# Binary Tree Counting

| | |
|---|---|
| Input file: | `standard input` |
| Output file: | `standard output` |
| Time limit: | 2 seconds |
| Memory limit: | 1024 megabytes |

Find the number of binary trees* with $N$ vertices labeled $1, 2, \ldots, N$ that satisfy all of the following conditions, and output the result modulo 998244353.

- For each $i = 1, 2, \ldots, N$, the **preorder index** of vertex $i$ is $i$. In particular, vertex 1 is the root.
- For each $i = 1, 2, \ldots, M$, the **inorder index** of vertex $A_i$ is $B_i$.

Two binary trees are considered different if there exists a vertex $v$ such that at least one of the following holds:

- The existence or the vertex number of the left child of $v$ is different.
- The existence or the vertex number of the right child of $v$ is different.

The preorder index and inorder index of each vertex $v$ in a binary tree are defined as the values recorded in `preorder[v]` and `inorder[v]`, respectively, when executing `dfs(root)` in the following pseudocode:

```
pre_cnt = 1; in_cnt = 1
def dfs(v):
    preorder[v] = pre_cnt; pre_cnt += 1
    if v has a left child:
        dfs(left child of v)
    inorder[v] = in_cnt; in_cnt += 1
    if v has a right child:
        dfs(right child of v)
```

## Input

The input is given in the following format:

$N$ $M$
$A_1$ $B_1$
$A_2$ $B_2$
$\vdots$
$A_M$ $B_M$

- All input values are integers.
- $1 \leq M \leq N \leq 500$
- $1 \leq A_i, B_i \leq N$
- $A_i \neq A_j$ $(i \neq j)$
- $B_i \neq B_j$ $(i \neq j)$

## Output

Output the answer on a single line.

*A **binary tree** is a rooted tree in which each vertex has at most one left child and at most one right child.

## Examples

| standard input | standard output |
|---|---|
| 3 1<br>2 1 | 2 |
| 5 3<br>1 3<br>2 5<br>4 2 | 0 |
| 30 6<br>12 26<br>9 5<br>15 14<br>19 15<br>4 2<br>10 4 | 550222816 |

## Note

In the first example, there are exactly two binary trees that satisfy the conditions. One has vertex 1 as the root, vertex 2 as the left child of vertex 1, and vertex 3 as the right child of vertex 2. The other has vertex 1 as the root, vertex 2 as the left child of vertex 1, and vertex 3 as the right child of vertex 1. Their preorder and inorder indices are consistent with the given constraints.

In the second example, no binary tree satisfies the conditions.