

ION 1202 模拟赛

ASDFZ

时间：1202 年 -1 月 -i 日 08:00 ~ 13:00

题目名称	七管荧光灯	混乱	超立方体
题目类型	传统型	传统型	交互型
目录	qgygd	chaos	hypercube
可执行文件名	qgygd	chaos	hypercube
输入文件名	qgygd.in	chaos.in	hypercube.in
输出文件名	qgygd.out	chaos.out	hypercube.out
每个测试点时限	1.0 秒	6.0 秒	4.0 秒
内存限制	256 MiB	256 MiB	256 MiB
子任务数目	4	4	4
是否有部分分	否	否	是

提交源程序文件名

对于 C++ 语言	qgygd.cpp	chaos.cpp	hypercube.cpp
-----------	-----------	-----------	---------------

编译选项

对于 C++ 语言	-lm -O2 -std=c++11
-----------	--------------------

注意事项与提醒：

1. 提交时需要建子文件夹。
2. 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）。
3. 程序可使用的栈内存空间限制与题目的内存限制一致。

七管荧光灯 (qgygd)

题目描述

双管荧光灯有一个七个荧光管的灯，形如下图：

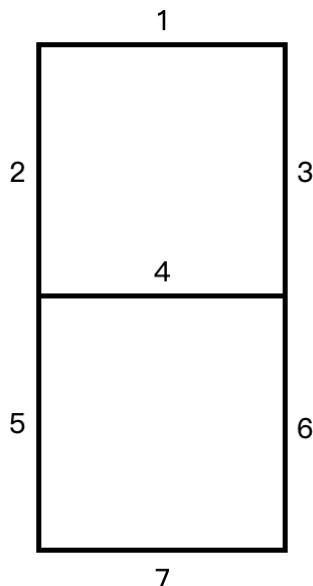


Figure 1: 一个七管荧光灯和它每根荧光管的编号

双管荧光灯拿它玩 nim 游戏，规则如下：

1. 一开始每一根荧光管上有一些石子。
2. 先后手轮流取石子，每一次可以选择某些连通的荧光管（但是选择的荧光管不得成环），并从其中的每一根荧光管上各选择若干个石子移走（可以为 0，但是一轮操作至少要取走一个石子）。
3. 不能取的人输。

但是双管荧光灯玩这个游戏百战百胜，现在他有点无聊，来问你这个问题：

请问，如果第 i 根荧光灯上的石子为 $[l_i, r_i]$ 中的任意整数，会有多少的局面使得先手必胜呢？由于局面非常多，你只要输出其对 998244353 取模的结果。

输入格式

输入共七行，每行两个整数 l_i, r_i ，分别表示第 i 个荧光管的石子数的下限和上限。

输出格式

输出一行一个整数，表示先手必胜的方案数对 998244353 取模的结果。

样例 1 输入

```
10 10
20 20
0 0
30 30
0 0
40 40
40 40
```

样例 1 输出

```
1
```

样例 2 输入

详见选手文件夹下的 `qgygd/qgygd2.in` 文件。

样例 2 输出

详见选手文件夹下的 `qgygd/qgygd2.ans` 文件。

样例 3 输入

详见选手文件夹下的 `qgygd/qgygd3.in` 文件。

样例 3 输出

详见选手文件夹下的 `qgygd/qgygd3.ans` 文件。

数据规模与约定

对于 100% 的数据，有 $0 \leq l_i \leq r_i \leq 10^{18}$ 。

本题采用捆绑测试，共 4 个 subtask，你必须通过每个 subtask 中的所有测试点才能获得该 subtask 的分数。

子任务编号	$r_i \leq$	特殊性质	分数
1	4	无	10
2	10^{18}	保证 $l_4 = r_4 = 0$	30
3	2×10^3	无	30
4	10^{18}	无	30

混乱 (chaos)

题目描述

双管荧光灯是混乱邪恶势力，而小 G 是正义的伙伴。

双管荧光灯在一个长为 n 的数轴上造成了若干混乱，而小 G 尝试去修补。

但是小 G 太弱小了没有力量，所以他无能为力。

于是小 G 找到了小 W 和她签订契约让她成为马猴烧酒，小 W 会在某些时刻学会魔法 i ，它可以表示为三元组 (l_i, r_i, w_i) ，代表她可以花 w_i 的体力修补 $[l_i, r_i]$ 中的所有混乱。由于某种机缘巧合， l_i 和 r_i 是随机的。

在某些时刻双管荧光灯会在一些位置造成混乱，而小 G 则想立即让小 W 修补好，但是他不知道怎样节约体力，所以你来求助你，由于在线等，急，所以本题强制在线。

输入格式

第一行二个正整数 n, q 代表数轴的长度和接下来会发生的事件的个数。

接下来 q 行。每行表示一个事件，描述事件的方式如下：

- $1\ l\ r\ w$ 代表小 W 学会了技能 (l', r', w) 其中 $l' = l \oplus lasans, r' = r \oplus lasans$ 且 $1 \leq l' \leq r' \leq n$ 。
- $2\ k\ s_1\ s_2 \cdots s_k (k > 0)$ 代表双管荧光灯在 $S = \{s_1 \oplus lasans, s_2 \oplus lasans, \cdots, s_k \oplus lasans\}$ (不保证升序排序) 这些位置上制造了混乱，现在你需要用已有的技能修补完这些混乱。

其中 \oplus 是异或运算， $lasans$ 为上次混乱你给出的答案对 $10^9 + 7$ 取模的结果，若在此之前还没有任何混乱，则 $lasans = 0$ 。

输出格式

对于每次混乱，请输出一行一个整数，代表最小体力消耗。若小 W 无论如何都无法修复，请输出 0。

样例 1 输入

```
10 7
2 4 1 5 7 8
1 4 5 1
1 2 6 4
2 2 4 5
1 7 8 3
1 8 11 5
2 6 6 3 8 5 9 11
```

样例 1 输出

```
0
1
12
```

样例 2 输入

详见选手文件夹下的 `chaos/chaos2.in` 文件。

样例 2 输出

详见选手文件夹下的 `chaos/chaos2.ans` 文件。

样例 3 输入

详见选手文件夹下的 `chaos/chaos3.in` 文件。

样例 3 输出

详见选手文件夹下的 `chaos/chaos3.ans` 文件。

数据规模与约定

对于 100% 的数据，有 $1 \leq n, q, \sum k \leq 3 \times 10^5$ ，对于所有技能，有 $1 \leq w \leq 10^9$ ， l', r' 为随机生成，随机策略为：从 1 到 n 中等概率随机两个正整数 a, b ，令 $l' = \min(a, b), r' = \max(a, b)$ 。

本题采用捆绑测试，共 4 个 subtask，你必须通过每个 subtask 中的所有测试点才能获得该 subtask 的分数。

子任务编号	$n, q, \sum k \leq$	分数
1	10^3	10
2	8×10^4	20
3	2×10^5	30
4	3×10^5	40

超立方体 (hypercube)

题目描述

这是一道交互题。

双管荧光灯有一棵树，编号为 $1 \sim n$ ，但是他在游历四维空间的时候不小心把它弄掉了。

一个有生命的超立方体捡到了这棵树，但是为了证明这棵树是他的，双管荧光灯必须要回答出这棵树的长什么样子。

双管荧光灯已经忘记了树的形态，但是超立方体允许他问不超过 Q 个问题，如果他能够在询问次数用完之前猜出这棵树，那么超立方体就会把树还给他。

每个问题由一个点 u 和一个点集 S 构成，超立方体会告诉你 $\bigoplus_{v \in S} \text{dist}(u, v)$ 。其中， $\text{dist}(u, v)$ 定义为 u 到 v 路径上所有点标号的异或和， \oplus 代表异或。

双管荧光灯当然能成功找回他的树，但是他觉得这一题十分有意义，所以他想拿这个问题来考考你。

若你不能在 Q 次询问中得到这棵树，你也可以获得一定的分数，详见【评分方式】一部分。

本题多组数据。

实现细节

选手目录下已经提供了一个 `template_hypercube.cpp`，你可以将其重命名为 `hypercube.cpp` 并直接在其基础上答题。

请确保你的程序开头有 `#include "hypercube.h"`

你不需要也不应该实现主函数，你只需要实现如下函数：

`void solve(int n, int Q, int data_type);`

你可以调用的程序接口如下：

`int query(int u, std::vector<int> S);`

表示询问点 u 和点集 S ，请注意， S 集合不能为空且 S 集合不能包含重复节点，保证交互器实现函数的时间复杂度为 $O(|S|)$ 。理论上对于 $\sum |S|$ 没有限制，但是你不应当使交互器使用的时间过多，这可能会导致你超出时间限制。

`void report(int u, int v);`

表示报告找到了一条无向边 (u, v) ，注意你只能报告 $n - 1$ 次且不能报告重复的边。

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，最终测试时所用的交互库实现与该参考实现有所不同，但时间效率一致，选手的解法不应该依赖交互库实现。

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 hypercube.cpp -o hypercube -O2 -std=c++11 -lm`

编译得到的可执行程序将从标准输入读入以下格式的数据：

第一行一个正整数 T ，代表数据组数。

接下来 T 组数据，每组数据由 n 行组成，其中第一行有三个整数 $n, Q, \text{data_type}$ 表示点数、最大询问数和数据点类型。

每组数据的接下来 $n-1$ 行，每行两个正整数 $u, v (1 \leq u, v \leq n)$ 描述了树的一条边。

读入完成之后，对于每组数据交互库将调用恰好一次函数 solve，用输入的数据测试你的函数。你的函数正确返回后，交互库会判断你的计算是否正确，若正确则会输出交互函数调用次数相关信息，否则会输出相应的错误信息并立即退出。

若你的程序正确运行完毕，交互器会在最后输出 Correct! 该测试点的分数和该测试点的总用时。

下发文件说明

在本试题目录下：

grader.cpp 是我们提供的交互库参考实现。

hypercube.h 是头文件，选手不用关心具体内容。

hypercube.in 是样例输入，可供测试，其 `data_type` 均为 0。

template_hypercube.cpp 是我们提供的样例解题源代码。

最终评测只会收取 hypercube.cpp，修改选手目录下其他文件对评测无效。

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、超过空间限制等会导致相应测试点得 0 分等。你只能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。

由于 report 的调用次数限制为 $n-1$ ，你的每次调用都必须记录一条新的且存在的边；即每次调用 report(x, y) 时，应满足：有一条连接 x 号节点和 y 号节点的边，且在这次调用之前从未调用过 report(x, y) 或 report(y, x)。

你实现的函数 solve 正常返回。

在 solve 函数返回时，你已经通过调用 report 记录了全部 $n-1$ 条边。

你的每次函数调用均合法，且调用 query 的次数不超过 $1.16Q$ 。

若你不满足上述条件中的任意一个，程序会立即退出，你会得到 0 分。

在此基础上，每个测试点的得分为每组数据的得分的最小值，计算一组数据的得分方式如下：

若你调用 query 的次数为 t ，该数据限制 query 的调用次数为 Q ，得分与满分之比 s 为关于 $x = t/Q$ 的函数 $f(x)$ 保留两位小数（直接截取小数点后 2 位）。

$$f(x) = \begin{cases} 1 & , x \leq 1 \\ \max(0, -6e^{x-1} + 7) & , x > 1 \end{cases}$$

数据规模与约定

对于 100% 的数据，有 $1 \leq n \leq 3000$ ， $1 \leq T \leq 10$ ， $Q = 4 \times 10^4$ ，data_type 为对应的子任务编号。

本题采用捆绑测试，共 4 个 subtask，每个的得分为该 subtask 所有测试点的分数的最小值。

子任务编号	$n \leq$	特殊性质	分数
1	2000	保证每个点的度数不超过 18	20
2	200	无	10
3	2000	无	30
4	3000	无	40