

第 35 届全国信息学奥林匹克竞赛

CCF NOI 2018

第一试模拟

竞赛时间：2017 年 7 月 12 日 7:30–12:30

题目名称	Game	Admirable	Tree
目录	game	admirable	tree
可执行文件名	game	admirable	tree
输入文件名	game.in	admirable.in	tree.in
输出文件名	game.out	admirable.out	tree.out
每个测试点时限	3 秒	1 秒	5 秒
内存限制	512M	512M	512M
测试点数目	捆绑测试	20	捆绑测试
每个测试点分值	--	5	--
是否有部分分	无	无	无
题目类型	传统型	传统型	传统型
是否有样例文件	是	是	是
是否有附加文件	否	否	否

提交源程序须加后缀

对于 C++语言	game.cpp	admirable.cpp	tree.cpp
对于 C 语言	game.c	admirable.c	tree.c
对于 Pascal 语言	game.pas	admirable.pas	tree.pas

编译开关

对于 C++语言	-O2 - 1m	-O2 - 1m	-O2 - 1m
对于 C 语言	-O2 - 1m	-O2 - 1m	-O2 - 1m
对于 Pascal 语言	-O2	-O2	-O2

1. Game

(game.pas/c/cpp)

1.1 description

小y是一个喜爱游戏的女孩子。

这天，小w和小y在玩一个经营国家的游戏。这个国家有一些交通线路，且这些线路呈一棵树的形状。

这个国家一共有 $n - 1$ 个城市和 n 个乡村，其中城市从1到 $n - 1$ 编号，乡村从1到 n 编号，且1号城市是首都。道路都是单向的，本题中我们只考虑从乡村通往首都的道路网络。

对于每一个城市，恰有一条公路和一条铁路通向这座城市。对于城市 i ，通向该城市的道路（公路或铁路）的起点，要么是一个乡村，要么是一个编号比 i 大的城市。没有道路通向任何乡村。除了首都以外，从任何城市或乡村出发只有一条道路；首都没有往外的道路。从任何乡村出发，沿着唯一往外的道路走，总可以到达首都。

小y在游戏中获得了一笔资金，她决定用这笔资金来改善交通。由于资金有限，她只能翻修任意的 $n - 1$ 条道路。

她希望从乡村通向城市可以尽可能地便利，于是根据人口调查的数据，她对每个乡村制定了三个参数，编号为 i 的乡村的三个参数是 a_i ， b_i 和 c_i 。假设从编号为 i 的乡村走到首都一共需要经过 x 条未翻修的公路与 y 条未翻修的铁路，那么该乡村的不便利值为 $c_i * (a_i + x) * (b_i + y)$ 。

在给定的翻修方案下，每个乡村的不便利值相加的和为该翻修方案的不便利值。翻修 $n - 1$ 条道路有很多方案，其中不便利值最小的方案称为最优翻修方案，小y自然希望找到最优翻修方案。请你帮助她求出这个最优翻修方案的不便利值。

1.2 input

第一行为正整数 n 。

接下来 $n - 1$ 行，每行描述一个城市。其中第 i 行包含两个数 s_i, t_i 。 s_i 表示通向第 i 座城市的公路的起点， t_i 表示通向第 i 座城市的铁路的起点。如果 $s_i > 0$ ，那么存在一条从第 s_i 座城市通往第 i 座城市的公路，否则存在一条从第 $-s_i$ 个乡村通往第 i 座城市的公路； t_i 类似地，如果 $t_i > 0$ ，那么存在一条从第 t_i 座城市通往第 i 座城市的铁路，否则存在一条从第 $-t_i$ 个乡村通往第 i 座城市的铁路。

接下来 n 行，每行描述一个乡村。其中第 i 行包含三个数 a_i, b_i, c_i ，其意义如题面所示。

1.3 output

输出一行一个整数，表示最优翻修方案的不便利值。

1.4 样例输入

6
2 3
4 5
-1 -2
-3 -4
-5 -6
1 2 3
1 3 2
2 1 3
2 3 1
3 1 2
3 2 1

1.5 样例输出

48

1.6 数据范围

subtask1(10'): $n \leq 12$ 。

subtask2(20'): $n \leq 50$ 。

subtask3(30'): $n \leq 2000$ 。

subtask4(40'): $n \leq 20000$ 。

对于100%的数据, $1 \leq a_i, b_i \leq 60, 1 \leq c_i \leq 10^9$, 任意乡村可以通过不超过40条道路到达首都。

2. Admirable

(admirable.pas/c/cpp)

2.1 问题描述

众所周知, 小 a 非常 *admirable*, 因此他有很多的仰慕者

小 a 有一个地盘, 他的地盘恰好是一棵 n 个节点的树, 作为 *admirable* 的小 a , 他可以分配 k 个仰慕者在他的地盘上进行巡逻, 每个仰慕者的巡逻的路径都是树上的一条简单路径

由于小 a 有时候要召集仰慕者们来做一些别的事情, 所以小 a 规定所有仰慕者的巡逻路径至少有一条公共边

同时, 小 a 为了维护他的 *admirable*, 他不允许存在一条边恰好被 $m \in [2, k - 1]$ 个仰慕者巡逻

此外, 小 a 为了让他变得更加 *admirable*, 他允许存在边没有被巡逻 (当然也可以所有边都被巡逻)

最让人 *admirable* 的是, 小 a 可以 1 秒内脑补出安排仰慕者的巡逻路径的方案数, 不过为了验证正确性以免装 ~~b~~ 失败, 小 a 要你再计算一下

注意, 对于巡逻路径 (a, b) 与 (b, a) 是等价的

由于仰慕者是两两不同的, 所以两个方案不同当且仅当至少存在一个仰慕者在这两个方案中的巡逻路径不同

2.2 输入格式

第一行 n, k

接下来 $n - 1$ 行, 第 i 行为 u_i, v_i , 表示一条边连接 u_i, v_i

2.3 输出格式

一行一个数, 表示方案数对 $10^9 + 9(1000000009)$ 取模的结果

2.4 样例输入

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

2.5 样例输出

```
7
```

2.6 样例解释

方案如下:

1. $((1, 2), (1, 2))$
2. $((1, 2), (1, 3))$
3. $((1, 3), (1, 2))$
4. $((1, 3), (1, 3))$
5. $((1, 3), (2, 3))$
6. $((2, 3), (1, 3))$
7. $((2, 3), (2, 3))$

2.7 数据范围

对于 10% 的数据, $1 \leq n, k \leq 5$

对于 30% 的数据, $1 \leq n, k \leq 100$

对于 60% 的数据, $1 \leq n, k \leq 5000$

对于另外 20% 的数据, 树的形态为链

对于 100% 的数据, $1 \leq n, k \leq 10^5$

3. Tree

(tree.pas/c/cpp)

3.1 description

给你一棵 n 个点的树，每个点有一个点权和所在集合编号。

有 m 个询问，每个询问有4种操作。

1:修改一个点的点权。

2:修改一个点的所在集合，保证修改前后点的集合不一样。

3:查询一条满足两 endpoints 都在集合 y 的最大路径权值，保证集合 y 不为空集。在这里，一条路径的权值定义为这条路径的在 y 集合的点的点权和。

4:查询一条满足所有点都在集合 y 的最大点权连通块，保证集合 y 不为空集。

请注意，一个单独的点也算一条路径或一个连通块。

由于小w有时想即时得到消息，所以他可能会让你在线解决这些问题。

3.2 input

第一行五个整数 n, m, k, T, O ，分别表示点数，询问数，集合数，是否强制在线和子任务编号。

接下来 n 行，每行两个整数 w_i, y_i ，分别表示 i 号点的点权，集合编号。

接下来 $n - 1$ 行，每行两个整数 u, v ，表示 u 到 v 之间有一条边。

接下来 m 行，每行第一个数为 op ，表示询问编号。且若 $T = 1$ ，那么这一行剩下的数全部都要异或上 g ，其中 $g = lastans \bmod 998244353$ ，初始 $lastans = 0$ 。

如果 $op = 1$ ，接下来两个整数 x, w ，表示将 x 的点权改为 w 。

如果 $op = 2$ ，接下来两个整数 x, y ，表示将 x 的集合改为 y 。

如果 $op = 3$ ，接下来一个整数 y ，表示查询一条满足两 endpoints 都在集合 y 的最大路径权值。

如果 $op = 4$ ，接下来一个整数 y ，表示查询一条满足所有点都在集合 y 的最大点权连通块。

3.3 output

对于所有的 $op \in \{3, 4\}$ 的操作，每行依次输出一个答案。

3.4 样例输入

5 5 5 0 1

476416874 1
-734291638 2
-209707479 3
248601378 1
473438305 5
2 1
3 1
4 1
5 4
4 1
4 5
1 3 857108629
2 5 4
1 2 716486897

3.5 样例输出

725018252
473438305

3.6 数据范围

subtask1(5'): $n, m \leq 10^4$ 。

subtask2(10'): 树是一条链。

subtask3(20'): $op \in \{1, 3\}$ 。

subtask4(15'): $k \leq 2$ 。

subtask5(15'): $T = 0$ 。

subtask6(35'): 无特殊性质。

对于100%的数据，满足 $n, m \leq 10^5$, $k \leq n$, $T \in \{0, 1\}$, $O \in \{1, 2, 3, 4, 5, 6\}$, $|w| \leq 10^9$, $y \leq k$ 。