Kick Start 2022 - Round D

Analysis: Suspects and Witnesses

Let each suspect be a node in a graph G. If A is a witness to B, then form a directed edge from A to B.

It can be proven that a [strongly connected component (SCC)](#) C in G has the property that either:

1. Everyone is innocent.
2. Or everyone is a cookie stealer.

As a proof, if an arbitrary node u is innocent, then, for every edge e that goes from u to v, node v is also innocent. By induction, every node in C is innocent. Similarly, if an arbitrary node u is a cookie stealer, then every node v, where (u, v) is an edge in G, is also a cookie stealer. By induction, every node in C is a cookie stealer.

Given our graph G, we find all the SCC in G and form a new graph G' where:

1. Each SCC in G corresponds to a node in G'.
2. Each node in G' has a value w which corresponds to the size of the SCC.
3. If there exists an edge (u, v) in G and the two nodes do not share a SCC, there is an edge (u', v') in G' where u' is the corresponding node for u and v' is the corresponding node for v.

You can form G' in $O(|V| + |E|)$ using Tarjan's strongly connected components algorithm .

If there are at least $\mathbf{K}$ suspects that can reach node u in G, then u must be innocent.

To prove this, we use contradiction. If u is a cookie stealer, then every node that can reach u is a cookie stealer. Since there are at least $\mathbf{K}$ of them, this implies at least $\mathbf{K} + 1$ cookie stealers which is a contradiction.

Hence, if a node u in G belongs to a SCC (C_1) of size larger than $\mathbf{K}$, that node must be an innocent person. Therefore, we can iterate over all nodes and check this. However, since a node can belong to a different component (C_2) from which C_1 is reachable, then we have to check the reachability of each node to other nodes.

We can go through all the nodes of G' in topologically sorted order. We can go through all the neighbour nodes, i.e., nodes v that have a directed edge from the current node u to v and, since u is reachable to v, we can add u to the set that belongs to v. Whenever a set reaches the size of $\mathbf{K} + 1$, we can stop adding items to it. We will continue this process according to the topologically sorted order. Since we are going through all nodes (topologically), checking every edge at most once at the worst case and performing at most $\mathbf{K} + 1$ insertions into the set, the overall time complexity for the method would be $O((\mathbf{N} + \mathbf{M})\mathbf{K} \log \mathbf{K})$. Once all nodes have been visited, we can go through all the nodes and count the number of nodes that have a reachability set of size is more than $\mathbf{K}$ which should take $O(\mathbf{N})$.

Therefore, the overall time complexity of this solution would be $O((\mathbf{N} + \mathbf{M})\mathbf{K} \log \mathbf{K})$. In terms of space complexity, we would require $O(\mathbf{N})$ for SCC construction and SCC graph construction. The space complexity of finding reachability sets will be $O(\mathbf{NK})$ since for every node we can have at most $\mathbf{K} + 1$ nodes that we store. Therefore, the overall space complexity should be $O(\mathbf{NK})$.