P1) The main goal of this question is to answer the problem in part (d). If you prefer you can ignore parts (a,b,c) and solve (d) directly.

a) Optional Let $T$ be a tree with $n$ vertices. Prove that there is a unique path in $T$ between each pair of vertices.

b) Optional Let $G = (V, E)$ be a connected undirected graph and $T \subseteq E$ be a spanning tree of $G$. Prove that for any edge $e \in T$, $G - e$ is connected iff there is an edge $f = (u, v) \in E - T$ such that the unique path between $u, v$ in $T$ has the edge $e$.

c) Show how to modify the code for recursive depth-first search of undirected graphs to obtain an $O(n + m)$ time algorithm that (i) assigns each node $v$ a number, dfsnum($v$), indicating a sequence number for when $v$ was first visited by DFS, and computes $\text{min}(v)$ for each node $v$, the smallest dfsnum of any node that was encountered in the recursive call $\text{dfs}(v)$.

For example, in the following picture edges of the DFS tree are marked in solid (and non-tree edges in dashed). Every node is labelled with its dfsnum. So, $\text{min}(\cdot)$ for the red node is 1 and min of the blue node is 5.

![Diagram of a tree with dfs numbers](image)

d) Given a graph $G = (V, E)$ with $n$ vertices and $m$ edges, design an $O(m + n)$ time algorithm that for any edge $e \in E$ outputs if $G - e$ is connected. For example, given the following graph you should output “yes” for all black edges and “no” for the red edge.

![Diagram of a graph](image)

P2) Prove or disprove: Every directed graph with $n$ vertices and at least $n(n - 1)/2 + 1$ directed edges has a cycle.