

伪伪随机 (prg)

这是一道交互题。

【题目背景】

小 D 发现一些比较简洁的函数生成的字符串具有某种伪随机性质，请你看看他说的是否有道理！

【题目描述】

对于一个 k 元布尔函数 f 以及 m 个取值在 $[0, n - 1]$ 中的整数构成的 k 元组 $x_{i,j}$ ($0 \leq i < m, 0 \leq j < k$)，定义其对应的 n, m 伪随机函数 $\{0, 1\}^n \mapsto \{0, 1\}^m$ 如下：该函数输入 $(y_0, y_1, \dots, y_{n-1}) \in \{0, 1\}^n$ ，输出 $(\hat{y}_0, \hat{y}_1, \dots, \hat{y}_{m-1}) \in \{0, 1\}^m$ ，其中

$$\hat{y}_i = f(y_{x_{i,0}}, y_{x_{i,1}}, \dots, y_{x_{i,k-1}}).$$

考虑如下一类 k 元布尔函数 f ：对于一个长度为 $k - 1$ 的二元布尔函数序列 $[op_0, op_1, \dots, op_{k-2}]$ ，其中 $op_i \in \{0, 1, 2\}$ ($0 \leq i \leq k - 2$)，分别表示二元布尔函数 AND, OR, XOR，定义其对应的 k 元布尔函数如下：

$$(z_0, z_1, \dots, z_{k-1}) \mapsto ((z_0 \text{ } op_0 \text{ } z_1) \text{ } op_1 \text{ } \dots) \text{ } op_{k-2} \text{ } z_{k-1}.$$

给定 n, m, k 与长度为 $k - 1$ 的二元布尔函数序列 $[op_0, op_1, \dots, op_{k-2}]$ ，记该二元布尔函数序列对应的 k 元布尔函数为 f 。保证 $m = 3n$ ， $k \in \{2, 3, 4\}$ 且 $op_0, op_1, \dots, op_{k-2}$ 不全为 2。

小 D 认为，对于在所有取值在 $[0, n - 1]$ 中的 k 元互不相同整数组中独立均匀随机的 $x_{i,j}$ ($0 \leq i < m, 0 \leq j < k$)， f 与 x 对应的 n, m 伪随机函数与独立均匀随机没有区别。你认为他的说法是错误的，为此，小 D 将会分别按照两种不同方式生成一些 m 元组，并让你判断是按照哪一种方式生成的。具体地，小 D 将会询问你 100 次，每次首先会在所有取值在 $[0, n - 1]$ 中的 k 元互不相同整数组中独立均匀随机生成 $x_{i,j}$ ($0 \leq i < m, 0 \leq j < k$)，记 f 与 x 对应的 n, m 伪随机函数为 \hat{f} ，然后以各 50% 的概率随机选择以下两种方式之一生成 c 个 m 元组 s_0, s_1, \dots, s_{c-1} ：

1. 直接在 $\{0, 1\}^m$ 中独立均匀随机生成 c 个 m 元组 s_0, s_1, \dots, s_{c-1} 。
2. 在 $\{0, 1\}^n$ 中独立均匀随机生成 c 个 n 元组 t_0, t_1, \dots, t_{c-1} ，然后令 $s_i = \hat{f}(t_i)$ ($0 \leq i < c$)；

你需要判断出小 D 生成的 c 个 m 元组是由哪一种方式生成的。由于真随机可能生成任何串，你只需要尽可能多地正确判断出小 D 给出的串由哪一种方式生成即可。

【实现细节】

你不需要，也不应该实现 `main` 函数。

你应确保提交的程序包含头文件 `prg.h`，可在程序开头加入以下代码实现：

```
1 #include "prg.h"
```

你需要实现以下函数：

```
1 int solve(int n, int m, int k, std::vector<int> op,
           std::array<std::vector<int>, 3000> x, int c,
           std::array<std::array<int, 3000>, 25> s);
```

- 上述所有变量和题目中的定义相同，且保证 $n = 1000$ ， $m = 3000$ ， $k \in \{2, 3, 4\}$ ， $c = 25$ ；
- 保证 op 的长度为 $k - 1$ ， $op_i \in \{0, 1, 2\}$ ($0 \leq i \leq k - 2$)，且 $op_0, op_1, \dots, op_{k-2}$ 不全为 2；
- 保证 x 中每个元素的长度均为 k ，且每个元素在所有取值在 $[0, n - 1]$ 中的 k 元互不相同整数组中独立均匀随机生成；
- 保证 s 按照题目描述中的两种方式之一生成；
- 你需要返回 $u \in \{1, 2\}$ 表示你认为的答案，其中返回 1 表示你认为 s 由第一种方式生成，返回 2 表示你认为 s 由第二种方式生成。
- 对于每个测试点，保证该函数会被交互库调用恰好 100 次。

题目保证在规定的限制下，交互库的运行时间不会超过 300 ms；交互库使用的内存大小固定，且不超过 32 MiB。

【测试程序方式】

下发文件中的 `grader.cpp` 是提供的交互库参考实现，最终测试时所用的交互库实现与该参考实现有所不同，因此你的解法不应该依赖交互库实现。

你可以在下发文件目录下使用如下命令编译得到可执行程序：

```
1 g++ grader.cpp prg.cpp -o prg -O2 -std=c++14 -static
```

对于编译得到的可执行程序：

- 可执行文件将从标准输入读入以下格式的数据：
 - 输入的第一行一个整数 $seed$ 表示将使用的随机种子。你需要保证 $0 \leq seed < 2^{64}$ 。
 - 输入的第二行两个整数 $k, type$ ，其中 k 的含义如题目描述中所示， $type$ 表示表达式的生成方式。你需要保证 $k \in \{2, 3, 4\}$ ， $type \in \{0, 1\}$ 。
 - * 若 $type = 0$ ，每次询问时， op 在所有长度为 $k - 1$ 且不全为 2 的二元布尔函数序列中独立均匀随机生成；
 - * 若 $type = 1$ ，输入的第三行 $k - 1$ 个整数 $op_0, op_1, \dots, op_{k-2}$ ，含义如题目描述中所示。

- 输入完成后，交互库进行以下过程恰好 100 次：
 - 按照题目描述中的方式生成 x, s ，其中所有的随机数使用 `std::mt19937_64` 与输入的 $seed$ 生成。
 - 调用恰好一次 `solve()` 函数。
- 可执行文件将输出以下格式的数据到**标准输出**：
 - 第一行表示你的测试结果，其中包含你在 100 组数据中的正确组数以及该测试点中的得分。

【下发文件说明】

在下发文件中：

1. `grader.cpp` 是提供的交互库参考实现。
2. `prg.h` 是头文件，你不需要关心其具体内容。
3. `template_prg.cpp` 是提供的示例代码，你可以在此代码的基础上实现。

【子任务】

对于所有测试数据，保证

- $n = 1000$, $m = 3000$, $k \in \{2, 3, 4\}$, $c = 25$,
- op 的长度为 $k - 1$, $\forall 0 \leq i \leq k - 2, op_i \in \{0, 1, 2\}$, 且 $op_0, op_1, \dots, op_{k-2}$ 不全为 2,
- x 中每个元素的长度均为 k , 且每个元素在所有取值在 $[0, n - 1]$ 中的 k 元互不相同整数组中独立均匀随机生成,
- s 按照题目描述中的两种方式之一生成。

子任务编号	分值	$k =$	特殊性质
1	5	2	无
2	15	3	
3	30	4	有
4	50		无

特殊性质：每次询问时 op 在所有长度为 $k - 1$ 且不全为 2 的二元布尔函数序列中独立均匀随机生成。

【评分方式】

注意：

- 你不应当通过非法方式获取交互库的内部信息，如试图直接读取交互库中存储的值，或直接与标准输入、输出流进行交互。此类行为将被视为作弊；

- 最终的评测交互库与样例交互库的实现有所不同，因此你的解法不应该依赖交互库实现。

本题首先会受到和传统题相同的限制，例如编译错误会导致整道题目得 0 分，运行时错误、超过时间限制、超过空间限制等会导致相应测试点得 0 分等。你只能在程序中访问自己定义的和交互库给出的变量及其对应的内存空间，尝试访问其他空间将可能导致编译错误或运行错误。

在上述条件基础上：

对于每个测试点，假设你在 100 组数据中的 x 组输出了错误的答案，则该测试点的得分为 $s \times \max(0, 1 - \frac{1}{4} \log_2(\max(x, 1)))$ ，其中 s 为该测试点对应子任务的分值。注意，即使 $x = 1$ ，你也会获得这个测试点的满分。

本题有多个子任务，每个子任务中恰好包含 20 个测试点，每个子任务的得分为该子任务中所有测试点得分的最小值。

【提示】

你在一个子任务的得分与该子任务总分的比值的期望 E 与你的代码通过一组数据的概率 p 有关。通过数值模拟可以得到如下数据（仅供参考）：

- 当 $p = 0.95$ 时， E 约为 0.2。
- 当 $p = 0.98$ 时， E 约为 0.4。
- 当 $p = 0.99$ 时， E 约为 0.6。
- 当 $p = 0.995$ 时， E 约为 0.75。
- 当 $p = 0.999$ 时， E 约为 0.98。

【后记】

当你切掉了这道题之后，你不难发现小 D 说的话没有道理！