NOI2025 广东省队集训

GDOI

时间：2025 年 4 月 xx 日

题目名称	序列变换	追忆	小方的疑惑
题目类型	传统题	传统题	传统题
目录	`trans`	`recall`	`square`
可执行文件名	`trans`	`recall`	`square`
输入文件名	`trans.in`	`recall.in`	`square.in`
输出文件名	`trans.out`	`recall.out`	`square.out`
每个测试点时限	1 秒	4 秒	2 秒
内存限制	1024 MB	1024 MB	1024 MB
子任务数目	10	20	25
测试点是否等分	是	是	是

提交源程序文件名

对于 C++ 语言	`trans.cpp`	`recall.cpp`	`square.cpp`

编译选项

对于 C++ 语言	`-O2 -std=c++14`

注意事项（请仔细阅读）

1. 测试机器：CPU(AMD Ryzen 5 3600 6-Core Processor *12), RAM 8.0G。
2. 系统环境：NOI Linux 2.0(基于 Ubuntu 20.04.1)。
3. 文件名（程序名和输入输出文件名）必须使用英文小写。
4. C/C++ 中函数 `main()` 返回类型必须是 `int`，程序正常结束返回值必须是 0。
5. 若无特殊说明，结果的比较方式为全文比较（过滤行末空格及文末回车）。
6. 选手提交的程序源文件必须不大于 100KB。
7. 程序可使用的栈空间内存限制与题目的内存限制一致。

序列变换（trans）

【题目描述】

给定一个长度为 n 的整数序列 a（下标从 1 开始），你可以进行以下操作若干次：

- 选择一个区间 $[l,r]$，对于所有 $i \in N, x \in \{0,1\}, l+3i+x \leq r$，将 a_{l+3i+x} 增加 $(-1)^x$。

问至少需要多少次操作，才能使 a 序列的所有位置值均为 0，若无解，则输出 -1。

【输入格式】

从文件 ***trans.in*** 中读入数据。
第一行两个正整数 T, n，表示数据组数以及每组数据的序列长度。
接下来 T 行，每行一个长度为 n 的整数序列 a。

【输出格式】

输出到文件 ***trans.out*** 中。
T 行，每行一个数，表示对应的数据的答案。

【样例 1 输入】

```
4 5
-1 0 1 -1 1
-1 -2 -3 -4 5
-11 -45 14 -1919 810
-1 -2 3 4 -5
```

【样例 1 输出】

```
2
11
1975
-1
```

【样例 1 解释】

对于第一组数据，进行操作 $[1,5], [2,3]$ 即可。

【样例 2】

见选手目录下的 ***trans/trans2.in*** 与 ***trans/trans2.ans***。

【样例 3】

见选手目录下的 ***trans/trans3.in*** 与 ***trans/trans3.ans***。

【样例 4】

见选手目录下的 ***trans/trans4.in*** 与 ***trans/trans4.ans***。

【数据范围】

对于所有测试数据，保证 $T = 30$。

测试点编号	$n =$	$\lvert a_i \rvert \leq$
1	5	5
2	20	10
3	70	100
4, 5	200	10^{10}
6, 7	2×10^3	2×10^3
8, 9, 10	3×10^4	10^{10}

追忆（recall）

【题目描述】

小 L 出了一个和 DAG 有关的问题并打算投给联合省选。虽然题目很烂，但因为是题，所以被纳入了被选题当中。小 L 因此很不重视，决定脚造数据。

然而，联合省选前一天，DAY1 T2 被爆破了，于是小 L 的题就变成了 DAY1 T2，并且因为小 L 太懒，所以没有重造数据。不出意外的话，意外发生了，小 L 的题被大量暴力通过了。

以上内容纯属虚构。

小 L 追忆了一下造数据的过程，发现他生成的 DAG 其实是按照以下方式生成的：

- 对于 $i = 2 \sim n$，a_i, b_i 在 $1 \sim i-1$ 中随机生成，然后 i 向 a_i, b_i 连有向边。

小 L 心想，数据既然是随机的，那么这个 DAG 处理起来肯定就很简单，于是给了你 $n-1$ 个询问，第 i $(1 \le i < n)$ 个询问有参数 x_i $(1 \le x_i \le i)$，求 $i+1$ 到 x_i 的最短路，如果 $i+1$ 无法到达 x_i 则输出 -1。

【输入格式】

从文件 ***recall.in*** 中读入数据。

第一行两个正整数 $n, seed$，其中 $seed$ 是用来生成 a_i, b_i 的随机种子。

具体的，a_i, b_i 按照如下程序生成：

```
unsigned shift(unsigned &a)
{
    a^=a<<13;
    a^=a>>7;
    a^=a<<17;
    return a;
}
void init(unsigned seed)
{
    for(int i=2;i<=n;i++)
        a[i]=shift(seed)%(i-1)+1,
        b[i]=shift(seed)%(i-1)+1;
}
```

保证 $seed$ 在 $[1, 2^{32})$ 中随机生成。

第二行 $n-1$ 个正整数 $x_1, x_2, \cdots, x_{n-1}$。

【输出格式】

输出到文件 ***recall.out*** 中。

记第 i 次询问的答案为 s_i，由于输出量过大，你只需要输出 $\oplus_{i=1}^{n-1}(s_i+2)\times i$ 即可，其中 $\oplus$ 为异或运算。

【样例 1 输入】

```
10 1
1 1 2 1 2 3 1 2 3 4
```

【样例 1 输出】

```
17
```

【样例 1 解释】

$a_{2\sim10}$ 分别为 $1, 2, 2, 4, 1, 5, 1, 4, 9$。

$b_{2\sim10}$ 分别为 $1, 2, 3, 2, 1, 1, 6, 7, 9$。

$s_{1\sim9}$ 分别为 $1, 2, 1, 2, -1, 3, 1, 2, 3$。

【样例 2】

见选手目录下的 ***recall/recall2.in*** 与 ***recall/recall2.ans***。

满足特殊性质 A

【样例 3】

见选手目录下的 ***recall/recall3.in*** 与 ***recall/recall3.ans***。

满足特殊性质 B

【样例 4】

见选手目录下的 ***recall/recall4.in*** 与 ***recall/recall4.ans***。

【数据范围】

为了方便选手调试，下发文件中的输出文件的内容为 $s_{1\sim n-1}$。

测试点编号	$n=$	特殊性质
1	10	A
2, 3	5×10^4	A
4, 5	10^5	无
6, 7	3×10^5	无
8, 9	5×10^5	A
10, 11	5×10^5	无
12, 13	10^6	B
14, 15	10^6	无
16, 17	2×10^6	B
18, 19, 20	2×10^6	无

特殊性质 A：保证 x_i 在 $1\sim i$ 中随机生成。

特殊性质 B：保证 $x_i\le10^3$。

小方的疑惑（square）

【题目描述】

小方（275307894a）是一名国家集训队队员，他一直有很多疑惑。

这天，小 L 送给他了一个长度为 n 的正整数序列 a 和一个长度为 $n-1$ 的正整数序列 b（下标均从 1 开始），其中序列 b 满足 $1 \le b_i \le i$。

小方可以执行如下操作若干次：

- 选择一个 $i(1 \le i < n)$ 满足 $a_{b_i} > 0$，将 a_{b_i} 减去 1，将 a_{i+1} 增加 1。

小方很疑惑有多少种本质不同的序列 a 可以被生成，可他想了很久却想不到一个优秀的做法，请你帮帮他。

由于答案很大，所以要对 10^9+7 取模。

【输入格式】

从文件 ***square.in*** 中读入数据。

第一行一个正整数 n。

第二行 $n-1$ 个正整数 $b_1, b_2, \cdots, b_{n-1}$。

第三行 n 个非负整数 $a_1, a_2, \cdots, a_n$。

【输出格式】

输出到文件 ***square.out*** 中。

一行一个数，表示最终的答案。

【样例 1 输入】

```
4
1 1 2
1 1 1 0
```

【样例 1 输出】

```
7
```

【样例 1 解释】

可以被生成的序列 a 的为:$[1,1,1,0]$,$[1,0,1,1]$,$[0,1,2,0]$,$[0,0,2,1]$,$[0,2,1,0]$,$[0,1,1,1]$,$[0,0,1,2]$。

【样例 2 输入】

```
6
1 1 2 2 3
1 1 4 5 1 4
```

【样例 2 输出】

```
63
```

【样例 3】

见选手目录下的 ***square/square3.in*** 与 ***square/square3.ans***。

【样例 4】

见选手目录下的 ***square/square4.in*** 与 ***square/square4.ans***。

【样例 5】

见选手目录下的 ***square/square5.in*** 与 ***square/square5.ans***。

【样例 6】

见选手目录下的 ***square/square6.in*** 与 ***square/square6.ans***。

【样例 7】

见选手目录下的 ***square/square7.in*** 与 ***square/square7.ans***。

【数据范围】

对于所有数据，保证 $0 \le a_i \le 10^9, 1 \le b_i \le i$。

测试点编号	$n=$	特殊性质
1	5	A
2, 3	5	无
4, 5	100	无
6, 7	500	A
$8 \sim 10$	500	无
$11 \sim 13$	8×10^3	A
$14 \sim 16$	8×10^3	B
$17 \sim 25$	8×10^3	无

特殊性质 A：对于任意 $1 \leq i \leq n$ 均满足 $a_i = 1$。

特殊性质 B：对于任意 $2 \leq i \leq n$ 均满足 $b_i = \lfloor \frac{i+1}{2} \rfloor$。