非专业软件能力认证提高级模拟赛

CZF~2022~Simulation

xdfz

时间2023年10月1日,14:10-18:10

题目名称	Summer Pokemon	追梦人	下围棋	酒罐王
题目类型	传统型	传统型	传统型	传统型
目录	umi	dream	nine	bar
可执行文件名	umi.exe	dream.exe	nine.exe	bar.exe
提交源程序文件名	umi.cpp	dream.cpp	nine.cpp	bar.cpp
输入文件名	umi.in	dream.in	nine.in	bar.in
输出文件名	umi.out	dream.out	nine.out	bar.out
每个测试点时限	3.0秒	2.0秒	2.0秒	4.0秒
内存限制	512MiB	512MiB	\$ \mathrm{MiB}\$	1024MiB
子任务(捆绑)数目	20	25	2	6

编译选项: -O2 -std=c++14

注意事项与提醒(请选手仔细阅读)

- 1. 选手提交的源程序请直接放在个人目录下, 无需建立子文件夹。
- 2. 文件名(程序名和输入输出文件名)必须使用英文小写。
- 3. C++ 中函数 main () 的返回值类型必须是 int, 值必须为 0。
- 4. 因违反以上三点而出现的错误或问题, 申诉时一律不予受理。
- 5. 若无特殊说明, 结果的比较方式为全文比较 (过滤行末空格及文末回车)。
- 6. 选手提交的程序源文件必须不大于 100 KB。
- 7. 程序可使用的栈空间内存限制与题目的内存限制一致。
- 8. 统一评测时采用的机器配置为: Intel Core i7-12700 CPU, 内存 16GB。上述时限以此配置为准。
- 9. 评测在 NOI Linux2.0 下进行, GCC 版本为 9.3.0, 并使用 LemonLime 进行评测。

Summer Pokemon

白鸟亦悲否,不为空之青

海之蓝所染, 弋于其间

白鸟不悲否,苍茫天水间

片羽皆不染, 无言自飘摇

与君同行, 蔚蓝之间, 常夏之岛。

唯有那份炫目,未曾忘却。

题目背景

umi: 这个手镯, 是作为 "岛可梦大乱斗" 参加者的证明。

umi: 所谓岛可梦大乱斗,就是指通过捉捕生活在这座岛上的生物,来互相进行战斗的比赛。

嗨梨: 感觉就像是现实生活中的宝可梦一样?

umi: 算是吧。

umi: 只要在晚上准备好诱饵, 到了早上就能Get到了。

umi:不介意的话,需要我把饵场的地图借给你吗?

嗨梨:岛可梦大乱斗么....

【决定参加】√

【果然还是不参加了】(【警告】如果您选择该选项,您将在此题获得0分)

嗨梨:好,那我就试试吧。

umi: 请注意不要死掉哦。

嗨梨: 欸? 还有可能会死吗?

还真是够危险的。

umi: 我也把自己拥有的诱饵分一些给你吧。

嗨梨: 嗯, 谢了。

【砂糖水】Get!

【西瓜皮】Get!

【蚯蚓】Get!

【昆虫果酱】Get!

【海唔叽】Get!

【究极的炒饭】Get!

【能够成为贤者的H书】Get!

题目描述

【注意:本题中的岛可梦大乱斗与原作有很大出入。如果对题目描述有疑惑,请务必参考样例解释】

嗨梨决定参加鸟白岛的岛可梦大乱斗!

鸟白岛共有 m 个岛可梦,它/他/她们的生命值分别为 $1, 2, \ldots, m$ 。

鸟白岛共有 n 处地点,它们之间由 n-1 条道路连接,每两处地点之间都能互达。记 d(u,v) 表示第 u 个地点到第 v 个地点的最短距离(经过的最少 **点数**)。

嗨梨与 umi 将在鸟白岛的乡间小道上进行对决。一次对决在从 a 到 b 的最短路径上进行,流程如下:

- 1. 双方各从 m 个岛可梦中随机选出 d(a,b) 个岛可梦出战,**两人不能选择相同的岛可梦**。
- 2. 从 a 开始走到 b ,每经过一个地点都会进行一次判定(包含起始点与结束点):
 - 。 双方各取出自己手中最弱的岛可梦, 进行生命值大小判定, 数值大的一方赢得该轮判定的胜利。
 - · 判定结束后, 该轮上场的岛可梦 均 退出本次对决。

受到七影蝶的影响,嗨梨与 umi **在每处地点的胜负情况是确定的**(即在一次对决中,判定结果由该次对决双方的出战岛可梦决定,而你已经知道了每轮判定的结果)。 s_i 表示嗨梨与 umi 在第 i 处地点的胜负情况,若 $s_i=1$ 则表示嗨梨赢,若 $s_i=0$ 表示 umi 赢。

嗨梨与 umi 共进行了 Q 次对决, 每次对决相互独立。

你已经知道了每次对决的路径,也就知道了每次对决的具体胜负情况。你想知道对于每次对决,嗨梨与 umi 出战岛可梦有多少种方案。

输出方案数对 998244353 取模后的结果。

输入格式

第一行三个正整数 n,m,id,表示鸟白岛共有 n 处地点和 m 个岛可梦,测试点编号为 id。如果为样例,则 id=0

接下来一行 n 个整数, 第 i 个整数 s_i 表示嗨梨与 umi 在第 i 个地点的胜负情况。

接下来 n-1 行,每行两个正整数 $u,v(1 \le u,v \le n)$,表示第 u 个地点与第 v 个地点有道路连接。

接下来一行一个正整数 Q,表示嗨梨和 umi 的对决次数。

接下来 Q 行,每行两个整数 $a,b(1 \le a,b \le n)$,表示该次对决的起点为第 a 个地点,终点为第 b 个地点。

输出格式

输出共Q行,每行一个正整数w,分别表示每次对决中嗨梨与umi 出战岛可梦的方案数。

样例 #1

样例输入#1

```
3 6 0
1 1 0
1 2
1 3
2
3 1
2 3
```

样例输出#1

```
15
2
```

样例 #2

样例输入#2

见附件下的 umi2.in。

样例输出#2

见附件下的 umi2.ans。

样例 #3

样例输入#3

见附件下的 umi3.in。

样例输出#3

见附件下的 umi3.ans。

样例 #4

样例输入#4

见附件下的 umi4.in。

样例输出#4

见附件下的 umi4.ans。

样例 #5

样例输入#5

见附件下的 umi5.in。

样例输出#5

见附件下的 umi5.ans。

样例 #6

样例输入#6

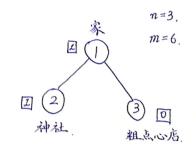
见附件下的 umi6.in。

样例输出#6

见附件下的 umi6.ans。

提示

【样例1解释】



*两人拥有的岛可梦并不确定: 如出战方采 (0 与出战方采 (B), 它们就一定对应看不同的拥有方案

• 3→1: 双方共进行 2 轮判定, 判定结果为: [0]. 共有以下15 种出战方案满足判定结果: (以下简记, 嫌轶为 H, umi 为 U)

- 0 H: 1,4 U: 2,3
- @ H:1,5 U:2,3
- 3 H: 1, 6 U: 2,3
- (H: 1,5 U: 2,4
- 5 H:1,6 U:2,4
- 6 H: 1,6 U: 2,5
- 1 H: 1,5 U: 3,4
- B H: 1,6 U: 3,4
- 9 H: 1,6 U:3,5
- (1) H: 1,6 U:4,5
- (II) H: 2.5 U: 3,4
- (2) H: 2.6 U: 3,4
- (B) H: 2.6 U: 3.5
- (A) H: 2,6 U: 4,5
- (J) H: 3.6 U: 4.5

• 2 → 3, 双方共进行 3 轮判定, 判定结果为: [IIO]. (2→1→3) 共有以下两种出战方案满足判定结果:

O H: 2,4,5 U: 1,3,6

Q. H: 3,4,5 U: 1,2,6

【样例 2】

该样例满足测试点1的限制。

【样例 3】

该样例满足测试点4的限制。

【样例4】

该样例满足测试点 $5\sim 10$ 的限制。

【样例5】

该样例满足测试点13,14的限制。

【样例6】

该样例满足测试点 $15\sim 20$ 的限制。

【数据范围】

对于 100% 的数据,保证 $1 \le n \le 10^5, 2n \le m \le 10^6, s_i \in \{0,1\}$ 。

追梦人

题目背景

让青春吹动了你的长发,让它牵引你的梦 不知不觉这城市的历史,已记取了你的笑容 秋来春去红尘中,谁在宿命里安排 冰雪不语寒夜的你,那难隐藏的光采

让流浪的足迹在荒漠里,写下永久的回忆 飘去飘来的笔迹,是深藏的激情你的心语 前尘后世轮回中,谁在声音里徘徊 痴情笑我凡俗的人世,终难解的关怀

.....

题目描述

今年 28 岁的小嘉成为专职探险家已经有 5 年了。5 年前,由于突发的恶劣天气,小嘉的哥哥在一次探险任务中不幸牺牲。在他的最后一次探险前,对危险早已有所预感的哥哥将陪伴他多年的电子罗盘和探险笔记交给了小嘉,希望记录着他的足迹的电子罗盘和笔记,能够指引小嘉前进的方向,也能在世间留下关于他的种种回忆。5 年来,小嘉一直致力于实现哥哥未完成的梦想,在一片片荒原与一座座城市之间穿梭奔走,寻找笔记中最后的目标点。

这一次,小嘉来到了哥哥牺牲的那片荒原,这里是离兄弟两人的梦想最近的地方。广袤的荒原上散布着 n 座城市,有 m 条双向道路将这些城市相连。笔记上记录了 q 条线路,但由于年代已久,荒漠中的地形变化又快,现在的道路早已与笔记上所记录的路线大相径庭。令人欣慰的是,这 n 座城市的位置没有发生变化,这也意味着笔记中所记录线路的起点和终点仍然准确。没有了笔记的帮助,小嘉所能依赖的只有哥哥留下的那只电子罗盘。那只电子罗盘可以为小嘉找出从起点到终点的最短路线,并指引小嘉前行。若起点到终点的最短路线有多条,电子罗盘指引的线路可能是任意一条。

遗憾的是,由于电子罗盘年代已久,在极端严酷的环境下,电子罗盘的指引可能出现偏航的情况,即在某个城市,电子罗盘所指引的接下来要走的道路与它预期规划的道路不符。偏航情况发生后,待小嘉跟随指引走到罗盘指引的道路所连接的下一个城市时,电子罗盘会为小嘉重新规划一条从当前城市出发,到达目标点的最短路线。若起点到终点的最短路线有多条,电子罗盘指引的线路仍然可能是任意一条。哥哥生前告诉小嘉,这个罗盘在一次从笔记上的起点到终点的探险路途之中一定会偏航 k 次。

我们规定小嘉所走的路线指的是小嘉从起点出发到达终点的路途中,所经过的包含起点和终点的城市的编号顺次排列形成的数列。希望你对于笔记上 q 条线路中的每一条,输出小嘉在**从线路的起点走到终点**的路途中可能走的**不同路线的数量**。答案对 10^9+7 取模。

输入格式

第一行包含五个正整数 n, m, q, k, id, 其中 id 表示当前的测试点编号。样例输入中 id=0。

接下来的 m 行,每行有三个正整数 u_i , v_i , w_i , 表示有一条双向道路两端的城市和这条道路的距离。需要注意的是,不保证不存在起点和终点相同的道路,也不保证两两城市之间只有 1 条道路连接。

接下来的 t 行,每行有两个正整数 s_i , t_i ,表示笔记上第 i 条线路的起点和终点。保证起点和终点不是同一个城市。

输出格式

输出 t 行,第 i 行输出一个正整数,表示小嘉从第 i 条线路的起点走到终点的路途中,可能走的不同路线的数量。 答案对 10^9+7 取模。

若某条线路不可能满足偏航的要求,或是起点与终点之间不连通,请输出0。

样例 #1

样例输入#1

```
4 5 2 1 0
```

1 2 1

2 3 1

1 3 2

2 4 5

3 4 2

3
 4

样例输出#1

5 11

样例 #2

样例输入#2

见附件下的 dream2.in。

样例输出#2

见附件下的 dream2.out。

样例 #3

样例输入#3

见附件下的 dream3.in。

样例输出#3

见附件下的 dream3.out。

样例 #4

样例输入#4

见附件下的 dream4.in。

样例输出#4

见附件下的 dream4.out。

样例 #5

样例输入#5

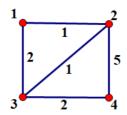
见附件下的 dream5.in。

样例输出#5

见附件下的 dream5.out。

提示

【样例1解释】



容易画出该样例如上图。

对于第一个询问,全程最短路线为 $1 \rightarrow 2 \rightarrow 3$ 和 $1 \rightarrow 3$ 。

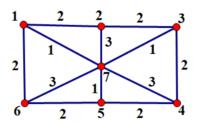
- 若最开始选择的是路线 $1\to 2\to 3$,可以在 1 处偏航至 3,路线为 $1\to 3$;可以在 2 处偏航至 1,路线为 $1\to 2\to 1\to 2\to 3$ 或 $1\to 2\to 1\to 3$;可以在 2 处偏航至 4,路线为 $1\to 2\to 4\to 3$ 。
- 若一开始选择的是路线 $1 \to 3$,可以在 1 处偏航至 2,路线为 $1 \to 2 \to 3$ 。

共计5条可能线路。

对于第二个询问, 总共的 11 条路线为:

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

【样例 2】



容易画出该样例如上图所示。

【样例 3】

该样例满足测试点6的限制。

【样例4】

该样例满足测试点17,18的限制。

【样例5】

该样例满足测试点 $21\sim25$ 的限制。

【数据范围】

对于 100% 的数据,保证: $2 \le n \le 1 \times 10^5$, $1 \le m \le 2 \times 10^5$, $1 \le w_i \le 10^9$, $q \le 20$ 。

测试点	$n \le$	$m \leq$	$q \leq$	k =	特殊性质
1, 2, 3, 4	10	15	10	1	无
5,6	1,000	2,000	20	0	无
7	1,000	2,000	20	1	A
8	1,000	2,000	20	1	B
9, 10, 11	1,000	2,000	20	1	无
12, 13, 14, 15	$1 imes10^5$	$2 imes10^5$	20	0	无
16	$1 imes10^5$	$2 imes10^5$	20	0	B
17, 18	$1 imes10^5$	$2 imes10^5$	20	1	B
19, 20	$1 imes10^5$	$2 imes10^5$	20	1	A
21, 22, 23, 24, 25	$1 imes10^5$	$2 imes10^5$	20	1	无

特殊性质 A: 保证给出的图是一条链。 特殊性质 B: 保证给出的图是一棵树。

尾声

几个月之后,小嘉终于来到了笔记上最终的那个目的地。走进山崖间的洞穴,小嘉在尽头的一座石台上找到一张写满文字的牛皮纸。小嘉辨认着字迹,泪水不禁夺眶而出。

牛皮纸上最后的几行字仍然清晰可见。

再见了, 小嘉。

——永远爱你的哥哥

2018年8月26日

测试点	$n \le$	$Q \leq$	$m \leq$	特殊性质
1	5	10	10	无
2	5	10	$1 imes10^6$	无
3	2,000	2,000	$1 imes 10^6$	A,B
4	2,000	2,000	$1 imes 10^6$	B
$5\sim 10$	2,000	2,000	$1 imes 10^6$	无
11, 12	$1 imes10^5$	$1 imes 10^5$	$1 imes 10^6$	A,B
13, 14	$1 imes10^5$	$1 imes10^5$	$1 imes10^6$	B
$15\sim 20$	$1 imes10^5$	$1 imes10^5$	$1 imes10^6$	无

特殊性质 A: 保证给出的树是一条链。

特殊性质 B:保证所有的 s_i 相等。

下围棋 (nine)

【题目背景】

*** (本内容AC后可自动解锁)

【题目描述】

平面上点 (0,0) 处有两颗黑色棋子,你可以通过一次操作将两颗黑色棋子同时移动 到距原来点曼哈顿距离恰好为 k 的同一处。你需要将两颗黑色棋子同时挪到点 (x,y) 处,最小化操作数并给出方案,若无解输出 -1。你可以认为两颗黑色棋子是一颗棋子。

【输入格式】

从文件 nine.in 中读入数据。 一行三个整数 k,x,y 。

【输出格式】

输出到文件 nine. out 中。

若无解,输出一行一个整数-1。

否则,第一行一个整数s,代表最小操作次数。

接下来 s 行, 第 i 行两个整数 (a_i, b_i) , 表示第 i 次操作后两颗黑色棋子的坐标。

如果有多组解,给出任意一组即可。

【输入输出数据】

见下发文件

【测试点约束】

对于所有数据, $1 \le k \le 10^9$, $-10^5 \le x, y \le 10^5$, $(x, y) \ne (0, 0)$ 。 你的输出中必须满足:

- $s < 10^6$.
- (a_i,b_i) 在 32 位有符号整数 (C++ 中的 int) 范围内。 可以证明, 如果有解, 存在满足上述限制的解。
- 子任务 1 (50 分): $k, |x|, |y| \leq 10$ 。
- 子任务 2 (50 分): 无特殊限制。

酒罐王(bar)

【题目描述】

给定一张 n 个节点带正边权有向简单图, 求第 k 短 1 到 n 的简单路径。 形式化地, 一条简单路径定义为序列 (p_1,p_2,\cdots,p_l) 满足:

- $p_1 = 1, p_l = n$.
- $\forall 1 \leq i < l$, 存在 p_i 连向 p_{i+1} 的有向边。
- $\forall 1 \leq i < j \leq l, p_i \neq p_j$ 。 定义其权值为经过的有向边的权值和。 称简单路径 p 比简单路径 q 短,当且仅当以下至少一条成立:
- p 的权值小于 q 的权值。
- p 的权值等于 q 的权值,且序列 (p_1, \dots, p_l) 的字典序小于序列 (q_1, \dots, q_r) 。

【输入格式】

从文件 bar.in 中读入数据。

第一行三个整数 n, m, k 。

接下来 m 行, 每行三个整数 u, v, w, 表示从 $u \rightarrow v$ 且权值为 w 的有向边。

【输出格式】

输出到文件 bar.out 中。

若无解输出一行一个整数 -1。

否则,第一行一个整数 l,表示答案路径 p 的点数。

第二行 l 个整数, 依次为 p_1, p_2, \dots, p_l 。

【输入输出数据】

见下发文件

选手目录下的 bar/bar4.in 与 bar/bar4.ans。 该样例约束与子任务 6 相同。

【测试点约束】

对于所有数据, 满足 $2 \leq n \leq 50, 1 \leq m \leq n(n-1), 1 \leq k \leq 500, 1 \leq w \leq 10^4$ 。

- 子任务 1 (15 分): *n* < 5
- 子任务 2 (15 分): n < 10
- 子任务 3 (15 分): $n \le 20$
- 子任务 4 (15 分): $n \le 30$
- 子任务 5 (15 分): n ≤ 40
- 子任务 6 (25 分): $n \le 50$