

3058. Friends With No Mutual Friends

Description

Table: Friends

```
+-----+-----+
\| Column Name \| Type \|
+-----+-----+
\| user_id1     \| int  \|
\| user_id2     \| int  \|
+-----+-----+
(user_id1, user_id2) is the primary key (combination of columns with unique values) for this table.
Each row contains user id1, user id2, both of whom are friends with each other.
```

Write a solution to find **all pairs** of users who are friends with each other and have **no mutual** friends.

Return *the result table ordered by* `user_id1`, `user_id2` *in ascending order.*

The result format is in the following example.

Example 1:

Input:

Friends table:

```
+-----+-----+
\| user_id1 \| user_id2 \|
+-----+-----+
\| 1        \| 2        \|
\| 2        \| 3        \|
\| 2        \| 4        \|
\| 1        \| 5        \|
\| 6        \| 7        \|
\| 3        \| 4        \|
\| 2        \| 5        \|
\| 8        \| 9        \|
+-----+-----+
```

Output:

```
+-----+-----+
\| user_id1 \| user_id2 \|
+-----+-----+
\| 6        \| 7        \|
\| 8        \| 9        \|
+-----+-----+
```

Explanation:

- Users 1 and 2 are friends with each other, but they share a mutual friend with user ID 5, so this pair is not included.
- Users 2 and 3 are friends, they both share a mutual friend with user ID 4, resulting in exclusion, similarly for users 2 and 4 who share a mutual friend with user ID 3, hence not included.
- Users 1 and 5 are friends with each other, but they share a mutual friend with user ID 2, so this pair is not included.
- Users 6 and 7, as well as users 8 and 9, are friends with each other, and they don't have any mutual friends, hence included.
- Users 3 and 4 are friends with each other, but their mutual connection with user ID 2 means they are not included, similarly for users 2 and 5 are friends but are excluded due to their mutual connection with user ID 1.

Output table is ordered by user_id1 in ascending order.

