Simurgh

Simplified Statement

Given a graph G with n vertices and m edges. Zal has selected a spanning tree of the graph, but you do not know which edges appear in his spanning tree. In every query, you can give him a spanning tree of the graph, and he will tell you how many edges your spanning tree has in common with his. You wish to find his spanning tree with a small number of queries.

Subtask 1

Iterate over all spanning trees and ask all of them.

Subtask 2

Start with an arbitrary spanning tree T and keep improving your solution as follows:

– Randomly choose an edge e.

– Add the edge to your solution.

– Remove a random edge from the cycle of $t \cup e$ to make it a tree T'.

– If T' has more edges in common with Zal's tree, then set $T \leftarrow T'$

– Stop if T is Zal's tree.

Subtask 3

In this subtask we can make exactly one query per edge. Decompose your graph into a number of disjoint (or almost disjoint) cycles. For each cycle C, find a tree T that connects C to all vertices of the graph ($C \cup T$ is a spanning tree with an extra edge). For each $e \in C$, determine the number of edges that $C \cup T\ e$ has in common with Zal's tree. If all of these numbers are equal, then none of the edges of C appear in Zal's tree. Otherwise, the edges whose removal decrease the number of the common edges are in Zal's tree.

Subtask 4

One can determine with 3 queries whether an edge e appears in Zal's tree; it only suffices to find 2 other edges that make a triangle together with e and do as mentioned earlier. Fix an arbitrary tree T and find out which of its edges appear in Zal's tree. Once we find that, for every forest F of G we can determine how many edges F shares with Zal's tree with a single query: add some of the edges of T to F to make it a spanning tree, query that tree, and determine how many edges of F are in common with Zal's tree. Determine the degree of each vertex in Zal's tree with n queries. Then we can find the edge connected of each leaf with $\log(n)$ queries and remove that edge from the solution. We continue with the new edges.

Subtask 5

The solution is almost the same as the previous subtask. The only difference is that finding a tree and determining which of its edges appear in Zal's tree is a bit harder. Roughly, we need to remove the cut edges (which we know are included in Zal's tree). Then every component is a 2-edge-connected graph and we can find an ear-decomposition for them. Note that for every cycle C we can figure out with $|C|$ queries which edges of C are in Zal's tree. The only extension that we need to that is that if we already know the status of k edges of C, we can do this with $|C| + k - 1$ queries. Therefore, we can solve the problem for each component separately with at most $2n$ queries.