

Day 1

题目名称	蒙德	璃月	稻妻	须弥
可执行文件名	mondstadt	liyue	inazuma	sumeru
输入文件名	mondstadt.in	liyue.in	inazuma.in	sumeru.in
输出文件名	mondstadt.out	liyue.out	inazuma.out	sumeru.out
每个测试点时限	1 秒	1 秒	1 秒	1 秒
内存限制	256MB	256MB	256MB	256MB
测试点数目	20	20	20	10
每个测试点分值	5	5	5	10
是否有部分分	否	否	否	否
题目类型	传统型	传统型	传统型	传统型
是否有样例文件	否	否	否	否
是否有附加文件	否	否	否	否

提交源程序须加后缀

对于 C++ 语言	mondstadt.cpp	liyue.cpp	inazuma.cpp	sumeru.cpp
-----------	---------------	-----------	-------------	------------

编译开关

对于 C++ 语言	-O2 -lm	-O2 -lm	-O2 -lm	-O2 -lm
-----------	---------	---------	---------	---------

蒙德(Mondstadt)

【问题描述】

在自由的城邦，人们晚上喜欢仰望着星空，尽管那是虚假的。

伟大的占星术士莫娜，绘制了提瓦特天空的无向星图。在星图中，一颗 k 大的星星指的是星图上一个拥有 $k+1$ 个点， k 条边的有向子图 ($k \geq 2$)，且所有的边的起点都是一个公共中心。那个公共中心便是星星的中心，而其余节点均为星星的外沿。

莫娜想要计算出无向星图上大小为 k 的星星的数量 cnt_k ，但是由于输出过于巨大，莫娜只想要知道 $\bigoplus_{k=2}^{n-1} cnt_k = cnt_2 \oplus cnt_3 \oplus \dots \oplus cnt_{n-1}$ (\oplus 表示异或)

【输入格式】

输入文件为 *mondstadt.in*。

输入文件的第一行包含两个正整数 n, m ，用空格隔开，表示星图节点数和边数。

接下来 m 行，包含用空格隔开的两个整数 u_i 和 v_i ，为星图中的一条边，保证 $u_i < v_i$ 。

【输出格式与部分分】

输出文件为 *mondstadt.out*。

输出仅一行一个整数，表示 $\bigoplus_{k=2}^{n-1} cnt_k$ 。

【样例 1 输入】

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

【样例 1 输出】

```
1
```

【样例 2 输入】

```
4 6
```

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

【样例 2 输出】

8

【样例 2 说明】

一共有 12 个大小为 2 的星和 4 个大小为 3 的星， $12 \oplus 4 = 8$ 。

【子任务】

所有测试数据的范围和特点如下：

对于 20%的数据有 $1 \leq n \leq 10, 1 \leq m \leq 20$

对于 40%的数据有 $1 \leq n \leq 200, 1 \leq m \leq 400$

对于 60%的数据有 $1 \leq n \leq 5000, 1 \leq m \leq 20000$ 。

对于 80%的数据有 图为 $n - 1$ 大小的星图

对于 100%的数据有 $1 \leq n \leq 1000000, 1 \leq m \leq 2000000$

璃月(Liyue)

【问题描述】

在岩的国度，人们喜欢正方的事物，比如完全平方数。

月海亭的秘书甘雨打算处理今日事务，她将今日的事务从 1 到 n 编号，然后打乱，打乱后的第 i 件事编号为 a_i 。每一次，她会从打乱后的事务中抽出连续的一摞，然后从中选择两个编号之和为完全平方数的事务处理。

甘雨想要知道，假如她抽出的连续的一摞是打乱之后的第 L 到 R 件事务，她有多少种选择两个编号之和为完全平方数的事务的方法。

【输入格式】

输入文件为 *liyue.in*。

输入文件的第一行包含一个正整数 n 。

接下来一行包含 n 个整数，表示打乱后的事务编号。

接下来一行包含一个正整数 q 表示询问次数。

接下来 q 行包含两个正整数 L, R ，表示甘雨这次抽出的是第 L 到 R 件事务 ($L \leq R$)。

【输出格式与部分分】

输出文件为 *liyue.out*。

输出 q 行,每行一个整数，该次询问甘雨可选择事务的方案数。

【样例 1 输入】

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

3 7

【样例 1 输出】

1
1
5
2
3
3
1
2
1
1

【子任务】

所有测试数据的范围和特点如下：

对于 30%的数据有 $1 \leq n \leq 500, 1 \leq q \leq 500$

对于 50%的数据有 $1 \leq n \leq 5 \times 10^3, 1 \leq q \leq 10^5$ 。

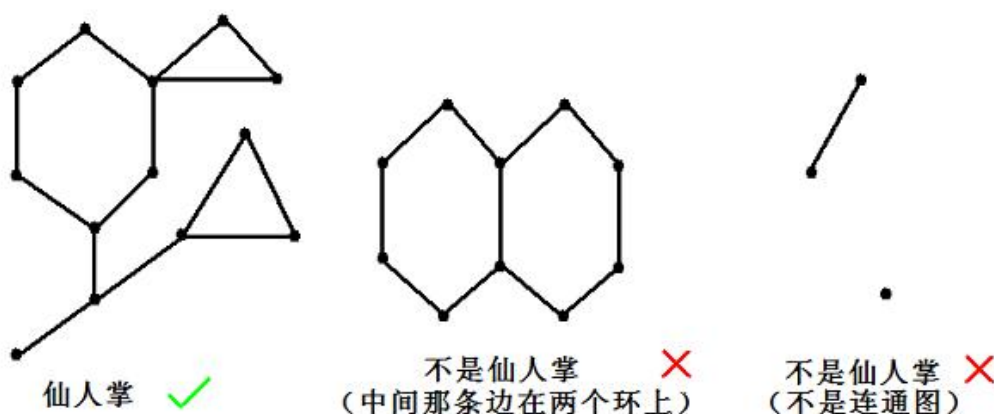
对于额外 20% 的数据有甘雨不会将今日事务打乱

对于 100%的数据有 $1 \leq n \leq 10^5, 1 \leq q \leq 10^5$

稻妻(Inazuma)

【问题背景】

在提瓦特，司掌雷电力量的魔神是来自稻妻鸣神岛的御建鸣神主尊大御所大人。在雷电的国度，电力供应由一个仙人掌形状的输电系统维持。如果一个无向连通图的任意一条边最多属于一个简单环，我们就称之为仙人掌。所谓简单环即不经过重复的结点的环。



然而，电线并不是永恒的，每根电线 $e_i(u_i, v_i)$ 都有其寿命 d_i 。向往永恒的魔神并不能改变万物的寿命，她只能控制每根电线的启用时间 s_i 。若一根电线的启用时间为 s_i ，则在时间段 $[s_i, s_i + d_i]$ 内，该电线可以传输电力，其余时间改电线不能传输电力。

整个输电系统只有一个建在 1 号节点的稻妻鸣神岛的发电站，御建鸣神主尊大御所大人想知道，这个输电系统最长能令所有节点拥有电力多长时间。

【输入格式】

输入文件为 *inazuma.in*。

输入文件的第一行包含两个正整数 n, m ，用空格隔开，表示节点数和边数。

接下来 m 行，每行表示一根电线，包含用空格隔开的三个整数 u_i, v_i 和 d_i ，为电线的两个端点和寿命。

【输出格式】

输出文件为 *inazuma.out*。

输出文件只有一行，包含一个正整数，即输电系统正常工作的最长时间。

【样例 1 输入】

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

【样例 1 输出】

```
2
```

【样例 1 说明】

你可以设置 $s_1 = s_3 = 0; s_2 = 1$, 这样所有的节点在时间区间 $[0; 1]$ 内都会通过边 1 和边 3 连接, 而在时间区间 $[1; 2]$ 内则通过边 2 和边 3 连接。

【样例 2 输入】

```
5 6
1 2 3
1 2 2
2 3 3
2 4 5
2 5 5
3 4 2
```

【样例 2 输出】

```
5
```

【子任务】

所有测试数据的范围和特点如下:

对于 20%的数据有 $1 \leq n \leq 10, 1 \leq m \leq 20$

对于 40%的数据有 $1 \leq n \leq 200, 1 \leq m \leq 400$

对于 70%的数据有 $1 \leq n \leq 10^5, 1 \leq m \leq 2 \times 10^5$

对于 100%的数据有 $1 \leq n \leq 10^6, 1 \leq m \leq 2 \times 10^6, 1 \leq d_i \leq 10^6$

须弥(Sumeru)

【问题背景】

智慧之神纳西妲经常为了封印五百年前的漆黑灾厄，智慧消耗过度（谁是大慈树王？）。现在，她遇到了连智慧之神都无法解决的问题：

对于四个三维向量 A_1, A_2, A_3, A_4 ，是否存在三个非负数 x_1, x_2, x_3 ，使得 $x_1 A_1 + x_2 A_2 + x_3 A_3 = A_4$ 。

$$\text{即} \begin{cases} x_1 a_{11} + x_2 a_{21} + x_3 a_{31} = a_{41} \\ x_1 a_{12} + x_2 a_{22} + x_3 a_{32} = a_{42}, \text{ s.t. } x_1, x_2, x_3 \geq 0 \\ x_1 a_{13} + x_2 a_{23} + x_3 a_{33} = a_{43} \end{cases}$$

【输入格式】

输入文件为 *sumeru.in*。

输入文件的第一行包含个正整数 T ，表示询问次数。

接下来 T 行，每行 12 个整数，其中包含

$a_{11}, a_{12}, a_{13}, a_{21}, a_{22}, a_{23}, a_{31}, a_{32}, a_{33}, a_{41}, a_{42}, a_{43}$ ，表示 A_1, A_2, A_3, A_4 。

【输出格式】

输出文件为 *sumeru.out*。

输出文件有 T 行，每行输出 YES 或者 NO，表示是否存在 x_1, x_2, x_3 满足条件。

【样例 1 输入】

```
2
3 4 4 4 3 0 2 3 2 9 10 6
0 3 1 0 1 3 4 0 4 4 1 10 3 3
```

【样例 1 输出】

```
YES
NO
```

【子任务】

所有测试数据的范围和特点如下：

对于 30% 的数据有 $0 \leq a_{ij} \leq 4$

对于 40%的数据有 A_1, A_2, A_3 线性无关。

对于 100%的数据有 $0 \leq a_{ij} \leq 10^4, T \leq 1000$