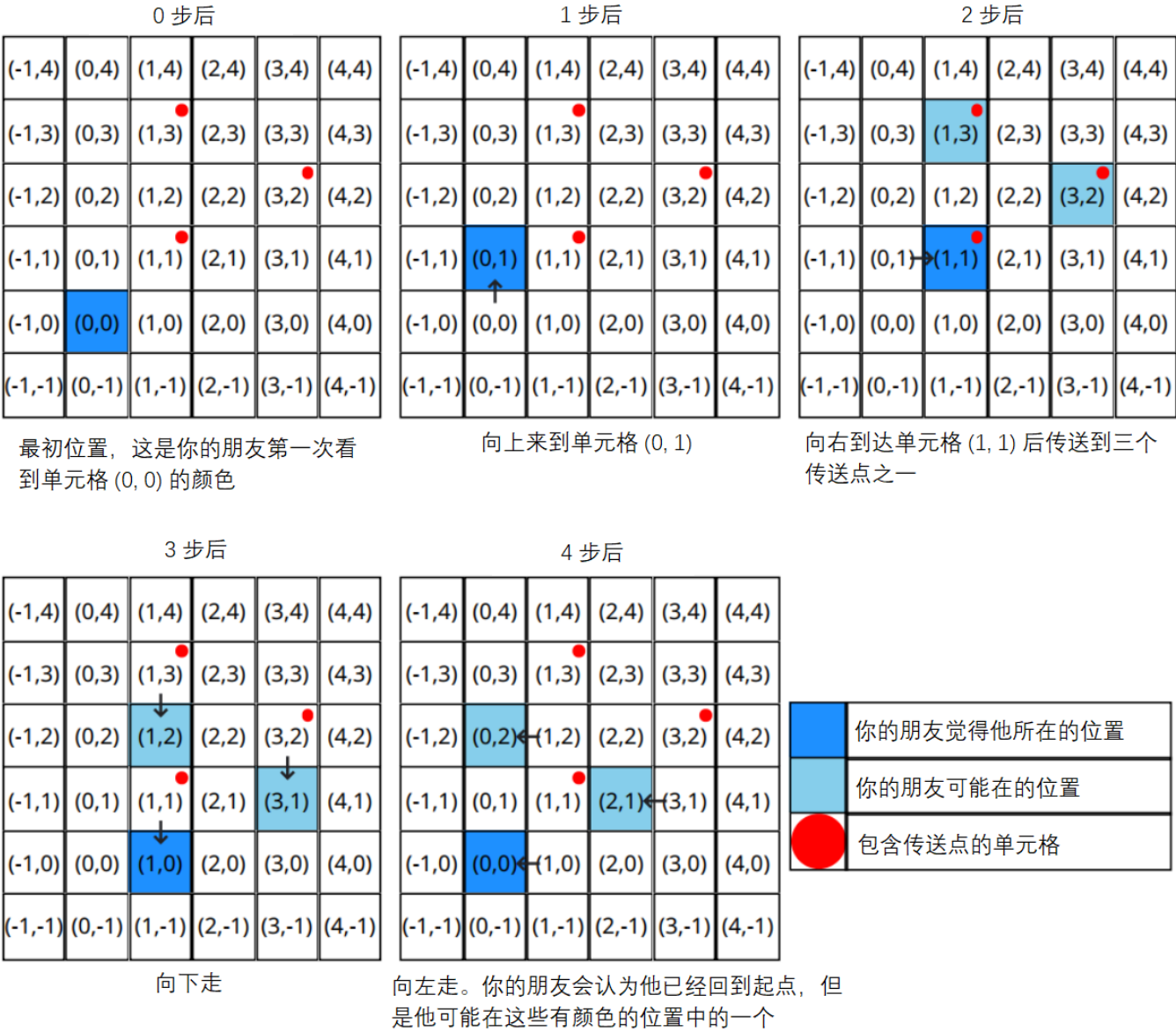
网格上有 N 个单元格包含一个传送门。一旦你的朋友踩到一个传送点，他就会瞬间传送到一个随机的传送点处（可能是他刚刚踩到的那个，也可能是不同的一个）。如果 $(0, 0)$ 单元格也有一个传送门，你的朋友一开始被放置在网格上时也会被传送。

作为这个恶作剧的一部分，你想让你的朋友完全不会发现有传送点的存在。你的朋友只能看到他当前所在单元格的顏色，所以你应该确保从你朋友的视角来看，单元格的顏色永远不会变化。特别是，如果你的朋友认为他已经进入过某个单元格（例如先向左移动，然后立即向右移动），他应该看到与第一次进入时相同的顏色。

注意：当你的朋友踩到传送点时，他会看到传送点所在单元格的顏色，以及他被传送到的单元格的顏色。因此，你需要将所有包含传送门的单元格染成相同的顏色，以避免传送时暴露。

一个简单的解决方案是将所有单元格染成相同的顏色。但是顏色很好！所以你想尽可能使用更多的顏色。

让我们考虑一个例子，其中传送门位于 $(1, 1)$, $(1, 3)$ 和 $(3, 2)$ 单元格，你的朋友进行了以下一系列移动：向上、向右、向下、向左。



在一系列的移动之后，朋友会认为他又回到了起点单元格 (0, 0)，但实际上他也有可能到了 (0, 2) 或 (2, 1)。他在开始时已经看到了单元格 (0, 0) 的颜色，所以如果他现在看到了不同的颜色，他就会意识到这里肯定有传送门。我们不希望发生这种情况，所以我们必须为这三个单元格选择相同的颜色。

没有任何一连串的移动会让你的朋友认为他最终到达了单元格 (0, 0)，而实际上他最终到达了单元格 (1, 0)，因此这些单元格可以安全地涂上不同的颜色。

下面是上面例子中 4 种颜色的着色。本例不可能使用超过 4 种颜色。

$(-1,4)$	$(0,4)$	$(1,4)$	$(2,4)$	$(3,4)$	$(4,4)$
$(-1,3)$	$(0,3)$	$(1,3)$	$(2,3)$	$(3,3)$	$(4,3)$
$(-1,2)$	$(0,2)$	$(1,2)$	$(2,2)$	$(3,2)$	$(4,2)$
$(-1,1)$	$(0,1)$	$(1,1)$	$(2,1)$	$(3,1)$	$(4,1)$
$(-1,0)$	$(0,0)$	$(1,0)$	$(2,0)$	$(3,0)$	$(4,0)$
$(-1,-1)$	$(0,-1)$	$(1,-1)$	$(2,-1)$	$(3,-1)$	$(4,-1)$

我们再来看一个不同的例子，在单元格 $(0,0)$, $(0,1)$, $(1,0)$, $(0,-1)$ 和 $(-1,0)$ 处都有传送点。假设你的朋友试图通过向右走一步然后向上走三步到达单元格 $(1,3)$ 。一种可能是，如果他在开始时和每一步之后都被传送到了单元格 $(0,0)$ 。如果你的朋友现在通过向下走三步，再向左走一步返回到他认为的单元格 $(0,0)$ ，并且在此过程中没有被传送离开他所到单元格，那么他最终会到达 $(-1,-3)$ 。而你的朋友会认为他们第二次来到了单元格 $(0,0)$ ，并希望看到相同的颜色。因此，你需要给 $(-1,-3)$ 和 $(0,0)$ 涂上相同的颜色。

请注意，我们最初选择的单元格 $(1,3)$ 并没有什么特别之处。同样，你也可以证明其他单元格必须与 $(0,0)$ 颜色相同。

在确保你的朋友不会发现有传送点存在的前提下，计算出你可以使用的颜色的最大数量。

输入格式

第一行包含一个整数 N ，表示传送点个数。

接下来 N 行每行两个整数。第 i 行包含 x_i 和 y_i ，表示单元格 (x_i, y_i) 处有一个传送点。

输出格式

输出一个整数，表示在你的朋友不会发现有传送点存在的情况下可以使用的最大颜色数，如果可以使用无限多种颜色，则输出 -1 。

样例 1

Input	Output
3 1 1 1 3 3 2	4

这是题目描述中讨论的第一个例子。

样例 2

Input	Output
5 0 0 1 0 -1 0 0 1 0 -1	1

这是题目描述中讨论的第二个例子。

样例 3

Input	Output
1 1 -1	-1

你的朋友只能被「传送」到传送点所在的同一个单元格，因此即使每个单元格的顏色都互不相同，他也不会注意到传送门的存在。

数据范围与提示

对于所有数据，满足：

- $1 \leq N \leq 10^5$
- $-10^6 \leq x_i, y_i \leq 10^6$ (对于所有 $1 \leq i \leq N$)

- 没有两个传送点位于同一单元格

详细子任务附加限制及分值如下表所示。

子任务编号	附加限制	分值
1	$N \leq 2$	1
2	$N \leq 3$	10
3	对于所有整数 x_1, x_2, y_1, y_2 : 如果 (x_1, y_1) 和 (x_2, y_2) 处有传送点, 那么 (x_1, y_2) 处也有传送点	10
4	$N \leq 100$ 且 $-100 \leq x_i, y_i \leq 100$ (对于 $1 \leq i \leq N$)	29
5	$N \leq 2000$	15
6	无附加限制	35