On a social network consisting of m users and some friendships between users, two users can communicate with each other if they know a common language.

You are given an integer n, an array languages, and an array friendships where:

* There are n languages numbered 1 through n,
* languages[i] is the set of languages the i​​​​​​th​​​​ user knows, and
* friendships[i] = [u​​​​​​i​​​, v​​​​​​i] denotes a friendship between the users u​​​​​​​​​​​i​​​​​ and vi.

You can choose **one** language and teach it to some users so that all friends can communicate with each other. Return *the* ***minimum****number of users you need to teach.*

Note that friendships are not transitive, meaning if x is a friend of y and y is a friend of z, this doesn't guarantee that x is a friend of z.

**Example 1:**

**Input:** n = 2, languages = [[1],[2],[1,2]], friendships = [[1,2],[1,3],[2,3]]

**Output:** 1

**Explanation:** You can either teach user 1 the second language or user 2 the first language.

**Example 2:**

**Input:** n = 3, languages = [[2],[1,3],[1,2],[3]], friendships = [[1,4],[1,2],[3,4],[2,3]]

**Output:** 2

**Explanation:** Teach the third language to users 1 and 3, yielding two users to teach.

**Constraints:**

* 2 <= n <= 500
* languages.length == m
* 1 <= m <= 500
* 1 <= languages[i].length <= n
* 1 <= languages[i][j] <= n
* 1 <= u​​​​​​i < v​​​​​​i <= languages.length
* 1 <= friendships.length <= 500
* All tuples (u​​​​​i,v​​​​​​i) are unique
* languages[i] contains only unique values