

第 3 题：毒瘤(duliu)，运行时限 1s，内存上限 512M，100 分。

【问题描述】

从前有一名毒瘤。

毒瘤最近发现了量产毒瘤题的奥秘。考虑如下类型的数据结构题：给出一个数组，要求支持若干种奇奇怪怪的修改操作（例如给一个区间内的数同时加上 c ，或者将一个区间内的数同时开平方根），并且支持询问区间的和。毒瘤考虑了 n 个这样的修改操作，并将它们编号为 $1 \sim n$ 。当毒瘤要出数据结构题的时候，他就将这些修改操作中选若干个出来，然后出成一道题。

当然了，这样出的题有可能不可做。通过精妙的数学推理，毒瘤揭露了这些修改操作之间的关系：有 m 对“互相排斥”的修改操作，第 i 对是第 u_i 个操作和第 v_i 个操作。当一道题中同时含有 u_i 和 v_i 这两个操作时，这道题就会变得不可做。另一方面，当一道题中不包含任何“互相排斥”的操作时，这个题就是可做的。此外，毒瘤还发现了一个规律： $m - n$ 是一个很小的数字（参见“数据范围”中的说明），且任意两个修改操作都是连通的。两个修改操作 a, b 是连通的，当且仅当存在若干操作 t_0, t_1, \dots, t_l ，使得 $t_0 = a$ ， $t_l = b$ ，且对任意 $1 \leq i \leq l$ ， t_{i-1} 和 t_i 都是“互相排斥”的修改操作。

一对“互相排斥”的修改操作称为互斥对。现在毒瘤想知道，给定值 n 和 m 个互斥对，他一共能出多少道可做的不同的数据结构题。两个数据结构题是不同的，当且仅当其中某个操作出现在了其中一个题中，但是没有出现在另一个题中。

【程序文件名】

源程序文件名为 `duliu.cpp/c/pas`。

【输入格式】

输入文件名为 `duliu.in`。

第一行为正整数 n, m 。

接下来 m 行，每行两个正整数 u, v ，代表一对“互相排斥”的修改操作。

【输出格式】

输出文件名为 `duliu.out`。

输出一行一个整数，表示毒瘤可以出的可做的不同的数据结构题的个数。这个数可能很大，所以只输出模998244353后的值。

【输入输出样例 1】

| duliu.in | duliu.out |
|-------------------|-----------|
| 3 2 1 2 2 3 | 5 |

【样例解释 1】

可做的数据结构题有：空集， $\{1\}$ ， $\{2\}$ ， $\{3\}$ ， $\{1,3\}$ 。注意：空集是合法的数据结构题。

【输入输出样例 2】

| duliu.in | duliu.out |
|----------|-----------|
| 6 8 | 16 |
| 1 2 | |
| 1 3 | |
| 1 4 | |
| 2 4 | |
| 3 5 | |
| 4 5 | |
| 4 6 | |
| 1 6 | |

【输入输出样例 3】

| duliu.in | duliu.out |
|----------|-----------|
| 12 18 | 248 |
| 12 6 | |
| 3 11 | |
| 8 6 | |
| 2 9 | |
| 10 4 | |
| 1 8 | |
| 6 2 | |
| 11 5 | |
| 10 6 | |
| 12 2 | |
| 9 3 | |
| 7 6 | |
| 2 7 | |
| 3 2 | |
| 7 3 | |
| 5 6 | |
| 2 11 | |
| 12 1 | |

【数据范围】

| 测试点编号 | $n \leq$ | $m \leq$ | 测试点编号 | $n \leq$ | $m \leq$ |
|---------|----------|----------|---------|----------|----------|
| 1 ~ 4 | 20 | $n + 10$ | 5 ~ 6 | 100000 | $n - 1$ |
| 7 ~ 8 | 100000 | n | 9 | 3000 | $n + 1$ |
| 10 ~ 11 | 100000 | $n + 1$ | 12 ~ 14 | 3000 | $n + 10$ |
| 15 ~ 16 | 100000 | $n + 7$ | 17 ~ 20 | 100000 | $n + 10$ |

对所有数据, $n \leq 10^5$, $n - 1 \leq m \leq n + 10$ 。

【编译命令】

对于 c++ 语言: `g++ -o duliu duliu.cpp -lm`

对于 c 语言: `gcc -o duliu duliu.c -lm`

对于 pascal 语言: `fpc duliu.pas`