

K. 找树 / Findtree

时间限制： 8.0 秒

空间限制： 512 MiB

【题目描述】

定义 $\otimes_1, \otimes_2, \otimes_3$ 分别为按位与、按位或、按位异或运算。记 a_i 表示 a 的从低位到高位第 i 个二进制位。定义一个作用在 w 位二进制数上的新运算 \oplus ，满足对于结果 $a \oplus b$ 的每一位 $(a \oplus b)_i$ 有 $(a \oplus b)_i = a_i \otimes_{o_i} b_i$ 。不难验证 \oplus 运算满足结合律和交换律。

给出一张 n 个点 m 条边的无向图，每一条边的权值是一个 w 位二进制数（即小于 2^w 的非负整数）。请你找一棵原图的生成树。设你找出的生成树中的边边权分别为 v_1, \dots, v_{n-1} ，请你最大化 $v_1 \oplus v_2 \oplus \dots \oplus v_{n-1}$ 。

【输入格式】

从标准输入读入数据。

第一行两个正整数 n, m ；

第二行一个长度为 w 的串，串中的每个字符为 $\&$ 、 \mid 、 \wedge 中的一个（分别代表与、或和异或），表示每一个 \otimes_{o_i} 。

接下来 m 行，每一行三个非负整数 x, y, v ，表示一条连接 x 和 y 权值为 v 的边，保证 $1 \leq x, y \leq n$ ， $0 \leq v < 2^w$ 。

保证 $n \leq 70$ ， $w \leq 12$ ， $m \leq 5000$ 。

【输出格式】

输出到标准输出。

输出一行一个数，表示答案。如果图不连通，输出 -1。

【样例 1 输入】

```
3 3
^
1 2 1
2 3 1
1 3 0
```

【样例 1 输出】

```
1
```