# Simurgh

## Simplified Statement

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

## Subtask 1

Iterate over all spanning trees and try 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

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 ech $e \in C$, determine the number of edges that $C \cup t\ e$ has in common with Zal's tree. If all 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 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 edge $F$ 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 every time we find the incident edge of a leaf with $\log(n)$ queries and remove that edge from the solution. We continue on with the new edges.

## Subtask 5

The same as 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 speaking, 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 $|C|$ querier 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 $|C| + k - 1$ queris. Therefore, we can solve the problem for each component separately with at most $2n$ quires