IOI2025 中国国家集训队集中培训

第二试

时间： 2024 年 12 月 4 日 08:30 ∼ 13:30

题目名称	阿尔塔尔 2	颠倒歌	路南柯
题目类型	交互型	传统型	传统型
输入	标准输入	标准输入	标准输入
输出	标准输出	标准输出	标准输出
每个测试点时限	1.0 秒	3.0 秒	1.0 秒
内存限制	512 MiB	512 MiB	512 MiB
子任务数目	3	11	2
测试点是否等分	否	否	否

阿尔塔尔 2（altar）

这是一道交互题。

【题目背景】

一年过去了，阿尔塔尔站在曾被他摧毁的祭坛面前。祭坛上的宝石早已失去了原有的光泽。

“解铃还须系铃人，是这样吗？”他喃喃自语。

【题目描述】

祭坛呈正 n 边形，每个顶点上有一个未激活的宝石。我们用 1 至 n 的整数给每种宝石编号。

宝石之间存在着能量流动关系，这使得激活一些宝石之后可以更容易地激活其他宝石。对于每一对宝石 (i, j)（$i \neq j$），要么从 i 到 j 存在流动，要么从 j 到 i 存在流动，即当我们把宝石看作点、从 p 到 q 的流动看作一条从 p 到 q 的有向边，得到的图是一个 n 个点的**竞赛图**。阿尔塔尔并不知道宝石之间的能量流动关系如何。

为了激活祭坛，阿尔塔尔需要激活所有的宝石。为此，阿尔塔尔需要首先选择 n 个宝石中的一个将其初始激活，然后进行若干次**能量传递**。每一次**能量传递**的效果为：若这次**能量传递**前宝石 y 未被激活，且存在一个激活的宝石 x 到 y 有流动，那么**能量传递**后 y 就会被激活。在**能量传递**前就被激活的宝石在**能量传递**之后依然是被激活的。

由于**能量传递**需要耗费大量体力，阿尔塔尔希望找到一个宝石，将其初始激活之后可以使用**最少次数的能量传递**激活所有宝石。为此，阿尔塔尔可以进行若干次**能量感知**：给出 $1 \leq i, j \leq n$ 且 $i \neq j$，一次**能量感知**可以确定宝石 i 和 j 之间的能量流动关系。

你需要帮助阿尔塔尔使用尽可能少的**能量感知**操作，确定初始激活的宝石编号。可以证明总是存在一个宝石，将其初始激活后使用有限次能量传递即可激活所有宝石。

【实现细节】

请确保你的程序开头有 `#include "altar.h"`

你不需要也不应该实现主函数。你需要实现以下函数：

```
int altar(int n);
```

- 其中 n 表示祭坛上的宝石数量。
- 你需要返回一个整数 x，表示阿尔塔尔需要初始激活宝石 x。你需要保证 $1 \leq x \leq n$，且在所有宝石中初始激活 x 可以让阿尔塔尔使用最少次数的**能量传递**激活所有宝石。
- **在最终测试时，交互库会在一次运行中调用 $T = 300$ 次 `altar` 函数。**

你可以调用以下函数进行**能量感知**：

```
bool sense(int i, int j);
```

- 你需要保证 $1 \le i, j \le n$ 且 $i \ne j$。
- 若 i 与 j 的能量流动为 i 向 j，其返回 `true`，否则返回 `false`。
- 你需要保证在一次 `altar` 的调用中 `sense` 的调用次数不超过 4.5×10^4。

在满足题目条件和数据范围的情况下，最终测试时交互库的运行时间不会超过 150 ms，运行空间不会超过 256 MiB。

交互库不是自适应的，即能量流动关系是固定的，不会随着交互过程改变。

【测试程序方式】

试题目录下的 `grader.cpp` 是我们提供的交互库参考实现。最终测试的交互库与样例交互库有一定不同，故你的实现不应该依赖样例交互库实现。

你需要在本题目录下使用如下命令编译得到可执行程序：

`g++ grader.cpp sample.cpp -o sample -O2 --std=c++14 -lm`

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：
  - 第一行一个整数 n 表示祭坛上的宝石数，你需要保证 $3 \le n \le 300$。
  - 接下来 n 行，第 i 行一个长度为 n 的 `01` 字符串，其中第 $j (j \ne i)$ 个字符为 `1` 表示能量流动从 i 到 j，否则表示能量流动从 j 到 i。你需要保证第 i 行第 j 个字符和第 j 行第 i 个字符间恰好有一个为 `1` 另一个为 `0`，**样例交互库并不会判断输入是否满足条件**。
- 读入完成之后，交互库将调用恰好一次函数 `altar`。
- 在 `altar` 函数退出后，如果你给出了正确的宝石编号，交互库将在标准输出流输出 `Correct. X`，其中 X 表示 `sense` 的调用次数，并在标准错误流输出你返回的宝石编号以及对应的**能量传递**次数；否则交互库将在标准输出流输出 `Wrong Answer` 并在标准错误流输出对应错误信息。

【样例 1 输入】

```
4
0011
1010
0000
0110
```

【样例 1 输出】

```
Correct. 1
You report 2, with number of energy propagations to be 2
```

【样例 1 解释】

该样例输出为下发的样例程序在该组样例下的输出。该样例中，`altar` 返回 1、2、4 均满足条件。

【子任务】

对于所有测试数据，单个测试点中 `altar` 函数的调用次数 $T = 300$，每一次调用都有 $3 \le n \le 300$。

子任务 1（10 分，共 1 个测试点）：$n = 300$，两个宝石之间的能量流动方向在两种可能间独立均匀随机。

子任务 2（10 分，共 2 个测试点）：存在一个宝石流向所有其他宝石。

子任务 3（80 分，共 7 个测试点）：没有特殊限制，依赖子任务 1、2。

【评分方式】

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、超过空间限制都会导致相应测试点得 0 分。选手只能在程序中访问自己定义的和交互库给出的变量或数据，及其相应的内存空间。尝试访问其他位置空间将可能导致编译错误或运行错误。

对于每个测试点，如果你的程序出现了非法的函数调用，或没有在所有测试数据中都返回正确的宝石，你将获得该测试点 0% 的分数。否则你会按照在 T 组 `altar` 调用中的 `sense` 调用次数的**平均值** X，依据如下公式计算该测试点的得分占比：

- 若 $45000 \ge X \ge 10^4$，你可以获得 $\left(1 + 29 \times \frac{45000-X}{35000}\right)\%$ 的分数；
- 若 $10^4 > X \ge 2100$，你可以获得 $\left(30 + 30 \times \frac{10^4-X}{7900}\right)\%$ 的分数；
- 若 $2100 > X \ge 700$，你可以获得 $\left(60 + 20 \times \left(\frac{2100}{X} - 1\right)\right)\%$ 的分数；
- 若 $X \le 700$，你可以获得该测试点 100% 的分数；

每个子任务的分数为该子任务中所有测试点的得分占比的最小值与该子任务满分的乘积。

选手不应通过非法方式获取交互库的内部信息，如试图与标准输入、输出流进行交互。此类行为将被视为作弊。

颠倒歌（diandao）

【题目描述】

对于树 $T(V,E)$ 和 $S \subseteq V$，记 $f(S,T)$ 表示 T 的对 S 的导出子图（即仅保留 S 中的点和两端都在 S 中的边得到的图）中度数小于等于 1 的点的数量。

对于两棵树 $T_1(V_1,E_1), T_2(V_2,E_2)$，若 $V_1 = V_2$，我们称 $T_1 \preccurlyeq T_2$ 当且仅当对于任意 $S_2 \subseteq V_2$，存在 S_1 满足 $S_2 \subseteq S_1 \subseteq V_1$ 且 $f(S_1,T_1) \le f(S_2,T_2)$。

称 T_1, T_2 等价当 $T_1 \preccurlyeq T_2$ 且 $T_2 \preccurlyeq T_1$，记作 $T_1 \sim T_2$。该等价关系将 n 个点的有标号树划分成若干等价类。

问：

1. 给定 k 棵 n 个点的树 $T_1, T_2 \dots, T_k$，求满足 $T \preccurlyeq T_i, \forall 1 \le i \le k$ 的**有标号树 T 构成的等价类**数量。
2. 给定 k 棵 n 个点的树 $T_1, T_2 \dots, T_k$，求满足 $T_i \preccurlyeq T, \forall 1 \le i \le k$ 的**有标号树** T 数量。

注意两问的计数对象不同。两问答案均对 $998,244,353$ 取模。

保证答案取模后非 0。

【输入格式】

从标准输入读入数据。

输入第一行一个整数 p，其中 $p \in \{0,1\}$，$p = 0$ 表示询问第一问，否则表示询问第二问。

接下来一行两个正整数 k, n，分别表示输入树的数量以及点数。

接下来依次输入 k 棵树，对于每棵树输入 $n-1$ 行每行两个正整数 u, v 描述树中的一条边。

【输出格式】

输出到标准输出。

输出一行一个整数表示答案对 $998,244,353$ 取模后的结果。

【样例 1 输入】

```
0
1 4
1 2
1 3
1 4
```

【样例 1 输出】

```
2
```

【样例 1 解释】

可以证明 4 个点的有标号树被划分成了 5 个等价类，所有的链在同一个等价类，而其它分别以每个点为菊花中心对应一个等价类。

可以验证链对应的等价类和该树本身所在的等价类均满足要求，而其他等价类不满足要求。

【样例 2 输入】

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

【样例 2 输出】

```
16
```

【样例 2 解释】

可以验证所有 4 个点的有标号树共 16 个均满足要求。

【样例 3】

见题目目录下的 *3.in* 与 *3.ans*。

【样例 4】

见题目目录下的 *4.in* 与 *4.ans*。

【样例 5】

见题目目录下的 *5.in* 与 *5.ans*。

【样例 6】

见题目目录下的 ***6.in*** 与 ***6.ans***。

【样例 7】

见题目目录下的 ***7.in*** 与 ***7.ans***。

【样例 8】

见题目目录下的 ***8.in*** 与 ***8.ans***。

【样例 9】

见题目目录下的 ***9.in*** 与 ***9.ans***。

【样例 10】

见题目目录下的 ***10.in*** 与 ***10.ans***。

【子任务】

对于所有数据，保证 $p \in \{0,1\}$，$3 \le n \le 5000$，$1 \le k \le 1000$，$1 \le u,v \le n$，答案取模后不等于 0。

<table>
<tr><th>子任务编号</th><th>p</th><th>n</th><th>k</th><th>树形态</th><th>分数</th></tr>
<tr><td>1</td><td>$\in \{0,1\}$</td><td>≤ 8</td><td>≤ 4</td><td>无特殊形态</td><td>10</td></tr>
<tr><td>2</td><td rowspan="5">$=0$</td><td rowspan="4">≤ 200</td><td rowspan="2">≤ 1</td><td>菊花</td><td>4</td></tr>
<tr><td>3</td><td rowspan="4">无特殊形态</td><td>8</td></tr>
<tr><td>4</td><td>≤ 2</td><td>9</td></tr>
<tr><td>5</td><td>≤ 40</td><td>10</td></tr>
<tr><td>6</td><td>$\le 5,000$</td><td>$\le 10^3$</td><td>14</td></tr>
<tr><td>7</td><td rowspan="5">$=1$</td><td rowspan="4">≤ 200</td><td rowspan="2">≤ 1</td><td>链</td><td>7</td></tr>
<tr><td>8</td><td rowspan="4">无特殊形态</td><td>7</td></tr>
<tr><td>9</td><td>≤ 2</td><td>10</td></tr>
<tr><td>10</td><td>≤ 40</td><td>10</td></tr>
<tr><td>11</td><td>$\le 5,000$</td><td>$\le 10^3$</td><td>11</td></tr>
</table>

“树形态”中，“菊花”指存在一个向所有点有直接连边的点，“链”指所有点度数不超过 2。

路南柯（nanke）

【题目描述】

称一个 $1 \sim n$ 的排列 $\{p\} = \{p_1, p_2, \cdots, p_n\}$ 是一棵 n 个点、点编号为 1 至 n 的树 T 的**拓扑序列**，当且仅当对于任意 $1 \le i < n$，恰好存在唯一的 $j > i$ 满足 p_i 与 p_j 之间有连边。

给定树 T，你需要给出尽可能少的该树的拓扑序列 $\{p_1\}, \{p_2\}, \cdots, \{p_k\}$，使得有且仅有树 T 满足 $\{p_1\}, \{p_2\}, \cdots, \{p_k\}$ 均为该树的合法拓扑序列。

【输入格式】

从标准输入读入数据。

本题有多组测试数据。输入第一行一个正整数 T，表示测试数据组数，接下来依次输入每组测试数据。

对于每组数据，输入第一行一个正整数 n，表示给定树的大小。接下来 $n-1$ 行，每行两个正整数 u, v 描述树中存在的一条边。

【输出格式】

输出到标准输出。

对于每组数据，输出第一行一个正整数 k 表示你给出的拓扑序列数量，接下来 k 行，每行输出一个 $1 \sim n$ 的排列，描述你给出的拓扑序列。你需要保证 $1 \le k \le n$，且这 k 个拓扑序列均为对应输入的合法拓扑序列，且只有一棵树满足这些拓扑序列都是其合法拓扑序列。

【样例 1 输入】

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

【样例 1 输出】

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

【子任务】

对于所有测试数据，$1 \le T \le 100$，$3 \le n \le 100$，$1 \le u, v \le n$。

本题共有两个测试点。

测试点编号	分值	T	n
1	20	$=10$	$=10$
2	80	$=10^2$	$=10^2$

特别地，所有测试点中每组数据均为从所有 n 个点的有标号树中等概率随机选择生成得到的。

【评分方式】

对于一个测试点内部的某组数据：

1. 你需要保证 k 以及你输出的序列中每个数都为 $[1, n]$ 范围内的正整数，否则**整个测试点**将会获得 0 分。
2. 若你给出的序列中存在一个不是输入给定树的拓扑序列，那么你**这组数据**的得分比例将会是 0。
3. 若存在多棵树满足其这些序列都是其合法的拓扑序列，那么你**这组数据**的得分比例将会是 0。
4. 否则如果最优解为 K，而你构造的序列数量为 k，那么你**这组数据**的得分比例将为 0.97^{k-K}。

一个测试点的得分比例将会是该测试点内所有数据的得分比例的**平均值**，一个测试点的实际得分值将会是其总分乘以得分比例。