Road Expansion Plan (expansionplan)

The city of Pordenone is planning an expansion of its road network. There are N towns (indexed from 0 to $N-1$) in Pordenone's hinterland and M bidirectional roads (numbered from 0 to $M-1$) connecting some pairs of towns. For each $i = 0, 1, \ldots, N-1$, road i connects towns U_i and V_i.

An **expansion plan** is any set of roads that contains the currently present roads. E. Domora, the mayor of Pordenone, cannot choose an expansion plan among all the possible ones and is in desperate need of your analytical skills.

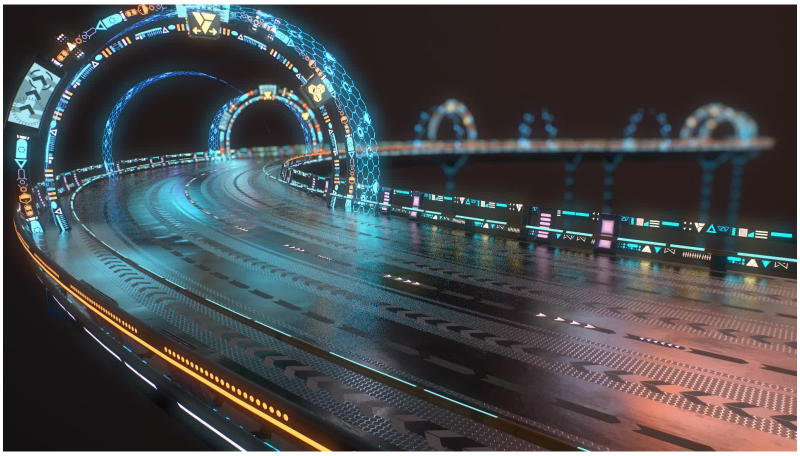


Figure 1: Your average road in Pordenone's hinterland.

A **cluster** C is a non-empty set of towns such that, for each $t \in C$ and any other town u, there exists a set of roads connecting (possibly indirectly) t and u if and only if $u \in C$. Note that, given a road network, there is an unique way to partition the towns into **clusters**.

The Scoazza™ Score of a road network is 2^k, where k is the number of **clusters** in which the network partitions the towns. Mayor Domora is asking you to compute the sum of the Scoazza™ Scores of all possible **expansion plans**. Since this number can be big, output it modulo 998244353.

> ✍ The *modulo* operation ($a \bmod m$) can be written in C/C++/Java/Python as `(a % m)`. To avoid the *integer overflow* error, remember to reduce all partial results through the modulus, and not just the final result!
> *Notice that if $x < 998244353$, then $2x$ fits into a* C/C++/Java `int`.

> ☞ Among the attachments of this task you may find a template file `expansionplan.*` with a sample incomplete implementation.

Input

The input file consists of:

- a line containing integers N, M.
- a line containing the M integers $U_0, \ldots, U_{M-1}$.
- a line containing the M integers $V_0, \ldots, V_{M-1}$.

Output

The output file must contain a single line consisting of a single integer: the answer to the problem.

Constraints

- $1 \leq N \leq 100\,000$.
- $0 \leq M \leq 200\,000$.
- $0 \leq U_i, V_i < N$ for each $i = 0 \ldots M-1$.
- $U_i \neq V_i$ for each $i = 0 \ldots M-1$.
- All the roads are distinct.

Scoring

Your program will be tested against several test cases grouped in subtasks. In order to obtain the score of a subtask, your program needs to correctly solve all of its test cases.

– **Subtask 1** (0 points) Examples.

– **Subtask 2** (17 points) $N \leq 7$.

– **Subtask 3** (40 points) $N \leq 1000$.

– **Subtask 4** (43 points) No additional limitations.

Examples

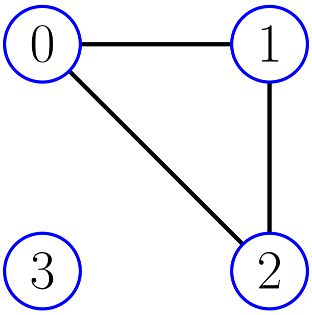
input	output

`4 3`
`0 1 2`
`1 2 0`
	`18`

`10 8`
`0 1 2 3 7 6 4 8`
`1 2 0 4 5 7 9 9`
	`360988397`

Explanation

In the **first sample case** there are 4 towns and 3 roads already built:



1. The road connecting town 0 and town 1
2. The road connecting town 1 and town 2
3. The road connecting town 2 and town 0

There are 3 possible roads that can be built:

1. The road connecting town 0 and town 3.
2. The road connecting town 1 and town 3.
3. The road connecting town 2 and town 3.

There are 8 possible expansion plans:

1. The one in which no road is added. In this case there will be 2 clusters: $0, 1, 2$ and 3.
2. The one in which only the road connecting towns 0 and 3 is added. In this case there will be a single cluster: $0, 1, 2, 3$.
3. ...

Note that, following any other expansion plan will lead to a single cluster containing all the cities. The answer is hence $2^2 + 2^1 + 2^1 + 2^1 + 2^1 + 2^1 + 2^1 + 2^1 = 18$.

In the **second sample case** there are 10 towns and 8 roads already built: