1101. The Earliest Moment When Everyone Become Friends

In a social group, there are `N` people, with unique integer ids from `0` to `N-1`.
We have a list of `logs`, where each `logs[i] = [timestamp, id_A, id_B]` contains a non-negative integer timestamp, and the ids of two different people.
Each log represents the time in which two different people became friends. Friendship is symmetric: if A is friends with B, then B is friends with A.

Let's say that person A is acquainted with person B if A is friends with B, or A is a friend of someone acquainted with B.

Return the earliest time for which every person became acquainted with every other person. Return -1 if there is no such earliest time.

Input:

logs = [[20190101,0,1],[20190104,3,4],[20190107,2,3],[20190211,1,5], [20190224,2,4], [20190301,0,3], [20190312,1,2],[20190322,4,5]], **N** = 6

Output: 20190301

Explanation:

- The first event occurs at timestamp = 20190101 and after 0 and 1 become friends we have the following friendship groups [0,1], [2], [3], [4], [5].
- The second event occurs at timestamp = 20190104 and after 3 and 4 become friends we have the following friendship groups [0,1], [2], [3,4], [5].
- The third event occurs at timestamp = 20190107 and after 2 and 3 become friends we have the following friendship groups [0,1], [2,3,4], [5].
- The fourth event occurs at timestamp = 20190211 and after 1 and 5 become friends we have the following friendship groups [0,1,5], [2,3,4].
- The fifth event occurs at timestamp = 20190224 and as 2 and 4 are already friend anything happens.
- The sixth event occurs at timestamp = 20190301 and after 0 and 3 become friends we have that all become friends.

Note:

1. `1 <= N <= 100`
2. `1 <= logs.length <= 10^4`
3. `0 <= logs[i][0] <= 10^9`
4. `0 <= logs[i][1], logs[i][2] <= N - 1`
5. It's guaranteed that all timestamps in logs[i][0] are different.
6. `Logs` are not necessarily ordered by some criteria.
7. `logs[i][1] != logs[i][2]`