1.3 Solution

This is a logical puzzle about equalities (friends) and inequalities (enemies).

The story tells us that the relation "friend of" is symmetric: two gangsters are friends (of each other). Then "friend of" can also be assumed reflexive: a gangster who is not a friend of himself cannot be a member of any gang, and this leads to the minimum answer $K = 0$ which we must try to improve. Ethics code 1 means that "friend of" is transitive. Hence "friend of" is an equality relation. It follows that a gang is an equivalence class of this relation.

These classes can be maintained as the well-known *Merge-Find* data structure [1, Chapter 21.3]. Initially each gangster is alone in his own class. Each fact `F` p q merges the classes for p and q.

Such merge operations include not only those given explicitly as input but also those that are given implicitly by ethics code 2 and the enemy facts `E` p q given as input: if p and q are enemies, and q and r are also enemies, then p and r are friends. Again, the story tells us that "enemy of" is symmetric.

These merge operations are exactly those that must be performed to represent the input. Hence the output K is the number of still distinct classes in the resulting data structure. However, we must also satisfy each fact `E` p q: if p and q end up in the same class, then the input is inconsistent, since p and q should be both friends and enemies, and the story tells us that these two relations are mutually exclusive. Hence the correct output is the minimum answer $K = 0$ in this case.

A rough complexity analysis is as follows. The algorithm needs $O(M)$ space: the Merge-Find structure for the gangster, and a *designated enemy* for every gangster. This designated enemy is used for maintaining the enemy information efficiently: by ethics code 2, all my enemies are friends of each other, so it suffices to remember the first of my enemies, and merge all later enemies with him. Initially nobody has a designated enemy.

The following algorithm needs $O(M + N\alpha(M))$ time, where α is the inverse of Ackermann's function [1, Chapter 21.4].

First, it takes $O(M)$ steps to initialize the data structures. Then it processes each of the N facts, one by one, as follows.

A friend fact `F` p q is processed by merging the classes of p and q.

An enemy fact `E` p q is in turn processed as follows. Consider p first. If p has no designated enemy r yet, then make q the designated enemy of p. Otherwise merge the classes of q and r. The other gangster q is processed symmetrically.

This generates less than $2N$ merge operations into a Merge-Find structure of M elements, and this takes $O(N\alpha(M))$ total time.

Then we can check that each enemy fact `E` p q is satisfied by checking that no gangster is in the same class as his designated enemy (if any). If such a gangster does exist, then $K = 0$.

Otherwise the correct answer K is the number of different classes. It can be computed by counting the number of those gangster nodes in the Merge-Find structure that are the roots of the trees representing the classes.

In either case, K can be computed in $O(M)$ steps.

Hence this problem can be solved with linear memory with respect to M and almost linear time with respect to N, so their upper bounds could be large in order to separate the optimal solutions from the slower ones.

References

[1] Cormen, T., Leiserson, C., Rivest, R., and Stein, C. *Introduction to Algorithms*, second ed. The MIT Press, 2001.