

NOI 2018 模拟题 E

C_SUNSHINE

题目名称	Effulgence	Elasticity	Equipoise
源文件名	effulgence	elasticity	equipoise
输入文件名	effulgence.in	elasticity.in	equipoise.in
输出文件名	effulgence.out	elasticity.out	equipoise.out
时间限制	2s	2s	2s
评分点数目	3	5	6
空间限制	666MB	666MB	666MB
题目类型	交互	传统	传统

注意事项：

1. 竞赛时间为 5 小时，比赛过程中不提供提交反馈，成绩以赛后统一评测为准。
2. 由于评测机配置可能不同，时间限制至少为标程最慢测试点运行时间的 1.5 倍上取整。
3. 最终评测时，打开 -O2 优化，C++ 语言使用 C++11 标准（编译参数加上 `--std=c++11`）。
4. C/C++ 语言下 64 位带符号整形请使用 `%lld` 输出。
5. 祝选手们比赛愉快。

1 Effulgence

1.1 问题描述

本题是交互题，仅支持 C++ 语言。

Lyra 和 Evan 喜欢周游世界，有时他们会向对方炫耀自己去过哪些国家，和在这些国家的见闻。一般来说这是令人愉悦的交谈，但为了凸显自己经历的丰富，Lyra 和 Evan 在描述时会带上夸张的成分，这就导致了，如果他们恰巧去过同一个国家，会因为维护自己吹得牛哔而吵起来……

Lyra 喜欢百花齐放，不收沾染的地域文化，而 Evan 着眼于文化之间密切的交流，所以 Lyra 访问的国家，没有任何一对国家间有航线链接，而 Evan 访问的国家，任意一对国家都有航线直接可达。更为令人费解的是，无论是 Lyra 还是 Evan，关于自己去过哪些国家，在其他人面前都守口如瓶。在热衷于八卦的你百般打听之下，他们决定用告诉他们去过的国家的编号——当然这编号是他们自己造的，而你跟不知道编号是怎么对应的。

然而，热衷于八卦的你才对 Lyra 和 Evan 去过哪些国家不感兴趣，你只关心他们会不会吵起来，你连忙比较 Lyra 和 Evan 告诉你去过的国家的编号集合，却发现，Lyra 和 Evan 使用的根本就是两套编号系统。

Lyra 和 Evan 决定告诉你不超过 K 对编号之间的对应关系，而你想从这消息中判断他们是否会吵架。

出于好心，Evan 和 Lyra 会各给你一份航线图，当然国家是用他们自己的方式编号的，编号都从 $1 \sim n$ ，但两人的编号方式并不相同。

1.2 交互说明

你需要在文件中包含 `effulgence.h`，即在文件头部写上：`#include "effulgence.h"`

你需要编写一个函数：

```
bool QuarrelOrNot(int n, std::vector<int> lyra_set, std::vector<int> evan_set);
```

其中 n 表示国家的数目，`lyra_set` 和 `evan_set` 分别表示两人访问过得国家的集合（自己的编号系统中，没有重复），你可以调用一些函数来获取两张航线图，再询问一些顶点之间的对应关系。

你的函数可以使用以下函数：

`std::pair<int, int> GetNextFlightFromLyra();` 会返回一个 `std::pair<int, int>` 表示一条 Lyra 地图上的双向航线，当返回 `(0,0)` 时，说明 Lyra 地图上的航线已添加完毕。

`std::pair<int, int> GetNextFlightFromEvan();` 会返回一个 `std::pair<int, int>` 表示一条 Evan 地图上的双向航线，当返回 `(0,0)` 时，说明 Evan 地图上的航线已添加完毕。

`int FromLyraToEvan(int id);` 会返回一个整数表示在 Lyra 的地图上编号为 `id` 的国家在 Evan 的地图上的编号。

`int FromEvanToLyra(int id);` 会返回一个整数表示在 Evan 的地图上编号为 `id` 的国家在 Lyra 的地图上的编号。

注意你的程序只有接收完毕 Lyra 和 Evan 地图上的所有航线才可以调用询问对应关系的两个函数，否则会被判错误。

你的程序可以在任意时刻返回一个答案，`true` 表示两个人访问过同一个国家，而 `false` 表示两个人访问的国家集合不相交。

数据不会有重边和子环。在一张地图上的航线已添加完毕之后继续调用返回航线的函数会继续返回 $(0, 0)$ 。

1.3 交互库输入格式

第一行四个正整数 n, m, nL, nE ，表示国家数，双向航线数，Lyra 访问过得国家数，Evan 访问过的国家数。

第二行 n 个整数为一个排列 p_i 表示 Lyra 地图上编号 i 的国家在 Evan 的地图上编号为 p_i 。

第三行 nL 个整数表示 Lyra 访问过的国家编号。

第四行 nE 个整数表示 Evan 访问过得国家编号，注意这里的编号是在原图上（即 Lyra 的编号系统）。

接下来 m 行，每行两个整数表示一条双向航线的两个端点。

1.4 交互库输出格式

若非正常结束交互库会输出错误信息。

当调用询问对应关系的函数吵过 10^6 次后交互库会返回错误。

否则第一行会输出一个字符串 **Correct!** 表示正确 **Wrong!** 表示错误。

若正确则第二行输出一个数 K 表示你使用的询问次数，接下来 K 行每行描述你调用的询问对应关系的函数记录。

1.5 样例输入输出

1.5.1 样例输入 1

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

1.5.2 样例输出 1

Correct!

3

FromLyraToEvan(1) = 1

FromLyraToEvan(2) = 2

FromLyraToEvan(3) = 3

1.5.3 样例输入 2

见 /effulgence/ex_effulgence2.in

1.5.4 样例输出 2

见 /effulgence/ex_effulgence2.ans

1.6 数据范围与约定

对于全部数据 $1 \leq n \leq 1000; 1 \leq nL, nE \leq 1000; 0 \leq m \leq \frac{n(n-1)}{2}$.

详细测试点和部分分信息如下表所示：

子任务	n	分值	30% 要求	60% 要求	100% 要求
1	≤ 15	20	$K \leq 20$	$K \leq 10$	$K \leq 5$
2	≤ 100	30	$K \leq 50$	$K \leq 15$	$K \leq 8$
3	≤ 1000	50	$K \leq 40$	$K \leq 15$	$K \leq 10$

提示：数据中可能存在高度对称的图（即可能有指数级数目的不同映射），所以尽可能不要使用任何图同构近似算法。

1.7 如何测试你的程序

下发文件中有 `grader.cpp` 和 `effulgence.h`，将你的源程序和这两个文件放在同一文件夹下并运行：

```
g++ -o effulgence grader.cpp effulgence.cpp -O2 --std=c++11
```

即可编译你的代码。

2 Elasticity

Lyra 最近收到了一个新的礼物——弹性项链。这个项链由一圈 n 个珠子组成，每个珠子有一个魔力值，魔力值是一个整数，越高则珠子的能量越强。Lyra 不希望自己的项链里有没用的珠子，所以她想通过一些手段把珠子的魔力值都变成非负整数。

由于这些珠子都是有弹性的，对于一个珠子 i ，设其魔力值为 a_i ，Lyra 只要用魔力积压它，它的魔力值就会变成 $-a_i$ ，与此同时与它相邻的两个珠子的魔力值都会增加 a_i 。Lyra 尝试了很久，但她发现她积压很多珠子并没有起到什么效果甚至产生了更多魔力值为负数的珠子。

Lyra 非常生气于是她来找你，想让你告诉她至少需要积压几颗珠子才能让所有的珠子的魔力值都为非负整数呢。

2.1 输入格式

数据包含多组输入，对于每组输入：

第一行一个正整数 n 表示珠子的数目，

第二行 n 个整数 a_i 按次序给出每颗珠子初始的魔力值。

输入以 $n = 0$ 结束。

2.2 输出格式

对于每组数据输出一行一个整数表示最少积压次数，若永远无法满足要求输出 -1 。

2.3 样例输入输出

2.3.1 样例输入 1

```
3
3 -2 1
4
4 -3 4 -1
6
1 -2 3 -4 5 -6
0
```

2.3.2 样例输出 1

```
2
2
-1
```

2.4 数据范围与约定

对于全部数据 $2 \leq n \leq 10^5$; $|a_i| \leq 10^5$; $|\sum a_i| \leq 10$.

Subtask 1[11pts]:

$$n \leq 6.$$

Subtask 2[17pts]:

$$n \leq 20.$$

Subtask 3[15pts]:

$$n \leq 50.$$

Subtask 4[22pts]:

$$n \leq 1000.$$

Subtask 5[35pts]:

$$n \leq 10^5.$$

3 Equipoise

3.1 问题描述

Lyra 对图的对称性很有研究，她格外喜欢一类对称性很强的图，记为 D_k ，其中 k 是一个非负整数， D_k 是一个有 2^k 个节点的图，若把节点编号为 $0 \sim 2^k - 1$ ，则两个节点之间连边当且仅当两个节点的编号的异或是某 2 的次幂。

Lyra 还特别喜欢研究图的乘积，记 $G_1 = (V_1, E_1), G_2 = (V_2, E_2)$ ，则 $G_1 \times G_2$ 定义为

$$(V_1 \times V_2, \{((u_1, u_2), (v_1, v_2)) \in (G_1 \times G_2)^2 | u_1 = v_1 \wedge (u_2, v_2) \in E_2 \vee u_2 = v_2 \wedge (u_1, v_1) \in E_1\})$$

即对点做笛卡尔积后，两个新点之间连边当前仅当其中一个点在对应原图中相同，另一个点在原图中有直接连边。

Lyra 曾经有一个 Evan 送给她的无向图 G 可惜她把图弄丢了，现在 Lyra 只能通过以前的实验数据来推测 G 。

Lyra 有一个图 H ，且已知其同构于 $G \times D_k$ ，并且可以假设不存在任何大于零的整数 k' 和无向图 G' 使得 $G' \times D_{k'}$ 同构与 G 。

Lyra 想请你还原出 G 并向她证明这确实是一个可行的 G ，为此，你要找出可行的 k 并对 H 进行重标号。

3.2 输入格式

第一行两个整数 n, m 表示点数和边数。

接下来 m 行每行两个整数 u, v 表示一条无向边。

3.3 输出格式

第一行输出一个整数 k 表示找到的 k 。

接下来 n 行，每行一个整数 t_i 和一个长度为 k 的 01 串 s_i 。

要求：

每个 $[1, \frac{n}{2^k}]$ 内的整数都要在 t_i 中出现恰好 2^k 次。

每个长度为 k 的 01 串要在 s_i 中出现恰好 $\frac{n}{2^k}$ 次。

(t_i, s_i) 有序对不存在重复。

存在一个点数 $\frac{n}{2^k}$ 的图 $G = ([1, \frac{n}{2^k}], E)$ ，对于任意两个点 $u, v: u, v$ 之间有边当且仅当 $(t_u, t_v) \in E \wedge s_u = s_v$ 或 $t_u = t_v \wedge s_u \oplus s_v = 2^p$ 。（ p 为任意非负整数， \oplus 表示异或）。

3.4 样例输入输出

3.4.1 样例输入 1

4 4

1 2

3 4

1 4
2 3

3.4.2 样例输出 1

2
1 00
1 01
1 11
1 10

3.4.3 样例输入 2

6 9
1 2
2 3
3 1
4 5
5 6
6 4
1 4
2 5
3 6

3.4.4 样例输出 2

1
1 0
2 0
3 0
1 1
2 1
3 1

3.4.5 样例输入 3

见 /equipoise/ex_equipoise3.in

3.4.6 样例输出 3

见 /equipoise/ex_equipoise3.ans

3.5 数据范围及约定

对于全部数据 $1 \leq n, m \leq 2 \times 10^5$.

令 K 表示答案中可行的 k 。

Subtask 1[9pts]:

$$n \leq 20.$$

Subtask 2[13pts]:

$$n \leq 100.$$

Subtask 3[10pts]:

$$n = 2^K \leq 1000.$$

Subtask 4[25pts]:

$$n \leq 1000.$$

Subtask 5[10pts]:

$$n = 2^K \leq 10^5.$$

Subtask 6[33pts]:

$$n \leq 10^5.$$