P1) Let $G$ be a tree. Use induction to prove that the number of leaves of $G$ is at least the number of vertices of degree at least 3 in $G$. For example, the following tree has 3 leaves and 1 vertex of degree at least 3, and $3 \geq 1$.

P2) Let $G$ be a graph with $n$ vertices and at least $n$ edges. Show that $G$ has a cycle.

P3) Given a connected undirected graph $G = (V, E)$ with $n$ vertices and $m$ edges. Design an $O(m + n)$ time algorithm that outputs an edge $e$ of $G$ such that if we delete $e$, $G$ remains connected. If no such edge exists output “Impossible”. For example in the following graph if you delete the red edges the graph remains connected.

We write the psuedo-code below, although the above description is already enough:
Function $BFS(s)$

Initialize: mark all vertices “undiscovered”
mark $s$ “discovered”
queue $= \{ \ s \ \}$
while queue not empty do
    $u = \text{remove}_\text{first}(\text{queue})$
    for each edge $\{u, x\}$ do
        if $x$ is “undiscovered” then
            mark $x$ “discovered”
            append $x$ on queue
        end
    else
        output $\{u, x\}$ and end the algorithm
    end
end
mark $u$ “fully-explored”
end
output “Impossible”

Algorithm 1: Algorithm for P3