EECS 343: Theoretical Computer Science, Homework Exercise 11
due Monday, April 13, 2020 before class

Problem 1: (Sipser 8.23) Prove that UCYCLE is in L where UCYCLE takes an undirected graph G as input and asks if G contains a simple cycle. G does not have to be connected. (Hint: PATH is not known to be in L so your algorithm has to detect cycles without doing an arbitrary path exploration.)

Problem 2: (The second half of Sipser 8.25) An undirected graph is **bipartite** if its nodes may be divided into two sets so that all edges go from a node in one set to a node in the other set. This is equivalent to stating that the graph does not contain a cycle that has an odd number of vertices. Let BIPARTITE $= \{\langle G \rangle \mid G$ is bipartite$\}$. Prove that BIPARTITE $\in$ NL.

Problem 3: (Sipser 8.27) A directed graph is **strongly connected** if every two nodes are connected by a directed path in each direction. Let

$$\mathrm{STRONGLY-CONNECTED} = \{\langle G \rangle \mid G \text{ is a strongly connected graph}\}.$$

Prove that STRONGLY-CONNECTED is NL-complete.