Question 1 (Pre- and Post-Orders, 30 points). Run DFS on the graph below and compute the pre-order and post-order numbers of each vertex. Whenever, the algorithm needs to make a decision about what order to do things in, always use alphabetical order.
**Question 2** (DAG-ification, 35 points). *Let G be a finite, directed graph that consists of a single strongly connected component. Suppose that there is an edge e in G from a vertex v to a vertex w so that removing e from G leaves a DAG. Prove that e is the only edge in G leaving the vertex v.*
Question 3 (Shapeshifter Maze, 35 points). Tristan is a shapeshifter. He has one form that can survive on land and another that can survive on water. Unfortunately, he is running out of energy to power his transformations and can only transform three more times before recharging. He has a map leading to a leyline that could be used to recharge.

The map is a graph $G$. Each vertex is either on land or under water and can only be survived in the correct form. Each edge takes exactly an hour for Tristan to traverse, and when he reaches a new vertex he can transform (if necessary) into a form suitable to survive there.

Suppose that you are given the graph $G$ along with markings as to which vertices are on land or in the water along with the locations of Tristan’s current location and the location of the leyline. Design an algorithm that computes the minimum number of hours that it will take Tristan to reach the leyline without using more than three transformations. For full credit your algorithm should run in linear time or better.