

冲刺NOIP2023模拟试题

时间：7:40-12:10

(请选手务必仔细阅读本页内容)

一、题目概况

中文题目名称	圆	网络流	过河卒	计算器
英文题目与子目录名	circle	flow	soldier	counter
可执行文件名	circle	flow	soldier	counter
输入文件名	circle.in	flow.in	soldier.in	counter.in
输出文件名	circle.out	flow.out	soldier.out	counter.out
每个测试点时限	1.0秒	1.0秒	1.0秒	3.0秒
测试点数目	Subtask	Subtask	Subtask	20
测试点是否等分	否	否	否	否
附加样例文件	有	有	有	有
结果比较方式	全文比较	全文比较	全文比较	全文比较
题目类型	传统型	传统型	传统型	传统型
运行内存上限	512 MB	512MB	512 MB	256 MB

二、提交源程序文件名

对于C++语言	circle.cpp	flow.cpp	soldier.cpp	counter.cpp
---------	------------	----------	-------------	-------------

三、编译选项

对于C++语言	-lm -std=c++14 -O2
---------	--------------------

四、注意事项

- 1、文件名（程序名和输入输出文件名）必须使用英文小写。
- 2、C/C++中函数 main()的返回值类型必须是 int，程序正常结束时的返回值必须是 0。
- 3、全国统一评测时采用的机器配置为：Intel(R) Core(TM) i7-8700K CPU @ 3.70GHz，内存 32GB。上述时限以此配置为准。
- 4、只提供 Linux 格式附加样例文件。
- 5、特别提醒：评测在当前最新公布的 NOI Linux 下进行，各语言的编译器版本以其为准。

1.圆 (circle)

【题目描述】

你现在有一个长为 $3n$ 的圆环。圆环上分布着 $3n$ 个点，这些点总共分为 n 组。每个点都会被分到恰好一组里面，且每一组里面恰好 3 个点。

你想要知道，如果在每一组里面选恰好两个点，然后在这两个点间在圆上画一条不经过这一组中没被选中的那个点的圆弧(圆环上的一个区间)，那么有多少种方案使得每对圆弧没有任何相交的部分？

由于显然的原因，答案对 $10^9 + 7$ 取模。

【输入格式】

从文件 circle.in 中读取数据。

第一行一个正整数 n 。

接下来一行 $3n$ 个正整数，第 i 表示第 i 个点所属的组的编号，保证编号在 $[1, n]$ 。

【输出格式】

输出到文件 circle.out 中。

一行一个整数表示答案。

【样例输入1】

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

【样例输出1】

```
8
```

【样例2,3】

见下发文件中。

【数据范围及约定】

对于所有数据： $n \leq 10^6$ 。

Subtask1(10%)： $n \leq 14$ 。

Subtask2(20%)： $n \leq 20$ 。

Subtask3(30%)： $n \leq 1000$ 。

Subtask4(20%)： $n \leq 100000$ 。

Subtask5(20%)： $n \leq 1000000$ 。

由于输入输出量较大，建议使用较快的读入输出方式。下发文件提供了出题人的 *iobuff* 供选手使用。注意，提供的 *iobuff* 程序在本机可能必须要开启文件操作才能正确运行。

2. 网络流 (flow)

【题目描述】

现在平面上有 n 个点围成了一个凸包，并且按照逆时针顺序依次标号为 $1, 2, 3, \dots, n$ 。

这些点之间总共有 m 条双向边，每条边有一个边权。其中第 i 号点与第 $i + 1$ 号点直接一定有边（如果 $i = n$ 则 n 号点与 1 号点之间有边）。所有边保证如果将其画为一条两端点间的直线段则任意两条边在除了端点的地方不交。

现在定义 $flow(s, t)$ 表示如果将边权视作边的容量，从 s 出发到 t 的最大流大小。你需要求出 $\sum_{s < t} flow(s, t)$ 。

由于答案可能很大，因此对 1000000007 取模。

【输入格式】

从文件 flow.in 中读取数据。

第一行两个正整数 n, m 。

接下来 m 行每行三个正整数 u, v, w ，表示有一条连接 u, v 的权值为 w 的边， $0 \leq w \leq 10^9$ 。

【输出格式】

输出到文件 flow.out 中。

一行一个数表示答案。

【样例输入1】

```
10 16
1 2 5
2 3 9
3 4 8
4 5 1
5 6 6
6 7 3
7 8 1
8 9 5
9 10 3
10 1 8
1 4 3
1 3 3
1 6 9
1 5 10
6 9 9
6 10 1
```

【样例输出1】

【样例2,3】

见下文件。

【数据范围及约定】

对于所有数据: $n \leq 5 \times 10^5, m \leq 2n - 3$ 。

Subtask1(20%): $n \leq 50$ 。

Subtask2(30%): $n \leq 200$ 。

Subtask3(20%): $n \leq 1000$ 。

Subtask4(30%): $n \leq 5 \times 10^5$ 。

3.过河卒 (soldier)

【题目描述】

给定一张包含 n 个顶点和 m 条边的 **无向连通图**，Tom 和 Jerry 在图上进行了 q 次追逐游戏。

在第 i 次游戏中，Tom 一开始位于顶点 a_i ，而 Jerry 一开始位于顶点 b_i （双方任何时候都知道自己和对方的位置），追逐规则如下：

- Jerry 和 Tom 交替行动，Jerry 先行动。
- Jerry 每次行动可以通过无向图中的 **任意多条边**（可以选择不移动），但是在移动过程中不能经过 Tom 当前所在的结点，否则就会被抓住。
- Tom 每次行动只能通过无向图中的 **至多一条边**（可以选择不移动）。
- 如果 Tom 在一次行动后到达了 Jerry 的位置，那么 Tom 胜利。

Tom 尽量想要胜利，而 Jerry 会尽量阻止 Tom 胜利。

现在你需要对于每一局游戏，求出 Tom 是否一定能在有限次行动内获胜。

【输入格式】

从文件 soldier.in 中读取数据。

第 1 行：三个整数 n, m, q ，分别表示无向连通图的点数，边数以及游戏的次数。

接下来 m 行：每行两个整数 x, y ，描述图中的一条无向边。

接下来 q 行：每行两个整数 a, b ，表示一局游戏中双方的初始位置。

【输出格式】

输出到文件 soldier.out 中。

共 q 行：对于每局游戏，如果 Tom 可以在有限个回合内获胜则输出 Yes，否则输出 No。

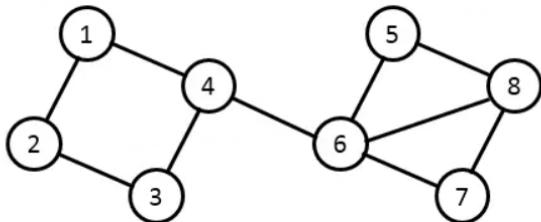
【样例输入1】

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

【样例输出1】

No
Yes
No

【样例1解释】



第一组询问中， $a_1 = 6, b_1 = 4$ ，则 Jerry 先走到 2 处，此后每一回合，若 Tom 行动完后与 Jerry 相邻，Jerry 只需要移动到环 $[1, 2, 3, 4]$ 中与 Tom 不相邻的那个点，可保证 Tom 不胜。

第二组询问中， $a_2 = 4, b_2 = 5$ ，无论 Jerry 如何行动，Tom 只需走到 6 处，此后 Jerry 可能在 $\{5, 7, 8\}$ ，无论如何 Tom 都可以一步追到。

第三组询问中， $a_3 = 5, b_3 = 7$ ，则 Jerry 按照第一组询问中的策略即可使得 Tom 无法获胜。

【样例2】

见下文件。

【数据范围及约定】

本题使用捆绑测试，子任务信息如下：

子任务编号	$n, m, q \leq$	特殊性质	分值
1	10	无	10
2	100	无	16
3	1000	无	24
4	10^5	$m = n$	16
5	10^5	无	34

对于 100% 的数据， $1 \leq n, m, q \leq 10^5$ ， $1 \leq x, y, a, b \leq n$ ， $a_i \neq b_i$ 。

保证给出的无向图连通，且不含重边和自环。

4.计算器 (counter)

【题目描述】

你有一个奇怪的计数器，计数器上有一个数字 x ，每次你可以做如下操作：

1. 选择一个 $x = b_k b_{k-1} \dots b_1 b_0{}_{(10)}$ 中的一个数位 b_i 。
2. 将 x 变为 $x + b_i$ 或者 $x - b_i$ 。

例如，当 $x = (616)_{10}$ 时，你可以选择 $b = 1$ ，然后让 x 变为 $x - b = (615)_{10}$ 。

你想要通过最少步数把 x 变成 y ，问最少步数是多少。

【输入格式】

从文件 counter.in 中读取数据。

第一行一个整数 T ，代表数据组数。

每组数据包含一行两个整数 x', y' ，令上一组数据答案为 $lastans$ ，初始时 $lastans = 0$ ，则该组数据为 $x = x' \oplus (lastans + 1)$, $y = y' \oplus (lastans + 1)$ 。

【输出格式】

输出到文件 counter.out 中。

对于每组询问，输出一个数字代表答案。如果无法从 x 变到 y ，则输出 -1 。

【样例输入1】

```
3
2 6
12 0
4 7
```

【样例输出1】

```
-1
3
-1
```

【样例1解释】

$x = 3, y = 7$ ，无解。

$x = 12, y = 0$ ，最优操作方案为 $12 \rightarrow 10 \rightarrow 9 \rightarrow 0$ 。

$x = 0, y = 3$ ，无解。

【样例2】

见下发文件中。

【数据范围及约定】

对于 10% 的数据， $T \leq 10, 0 \leq x, y \leq 10$ 。

对于另外 20% 的数据， $0 \leq x, y \leq 2 \times 10^3$ 。

对于另外 30% 的数据， $0 \leq x, y \leq 1 \times 10^4$ 。

对于另外 20% 的数据， $0 \leq x, y \leq 3 \times 10^4$ 。

对于 100% 的数据， $0 \leq T, x, y \leq 1 \times 10^5$ 。