

预览输出

题目名称	命运
题目类型	传统型
目录	destiny
可执行文件名	destiny
输入文件名	destiny.in
输出文件名	destiny.out
每个测试点时限	1.0 秒
内存限制	512 MB
子任务数目	25
测试点是否等分	否

提交源程序文件名

对于 C++ 语言	destiny.cpp
-----------	--------------------

编译选项

对于 C++ 语言	-std=c++11 -O2 -lm
-----------	---------------------------

命运 (destiny)

【题目描述】

提示：我们在题目描述的最后一段提供了一份简要的、形式化描述的题面。

在遥远的未来，物理学家终于发现了时间和因果的自然规律。即使在一个人出生前，我们也可以通过理论分析知晓他或她人生的一些信息，换言之，物理学允许我们从一定程度上“预言”一个人的“命运”。

简单来说，一个人的命运是一棵由时间点构成的有根树 T ：树的根结点代表着出生，而叶结点代表着死亡。每个非叶结点 u 都有一个或多个孩子 v_1, v_2, \dots, v_{c_u} ，表示这个人在 u 所代表的时间点做出的 c_u 个不同的选择可以导向的不同的可能性。形式化的，一个选择就是树上的一条边 (u, v_i) ，其中 u 是 v_i 的父结点。

一个人的一生是从出生（即根结点）到死亡（即某一个叶子结点）的一条不经过重复结点的路径，这条路径上任何一个包含至少一条边的子路径都是这个人的一段人生经历，而他或她以所有可能的方式度过一生，从而拥有的所有人生经历，都被称为潜在的人生经历。换言之，所有潜在的人生经历就是所有 u 到 v 的路径，满足 $u, v \in T$ ， $u \neq v$ ，并且 u 是 v 的祖先。在数学上，这样一个潜在的人生经历被记作有序对 (u, v) ，树 T 所有潜在的人生经历的集合记作 \mathcal{P}_T 。

物理理论不仅允许我们观测代表命运的树，还能让我们分析一些潜在的人生经历是否是“重要”的。一个人所作出的每一个选择——即树上的每一条边——都可能是重要或不重要的。一段潜在的人生经历被称为重要的，当且仅当其对应的路径上存在一条边是重要的。我们可以观测到一些潜在的人生经历是重要的：换言之，我们可以观测得到一个集合 $\mathcal{Q} \subseteq \mathcal{P}_T$ ，满足其中的所有潜在的人生经历 $(u, v) \in \mathcal{Q}$ 都是重要的。

树 T 的形态早已被计算确定，集合 \mathcal{Q} 也早已被观测得到，一个人命运的不确定性已经大大降低了。但不确定性仍然是巨大的——来计算一下吧，对于给定的树 T 和集合 \mathcal{Q} ，存在多少种不同的方案确定每条边是否是重要的，使之满足所观测到的 \mathcal{Q} 所对应的限制：即对于任意 $(u, v) \in \mathcal{Q}$ ，都存在一条 u 到 v 路径上的边被确定为重要的。

形式化的：给定一棵树 $T = (V, E)$ 和点对集合 $\mathcal{Q} \subseteq V \times V$ ，满足对于所有 $(u, v) \in \mathcal{Q}$ ，都有 $u \neq v$ ，并且 u 是 v 在树 T 上的祖先。其中 V 和 E 分别代表树 T 的结点集和边集。求有多少个不同的函数 $f: E \rightarrow \{0, 1\}$ （将每条边 $e \in E$ 的 $f(e)$ 值置为 0 或 1），满足对于任何 $(u, v) \in \mathcal{Q}$ ，都存在 u 到 v 路径上的一条边 e 使得 $f(e) = 1$ 。由于答案可能非常大，你只需要输出结果对 998,244,353（一个素数）取模的结果。

【输入格式】

从文件 `destiny.in` 中读入数据。

- 第一行包含一个正整数 n ，表示树 T 的大小，树上结点从 1 到 n 编号，1 号结点为根结点；

- 接下来 $n - 1$ 行每行包含空格隔开的两个数 x_i, y_i ，满足 $1 \leq x_i, y_i \leq n$ ，表示树上的结点 x_i 和 y_i 之间存在一条边，但并不保证这条边的方向；
 - 接下来一行包含一个非负整数 m ，表示所观测得到信息的条数。
 - 接下来 m 行每行包含空格隔开的两个数 u_i, v_i ，表示 $(u_i, v_i) \in \mathcal{Q}$ 。请注意：输入数据可能包含重复的信息，换言之可能存在 $i \neq j$ ，满足 $u_i = u_j$ 且 $v_i = v_j$ 。
- 输入数据规模和限制参见本题末尾的表格。

【输出格式】

- 输出到文件 `destiny.out` 中。
- 输出仅一行一个整数，表示方案数对 998,244,353 取模的结果。

【样例 1 输入】

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

【样例 1 输出】

```
1 10
```

【样例 1 解释】

- 共有 16 种方案，其中不满足题意的方案有以下 6 种：
- $(1, 2), (2, 3), (3, 5)$ 确定为不重要， $(3, 4)$ 确定为重要：集合 \mathcal{Q} 中没有限制被满足。
 - $(1, 2), (2, 3), (3, 4), (3, 5)$ 确定为不重要：集合 \mathcal{Q} 中没有限制被满足。
 - $(1, 2), (2, 3)$ 确定为不重要， $(3, 4), (3, 5)$ 确定为重要：集合 \mathcal{Q} 中 $(1, 3)$ 没被满足。
 - $(1, 2), (2, 3), (3, 4)$ 确定为不重要， $(3, 5)$ 确定为重要：集合 \mathcal{Q} 中 $(1, 3)$ 没被满足。
 - $(2, 3), (3, 5)$ 确定为不重要， $(1, 2), (3, 4)$ 确定为重要：集合 \mathcal{Q} 中 $(2, 5)$ 没被满足。
 - $(2, 3), (3, 4), (3, 5)$ 确定为不重要， $(1, 2)$ 确定为重要：集合 \mathcal{Q} 中 $(2, 5)$ 没被满足。
- 其他方案下，集合 \mathcal{Q} 中的限制都被满足了。

【样例 2 输入】

```
1 15
2 2 1
3 3 1
4 4 3
5 5 2
6 6 3
7 7 6
8 8 4
9 9 5
10 10 7
11 11 5
12 12 10
13 13 3
14 14 9
15 15 8
16 6
17 3 12
18 5 11
19 2 5
20 3 13
21 8 15
22 1 13
```

【样例 2 输出】

```
1 960
```

【样例 3】

见选手目录下的 *destiny/destiny3.in* 与 *destiny/destiny3.ans*。

【样例 4】

见选手目录下的 *destiny/destiny4.in* 与 *destiny/destiny4.ans*。

【测试点约束】

测试点编号	n	m	T 为完全二叉树			
1	≤ 10	≤ 10	否			
2						
3						
4						
5	≤ 500	≤ 15				
6	≤ 10000	≤ 10				
7	≤ 100000	≤ 16				
8	≤ 500000					
9	≤ 100000	≤ 22				
10	≤ 500000					
11	≤ 600	≤ 600				
12	≤ 1000	≤ 1000				
13	≤ 2000	≤ 500000				
14						
15	≤ 500000	≤ 2000				
16						
17	≤ 100000	≤ 100000	是			
18						
19	≤ 50000		否			
20	≤ 80000					
21	≤ 100000	≤ 500000				
22						
23	≤ 500000					
24						
25						

全部数据满足： $n \leq 5 \times 10^5$ ， $m \leq 5 \times 10^5$ 。输入构成一棵树，并且对于 $1 \leq i \leq m$ ， u_i 始终为 v_i 的祖先结点。

完全二叉树：在本题中，每个非叶结点都有左右子结点，且所有叶子结点深度相同的树称为满二叉树；将满二叉树中的结点按照从上到下、从左向右的顺序编号，编号最小的若干个结点形成的树称为完全二叉树。