1) Tree diameter: The diameter of a tree $T = (V, E)$ is defined as $\max_{u,v \in V} \delta(u, v)$, i.e. the largest of all shortest-path distances in the tree. Give an efficient algorithm to compute the diameter of a tree, and analyze the running time of your algorithm. Hint: Think about what happens when you run BFS from any node. The tree in this case is *unrooted*.

The algorithm is to run BFS from an arbitrary node. The farthest node from the starting point is one end of a diameter path. Initiate another BFS here, then the farthest node will be at the other end of the diameter and it's distance is the diameter of the tree.

Why is this correct? The crux is to prove that the first BFS, from any starting point *s*, finds an endpoint of the diameter. Given that the graph is connected, BFS will eventually explore some node *x* on the diameter at distance $\delta(s, x)$. The farthest node from *x* must be one of the endpoints *a* --- if not, the diameter path actually isn't the largest path in the graph. For the same reason, $\delta(s, x) \leq \delta(x, a)$ and *a* is the farthest node from *s*. We need the graph to be a tree so that paths between nodes are unique and thus there is no alternate path that is longer than the shortest path.

2) DFS: Explain how a vertex u of a directed graph can end up in a depth-first tree containing only u, even though u has both incoming and outgoing edges in G.

A simple example is

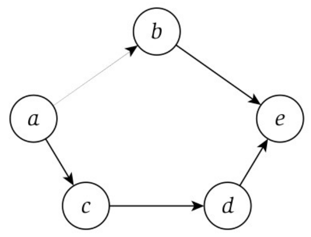
a → u → b

Start the DFS at b, it will finish, then restart it at u and again restart at a. All three nodes will be in their own trees.

3) Prove that in every directed, acyclic graph (DAG = a directed graph with no cycles) there is a node v with no incoming edges.

If all nodes have incoming edges then we can follow these edges backwards forever. Since the graph is finite, eventually we will have traversed a cycle. This is a contradiction, since the graph is a DAG. Thus there must be some node without any incoming edges.

4) How many topological orderings does this DAG have? List them.



a,b,c,d,e
a,c,b,d,e
a,c,d,b,e

Just 3 here.