

钥匙 (keys)

【题目描述】

有 n 座城市，编号为 $1, 2, \dots, n$ 。这些城市由 $n - 1$ 条无向道路相连，每条无向道路连接两座城市，保证任意两个城市连通。即这 n 座城市构成一棵树。

每座城市都有一件宝物。宝物分为两种：钥匙和宝箱。在一座城市里，要么有一把钥匙，要么有一个宝箱。钥匙和宝箱有不同的颜色，颜色为 i 的钥匙只能打开颜色为 i 的宝箱，打开宝箱后可以获得一枚金币，同时这把钥匙会损坏。

由于某种特殊的原因，同一种的钥匙最多只有 5 把（同一种颜色的宝箱数量不限）。

现在小 R 规划了 m 次旅行，第 i 次旅行的起点为 s_i ，终点为 e_i 。小 R 从 s_i 沿最短路径走到 e_i 。当他走到一座有钥匙的城市时，他可以将钥匙放入背包。当他走到一座有宝箱的城市时，如果他有相应颜色的钥匙，那么他就会打开这个宝箱并获得一个金币；如果他没有相应颜色的钥匙，那么他什么都不做（宝箱不能带走）。问每次旅行能获得多少枚金币。

注意：旅行相互独立，即一次旅行完之后所有的钥匙和宝箱都会恢复到初始状态。

【输入格式】

从文件 `keys.in` 中读入数据。

第一行两个正整数 n, m ，表示城市数量和旅行数量。

接下来 n 行，每行两个正整数 t_i, c_i ， $t_i = 1$ 表示第 i 个城市里有一把钥匙， $t_i = 2$ 表示有一个宝箱。 c_i 表示第 i 座城市里钥匙或宝箱的颜色。数据保证每种颜色的钥匙都不超过 5 把。

接下来 $n - 1$ 行，每行两个正整数 u_i, v_i ，表示有一条连接 u_i 和 v_i 的无向道路。

接下来 m 行，每行两个正整数 s_i, e_i 表示一次旅行的起点和终点。

【输出格式】

输出到文件 `keys.out` 中。

输出 m 行，每行一个整数，表示第 i 次旅行能获得的金币数量。

【样例 1 输入】

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

```

6 2 2
7 1 2
8 1 3
9 3 4
10 3 5
11 2 4
12 2 5
13 4 2

```

【样例 1 输出】

```

1 1
2 1
3 1

```

【样例 2】

见选手目录下的 *keys/keys2.in* 与 *keys/keys2.ans*。

【样例 3】

见选手目录下的 *keys/keys3.in* 与 *keys/keys3.ans*。

【样例 4】

见选手目录下的 *keys/keys4.in* 与 *keys/keys4.ans*。

该组样例满足 $n, m \leq 10^5$ 和特殊性质 A。

【子任务】

对于 100% 的数据, $1 \leq n \leq 5 \times 10^5, 1 \leq m \leq 10^6, 1 \leq t_i \leq 2, 1 \leq c_i, u_i, v_i, s_i, e_i \leq n$, 每种颜色的钥匙都不超过 5 把。

测试点编号	$n \leq$	$m \leq$	特殊性质
1	100	100	无
2 ~ 3	5000	5000	
4 ~ 5	10^5	10^5	
6 ~ 8	5×10^5	10^6	A
9 ~ 10			无

特殊性质 A：对于每种出现过的颜色，恰有一把钥匙和一个宝箱对应该颜色。

【提示】

输入输出数据较大，请使用较为快速的输入输出方式。