# I 君的探险（explore）

**这是一道交互题。**

## 【题目背景】

时隔半年，I 君的商店终于开不下去了，他决定转让商店，做一名探险家去探索未知的广阔世界。

根据古书记载，他在一个大荒漠的腹地找到了未知文明创造的地下宫殿，宫殿由 $N$ 个大型洞穴和 $M$ 条连接这些洞穴的双向通路构成。I 君能借助古书分辨所处的洞穴，但书中并没有记录 $M$ 条通路的连接结构，因此他难以搜寻传说中藏在宫殿里的无尽财宝。

不过现在 I 君发现了一个神秘机关，通过它可以获知宫殿的信息，I 君决定利用这个机关来得到宫殿的连接结构，请你来协助他。

## 【题目描述】

地下宫殿可以抽象成一张 $N$ 个点、$M$ 条边的**无向简单图**（简单图满足任意两点之间至多存在一条直接相连的边），洞穴从 $0 \sim n-1$ 编号。目前你**并不知道**边有哪些。

每个洞穴都拥有一个光源，光源有开启、关闭两种状态，只有当光源处于开启状态时它所在的洞穴才会被照亮。初始时所有的光源都处于**关闭**状态，而光源的状态只能用 I 君发现的神秘机关改变。更具体的，使用神秘机关可以进行如下四种操作：

1. 向机关给定一个编号 $x$，机关将会**改变** $x$ 号洞穴，以及**与 $x$ 号洞穴有通路直接相连的洞穴**的光源状态。即原来开启的光源将会关闭；原来关闭的光源将会开启。
2. 向机关给定一个编号 $x$，机关将会显示**当前** $x$ 号洞穴光源的状态。
3. 向机关给定两个编号 $x, y$，表示你确定有一条连接 $x$ 号洞穴与 $y$ 号洞穴的通路，并让机关记录。
4. 向机关给定一个编号 $x$，机关将会判断与 $x$ 号洞穴相连的通路是否**都已被记录**。

机关在完成上一次操作后才能进行下一次操作。机关不能随意使用，因此每种操作的使用次数都有限制，分别为 $L_m, L_q, M, L_c$。你的任务是，编写一个程序，帮助 I 君决定如何合理利用神秘机关，从而正确地找到这 $M$ 条通路。

## 【实现细节】

你不需要，也不应该实现主函数，**你只需要**实现函数 `explore(N, M)`，这里的 `N` 和 `M` 分别表示洞穴和通路的个数。你可以通过调用如下四个函数来和交互库进行交互：

1. `modify(x)`
   - 这个函数可以令机关执行操作 1，给定的编号为 $x$。
   - 你需要保证 $0 \le x < N$，这个函数没有返回值。
2. `query(x)`

- 这个函数可以令机关执行操作 2，给定的编号为 $x$。
- 你需要保证 $0 \le x < N$，这个函数返回 0 或 1，表示目前 $x$ 号洞穴的光源为关闭（0 表示）或开启（1 表示）状态。

3. `report(x, y)`

- 这个函数可以令机关执行操作 3，给定的编号为 $x, y$。
- 你需要保证 $0 \le x, y < N$ 且 $x \neq y$，这个函数没有返回值。

4. `check(x)`

- 这个函数可以令机关执行操作 4，给定的编号为 $x$。
- 你需要保证 $0 \le x < N$，这个函数返回 0 或 1，其中返回 1 当且仅当与 $x$ 号洞穴相连的所有通路都已通过操作 3 被记录。

评测时，交互库会恰好调用 `explore` 一次。

**本题保证所使用的图在交互开始之前已经完全确定，不会根据和你的程序的交互过程动态构造，因此题目中的交互操作都是确定性的，你不需要关心这些操作在交互库中的具体实现。**

**数据保证在调用次数限制下，交互库运行所需的时间不超过 1s；交互库使用的内存大小固定，且不超过 128MB。**

## 【实现方法】

选手工作目录下已经提供了一个 `template_explore.cpp/c/pas`，请将这个文件拷贝一份，重命名为 `explore.cpp/c/pas`，然后在其基础上答题。

1. 对 C++ / C 语言选手

- 请确保你的程序开头有 `#include "explore.h"`。
- 你需要实现的函数 `explore` 的接口信息如下：

```
void explore(int N, int M);
```

- 你可以调用的交互函数的接口如下：

```
void modify(int x);
int query(int x);
void report(int x, int y);
int check(int x);
```

2. 对 Pascal 语言选手

- 注意：Pascal 的代码中实现接口的语法较为复杂，**请选手直接在下发的 `template_explore.pas` 的基础上进行答题，而不是自己从头实现代码。**
- 你需要实现的函数 `explore` 的接口信息如下：

```
procedure _explore(N, M : longint);
```

- **注意：这里的函数名称是 `_explore` 而非 `explore`，如果使用 `explore` 将导致编译失败。**

- 你可以调用的交互函数的接口如下：

```
procedure modify(x : longint);
function query(x : longint) : longint;
procedure report(x : longint; y : longint);
function check(x : longint) : longint;
```

**【测试程序方式】**

试题目录下的 **grader.cpp/c** 以及 **graderhelperlib.pas** 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，因此选手的解法不应该依赖交互库实现。

1. 对 C/C++ 语言的选手：
- 你需要在本题目录下使用如下命令编译得到可执行程序：
- 对于 C 语言：`gcc grader.c explore.c -o explore -O2 -lm`
- 对于 C++ 语言：`g++ grader.cpp explore.cpp -o explore -O2 -lm`

2. 对于 Pascal 语言的选手：
- 你需要在本题目录下使用如下命令编译得到可执行程序：

`fpc grader.pas -o"explore" -O2`

3. 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：

第一行包含三个整数 $L_m, L_q, L_c$ ，第二行包含两个整数 $N, M$，意义如题面描述。

接下来 $M$ 行，每行两个整数 $x, y$，描述一条连接 $x$ 号洞穴与 $y$ 号洞穴的通路。
- 读入完成之后，交互库将调用恰好一次函数 `explore`，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 Correct 和交互函数调用次数相关信息，否则会输出相应的错误信息。

**【交互示例】**

假设可执行文件读入的数据为：

```
100 200 300
3 2
0 1
1 2
```

数据第一行的三个整数分别表示三种操作的调用次数限制，即 `modify(x)` 调用次数不能超过 100，`query(x)` 调用次数不能超过 200，`check(x)` 调用次数不能超过 300。

数据第二行的两个整数分别表示洞穴数和通路条数，即 $N = 3$ , $M = 2$。

report(x, y) 调用次数不能超过 $M$，该例子中即不超过 2 次。

下面是一个正确的交互过程：

| 选手程序 | 交互库 | 说明 |
|---|---|---|
| | 调用 explore(3, 2) | 开始测试 |
| 调用 modify(0) | | 对 0 号洞穴做操作 1 |
| 调用 query(2) | 返回 0 | 目前 2 号洞穴的光源状态是关闭 |
| 调用 report(0, 1) | | 发现了通路 $(0,1)$ 并记录 |
| 调用 check(0) | 返回 1 | 与 0 号洞穴相关的通路都已被记录 |
| 调用 report(2, 1) | | 发现了通路 $(2,1)$ 并记录 |
| 运行结束并返回 | 向屏幕打印 Correct | 交互结束，结果正确 |

## 【下发文件说明】

在本试题目录下：

1. grader.cpp/c 以及 graderhelperlib.pas 是我们提供的交互库参考实现。
2. explore.h 和 grader.pas 是头文件，选手不用关心具体内容。
3. template_explore.cpp/c/pas 是我们提供的样例解题源代码。
4. explore1.in、explore2.in、explore3.in 是样例输入，可供测试。

**选手注意对所有下发文件做好备份。评测只收取本试题目录下的 explore.c/cpp/pas，并且对该程序以外的文件的修改无效。**

## 【评分方式】

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

**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。

在上述条件基础上，在一个测试点中，你得到满分，当且仅当：

1. 你的每次函数调用均合法，且调用 modify、query 和 check 的次数分别不超过 $L_m, L_q, L_c$。
2. 由于 report 的调用次数限制为 $M$，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 $x$ 号洞穴和 $y$ 号洞穴的通路，且在这次调用之前从未调用过 report(x, y) 或 report(y, x)。
3. 你实现的函数 explore 正常返回。
4. 在 explore 函数返回时，你已经通过调用 report 记录了全部 $M$ 条通路。

## 【数据范围与提示】

本题共 25 个测试点，每个测试点 4 分。每个测试点的数据规模和相关限制见下表。

<table>
<tr><th>测试点编号</th><th>$N=$</th><th>$M=$</th><th>$L_m=$</th><th>$L_q=$</th><th>$L_c=$</th><th>特殊性质</th></tr>
<tr><td>1</td><td>3</td><td>2</td><td>100</td><td>100</td><td>100</td><td rowspan="5">无</td></tr>
<tr><td>2</td><td>100</td><td rowspan="4">$10N$</td><td rowspan="2">200</td><td>$10^4$</td><td rowspan="4">$2M$</td></tr>
<tr><td>3</td><td>200</td><td>$4\times10^4$</td></tr>
<tr><td>4</td><td>300</td><td>299</td><td>$9\times10^4$</td></tr>
<tr><td>5</td><td>500</td><td>499</td><td>$1.5\times10^5$</td></tr>
<tr><td>6</td><td>59998</td><td rowspan="3">$N/2$</td><td>$17N$</td><td>$17N$</td><td rowspan="5">0</td><td rowspan="3">A</td></tr>
<tr><td>7</td><td>99998</td><td>$18N$</td><td>$18N$</td></tr>
<tr><td>8</td><td rowspan="2">199998</td><td rowspan="2">$19N$</td><td rowspan="2">$19N$</td></tr>
<tr><td>9</td><td></td></tr>
<tr><td>10</td><td>99997</td><td rowspan="8">$N-1$</td><td>$18N$</td><td>$18N$</td><td rowspan="2">B</td></tr>
<tr><td>11</td><td>199997</td><td>$19N$</td><td>$19N$</td></tr>
<tr><td>12</td><td>99996</td><td rowspan="9">$10^7$</td><td rowspan="9">$10^7$</td><td rowspan="3">C</td></tr>
<tr><td>13</td><td rowspan="2">199996</td></tr>
<tr><td>14</td></tr>
<tr><td>15</td><td rowspan="2">99995</td><td rowspan="3">D</td></tr>
<tr><td>16</td></tr>
<tr><td>17</td><td>199995</td></tr>
<tr><td>18</td><td rowspan="3">1004</td><td>2000</td><td rowspan="3">$5\times10^4$</td><td rowspan="8">$2M$</td></tr>
<tr><td>19</td><td rowspan="2">3000</td></tr>
<tr><td>20</td></tr>
<tr><td>21</td><td>50000</td><td rowspan="2">$2N$</td><td rowspan="4">$10^7$</td><td rowspan="5">无</td></tr>
<tr><td>22</td><td>100000</td></tr>
<tr><td>23</td><td>150000</td><td>200000</td></tr>
<tr><td>24</td><td rowspan="2">200000</td><td>250000</td></tr>
<tr><td>25</td><td>300000</td></tr>
</table>

再次提醒，题目保证测试所使用的图**在交互开始之前已经完全确定，而不会根据和你的程序的交互动态构造**。

表中特殊性质栏中变量的含义如下：

A：保证每个点的度数恰好为 1。

B：保证对于每个 $x>0$，存在恰好一个 $y<x$ 的 $y$ 使得 $x$ 号洞穴与 $y$ 号洞穴有通路直接相连。

C：存在 $0 \sim N-1$ 的一个排列 $p_0, p_1, \cdots, p_{N-1}$，使得对任意 $1 \le i < N$，存在一条连接洞穴编号分别为 $p_{i-1}$ 与 $p_i$ 的通路。

D：保证图连通。

- 提示：你的程序可以通过判断传入的 $N$ 的个位来区分上述不同的数据类型。