

NOIP2019 模拟赛

EndSaH

题目名称	最大流	拼数	火花
题目类型	传统型	传统型	传统型
目录	flow	piece	fire
源文件名	flow	piece	fire
输入文件名	flow.in	piece.in	fire.in
输出文件名	flow.out	piece.out	fire.out
测试点个数	20	10	20
每个测试点时限	1s	1.5s	1.5s
内存限制	512MB	512MB	512MB
代码长度限制	50KB	50KB	50KB
是否有下发样例	是	是	是
编译命令	-O2 -std=c++11		

注意事项：

1. 题目难度与顺序无关。
2. 认真读题。
3. 评测时开启无限栈。
4. Best wishes!

1 最大流

1.1 题目描述

在战火纷飞，饿殍遍野的黑暗时代，jambow 以其怀天下苍生的仁心和他人难以望其项背的实力，统一了 X501-G 大陆，建立起了塔克国。塔克国最高领袖的称号为“塔克”。

岁月更迭，由于地理位置以及其他种种复杂的原因，jambow 决定迁移首都。

迁移首都很多必要工作，而其中有一项便是重新规划地下水道。

jambow 由于这项工作繁杂且小弟们无一能胜任这项工作，所以决定亲自上阵，展现他作为塔克的实力。

虽然在之前 jambow 还没有研究过这类叫做网络流的问题，但以他的实力， $10^{-16}s$ 内学会当然不在话下。

在学习过程中，jambow 抄了一些笔记：

通常在运筹学中，有向图称为网络，顶点称为节点而边称为弧。

如果带权有限的有向图 $G = (V, E)$ 满足如下条件，则称之为**网络流图**（或**容量网络**）：

1. 有且仅有一个节点 $s \in V$ 入度为 0，称为**源点**。
2. 有且仅有一个节点 $t \in V$ 出度为 0，称为**汇点**。
3. $\forall (u, v) \in E \exists c(u, v) \in R^+$ ，称为这条弧的容量。

我们将通过容量网络中一条弧 (u, v) 的**流量**（或**净流**）记为 $f(u, v)$ 。如果一个流量的集合 $F = f(u, v)$ 包含所有弧上的所有流，则称 F 为这个容量网络的一个**网络流**。

在任意时刻， G 的网络流均满足如下性质。

1. **容量限制**： $f(u, v) \leq c(u, v)$ 。
2. **反对称性**： $\forall u, v \in V, f(u, v) = -f(v, u)$ 。
3. **流守恒**： $\forall u \in V - \{s, t\}$ ，有 $\sum_{v \in V} f(u, v) = 0$ 。

对于一个容量网络，其**最大流**指的是在所有可能的网络流中，汇点收到的流量最大的网络流。

（以上为笔记内容）

jambow 思来想去，在 $5 \times 10^{-19}s$ 后终于得到了一个 $O(n + m)$ 的求解最大流的算法。他觉得这个问题有些过于简单，于是稍微扩充了一下定义：

容量网络：有向图 \rightarrow 无向图，任意两点都可为源点汇点。

弧：若 $(u, v) \notin E$ ，则假定存在一条 (u, v) 的弧，且 $c(u, v) = 0$ 。

过了 $5 \times 10^{-19}s$ 后，**jambow** 又基于扩充定义得到了一个 $O((n + m) \log n)$ 求解任意两点之间最大流的算法。

此时问题已经解决完毕了。**jambow** 准备将他的知识传授给小弟，于是准备先让他们思考如何求解任意两点最大流这个问题再授课。

jambow 很快就意识到这个问题对他的小弟来说有些困难，于是限制了 $\forall u \in V, \deg_u \leq d$, $\forall (u, v) \in E, c(u, v) = 1$ (\deg_i 表 i 点的度数)，并且只需要他们回答所有无序点对间最大流的和。

然而 EndSaH 依然不会求解这个弱化版的问题，又直接不能去问塔克 **jambow**，所以只能来问你了。

1.2 输入格式

从文件 *flow.in* 中读入数据。

第一行为三个正整数 n, m, d ，含义如题目所示。

接下来的 m 行每行有两个正整数 u, v ，描述了这个无向图，保证无自环重边。

1.3 输出格式

输出到文件 *flow.out* 中。

输出一行一个正整数，表示最终答案。

1.4 样例 1

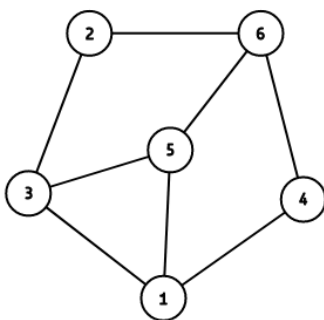
1.4.1 输入

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

1.4.2 输出

36

1.4.3 解释



样例所给出的无向图如图所示。

1.5 样例 2-3

见下发文件。

其中样例 2 满足 $n, m \leq 200$ 。

1.6 数据规模与约定

对于所有数据，满足 $1 \leq n \leq 3000, 1 \leq m \leq 4500, d \in \{1, 2, 3\}, 1 \leq u, v \leq n, u \neq v$ 。

对于前 10% 的数据， $1 \leq n, m \leq 8$ 。

对于前 30% 的数据， $1 \leq n, m \leq 200$ 。

对于另外 20% 的数据， $d = 2$ 。

2 拼数

2.1 题目描述

EndSaH 有 n 个数 a_1, \dots, a_n ，他打算选出这些数中的两个数进行拼接。

一次拼数操作指的是将 x, y 两个正整数视作数字串， x 在前 y 在后拼接成一个新数字串，该新数字串所表示的正整数即这次拼数操作的结果。

例如将 1234 与 56 拼数，将得到结果 123456。

注意拼数操作是有顺序的，如拼接 56 与 1234 会得到 561234。

但不知道为什么，EndSaH 一点也不喜欢 k 的倍数。

他很好奇一个问题：有多少数对 $(i, j) (1 \leq i, j \leq n, i \neq j)$ 满足对 a_i 与 a_j 进行拼数操作后，结果不是 k 的倍数？

由于他太菜了实在不会，只能向你求助了。

2.2 输入格式

从文件 *piece.in* 中读入数据。

第一行为两个正整数 n, k ，含义如题目描述。

第二行为 n 个正整数，第 i 个数为 a_i 。

2.3 输出格式

输出到文件 *piece.out* 中。

输出一行一个正整数，表示满足要求的数对的个数。

2.4 样例 1

2.4.1 输入

```
4 11
45 1 10 12
```

2.4.2 输出

```
5
```

2.4.3 解释

数对 $(2, 1), (3, 2), (2, 4), (1, 4), (4, 1)$ 满足条件。

拼数所得结果分别为 145, 101, 112, 4512, 101，均不是 11 的倍数。

2.5 样例 2-3

见下发文件。

其中样例 2 满足 $a_i < 10^3$ 。

2.6 数据规模及约定

对于所有数据，保证 $1 \leq n \leq 10^5, 1 \leq k, a_i \leq 10^9$ 。

对于前 30% 的数据，保证 $n \leq 1000$ 。

对于前 60% 的数据，保证 $a_i < 10^3$ 。

3 火花

3.1 题目描述

细雨迷蒙中，T_{ea} 在路上散步。忽而，一棵参天大树出现在眼前。

细数一下，可以数出这棵树上 n 个节点。便于描述，我们给每个节点编号，其中 1 号节点为根节点。

T_{ea} 突然想施展自己多年没有释放过的低阶法术——火花术，并想尝试着烧掉这棵树。

若是正常情况，火势会迅速蔓延，将整棵树化作焦炭。但现在，他发现了两件有趣的事情：

1. 刚下了雨，树枝有些湿润。T_{ea} 定义了每根树枝的湿润度 w ，是因为他发现如果他这次用于火花术的法力值小于 w ，那么火花就无法点燃这根树枝；否则，火就从当前节点可以蔓延到这根树枝所指向的那个节点上，并烧掉这根树枝。
2. 无论走到哪里，强者总会散发出气场。由于 T_{ea} 现在站在树下，强大的气场风让火无法向根而只能向上蔓延。

若是燃烧掉了一条树枝，T_{ea} 将会获得与其湿润度相等的成就感。现在，T_{ea} 想问你，若是他用 k 的法力值对节点 u 释放火花术，他得到的成就感是多少。询问之间互不影响。不过，为了确定你没有糊弄他，他有时候会要求你即时回答。

本题读入输出量较大，请使用较快的读入输出方式。

3.2 输入格式

从文件 `fire.in` 中读入数据。

第一行为一个正整数 id ，表示该测试点编号。

第二行为两个正整数 n, T ，其中 T 为强制在线参数。

接下来的 $n-1$ 行每行有两个正整数 fa_i, w_i ，描述了 i 号节点的父亲 fa_i ，以及其到父亲这条边的湿润度 w_i ， i 从 2 开始编号。

再下一行为一个正整数 q ，表示询问的组数。

接下来的 q 行每行有两个正整数 u, k ，描述了一组询问。若 $T=1$ ，定义 $lastans$ 为上一次询问的答案，若当前是第一次询问，则 $lastans=0$ 。你需要将这两个整数异或上 $lastans$ 对 2^{20} 取余的结果，才能得到正确的输入。

3.3 输出格式

输出到文件 `fire.out` 中。

输出共 q 行，每行为一个整数，表示询问的答案。

3.4 样例 1

3.4.1 输入

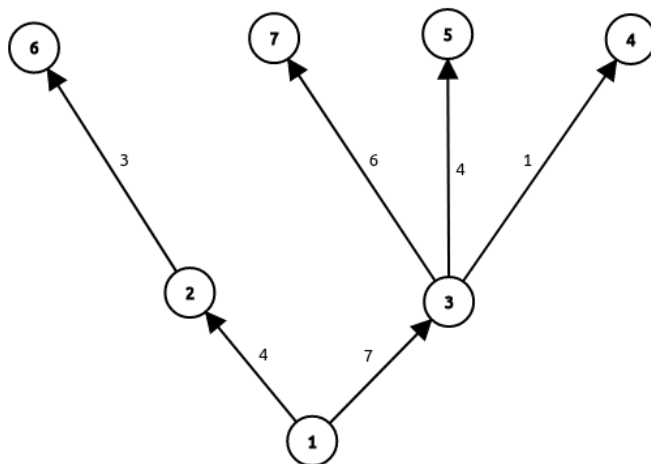
```
0
7 0
1 4
1 7
3 1
3 4
2 3
3 6
5
1 4
3 5
2 7
1 7
1 3
```

3.4.2 输出

```
7
5
3
25
0
```

3.4.3 解释

样例 1 的树如下所示：



第一组询问：被点燃的边有 $(1,2), (2,6)$ 。

第二组询问：被点燃的边有 $(3,5), (3,4)$ 。

第三组询问：被点燃的边有 $(2,6)$ 。

第四组询问：所有边均可以点燃。

第五组询问：无法点燃任何边。

3.5 样例 2-4

见下发文件。

其中样例 2 满足 $n, q \leq 5000, T = 0$ ，样例 3 满足 $fa_i = i - 1, T = 1$ ，样例 4 满足 $n, q \leq 10^5, T = 0$ 。

3.6 数据规模与约定

对于所有数据，保证给出的是一棵树，且满足 $T \in \{0, 1\}, 1 \leq n, q \leq 3 \times 10^5, 1 \leq w_i, k \leq 10^9$ ，此处的 k 指的是真实的询问。

测试点编号	n, q	T	特殊性质
1, 2, 3	$\leq 5 \times 10^3$	0	无
4, 5, 6, 7	$\leq 10^5$	1	
8	$\leq 3 \times 10^5$		$fa_i = 1$
9, 10, 11, 12			$fa_i = i - 1$
13, 14, 15, 16		0	无
17, 18, 19, 20	1		